# MySQL 8.0 Release Notes

**Abstract**

**Important**

As of April 2026, with version 8.0.46, MySQL 8.0 reaches End of Life (EoL). MySQL 8.0 users are encouraged to upgrade to the latest MySQL 8.4 LTS or MySQL Innovation release.

This document contains release notes for the changes in each release of MySQL 8.0, up through MySQL 8.0.46. For information about changes in a different MySQL series, see the release notes for that series.

For additional MySQL 8.0 documentation, see the MySQL 8.0 Reference Manual, which includes an overview of features added in MySQL 8.0 (What Is New in MySQL 8.0), and discussion of upgrade issues that you may encounter for upgrades from MySQL 5.7 to MySQL 8.0 (Changes in MySQL 8.0).

Before upgrading to MySQL 8.0, review the information in https://docs.oracle.com/cd/E17952_01/mysql-8.0-en/upgrading.html and perform any recommended actions. Perform the upgrade on a test system first to make sure everything works smoothly, and then on the production system.

Downgrade from MySQL 8.0 to MySQL 5.7, or from a MySQL 8.0 release to a previous MySQL 8.0 release, is not supported. The only supported alternative is to restore a backup taken before upgrading. It is therefore imperative that you back up your data before starting the upgrade process.

MySQL platform support evolves over time; please refer to https://www.mysql.com/support/supportedplatforms/database.html for the latest updates.

Updates to these notes occur as new product features are added, so that everybody can follow the development process. If a recent version is listed here that you cannot find on the download page (https://dev.mysql.com/downloads/), the version has not yet been released.

The documentation included in source and binary distributions may not be fully up to date with respect to release note entries because integration of the documentation occurs at release build time. For the most up-to-date release notes, please refer to the online documentation instead.

For legal information, see the Legal Notices.

For help with using MySQL, please visit the MySQL Forums, where you can discuss your issues with other MySQL users.

Document generated on: 2026-04-28 (revision: 31150)

# Table of Contents


Preface and Legal Notices ..... 2
Changes in MySQL 8.0.46 (2026-04-21) ..... 4
Changes in MySQL 8.0.45 (2026-01-20) ..... 6
Changes in MySQL 8.0.44 (2025-10-21) ..... 7
Changes in MySQL 8.0.43 (2025-07-22) ..... 9
Changes in MySQL 8.0.42 (2025-04-15) ..... 14
Changes in MySQL 8.0.41 (2025-01-21) ..... 20
Changes in MySQL 8.0.40 (2024-10-15) ..... 26
Changes in MySQL 8.0.39 (2024-07-23) ..... 31
Changes in MySQL 8.0.38 (2024-07-01) ..... 31
Changes in MySQL 8.0.37 (2024-04-30) ..... 36
Changes in MySQL 8.0.36 (2024-01-16) ..... 42

Changes in MySQL 8.0.35 (2023-10-25) ..... 45
Changes in MySQL 8.0.34 (2023-07-18) ..... 51
Changes in MySQL 8.0.33 (2023-04-18) ..... 62
Changes in MySQL 8.0.32 (2023-01-17) ..... 71
Changes in MySQL 8.0.31 (2022-10-11) ..... 84
Changes in MySQL 8.0.30 (2022-07-26) ..... 98
Changes in MySQL 8.0.29 (2022-04-26) ..... 114
Changes in MySQL 8.0.28 (2022-01-18) ..... 131
Changes in MySQL 8.0.27 (2021-10-19) ..... 150
Changes in MySQL 8.0.26 (2021-07-20) ..... 171
Changes in MySQL 8.0.25 (2021-05-11) ..... 195
Changes in MySQL 8.0.24 (2021-04-20) ..... 195
Changes in MySQL 8.0.23 (2021-01-18) ..... 211
Changes in MySQL 8.0.22 (2020-10-19) ..... 228
Changes in MySQL 8.0.21 (2020-07-13) ..... 247
Changes in MySQL 8.0.20 (2020-04-27) ..... 269
Changes in MySQL 8.0.19 (2020-01-13) ..... 289
Changes in MySQL 8.0.18 (2019-10-14) ..... 314
Changes in MySQL 8.0.17 (2019-07-22) ..... 334
Changes in MySQL 8.0.16 (2019-04-25) ..... 360
Changes in MySQL 8.0.15 (2019-02-01) ..... 385
Changes in MySQL 8.0.14 (2019-01-21) ..... 385
Changes in MySQL 8.0.13 (2018-10-22) ..... 408
Changes in MySQL 8.0.12 (2018-07-27) ..... 432
Changes in MySQL 8.0.11 (2018-04-19) ..... 448
Changes in MySQL 8.0.5 - 8.0.10 (Skipped version numbers) ..... 472
Changes in MySQL 8.0.4 (2018-01-23, Release Candidate) ..... 472
Changes in MySQL 8.0.3 (2017-09-21, Release Candidate) ..... 507
Changes in MySQL 8.0.2 (2017-07-17, Development Milestone) ..... 528
Changes in MySQL 8.0.1 (2017-04-10, Development Milestone) ..... 554
Changes in MySQL 8.0.0 (2016-09-12, Development Milestone) ..... 597
Index ..... 629

# Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL 8.0, up through MySQL 8.0.46.

## Legal Notices

Copyright © 1997, 2026, Oracle and/or its affiliates.

**License Restrictions**

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

**Warranty Disclaimer**

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

**Restricted Rights Notice**

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

**Hazardous Applications Notice**

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

**Trademark Notice**

Oracle, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

**Third-Party Content, Products, and Services Disclaimer**

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

**Use of This Documentation**

This documentation is NOT distributed under a GPL license. Use of this documentation is subject to the following terms:

You may create a printed copy of this documentation solely for your own personal use. Conversion to other formats is allowed as long as the actual content is not altered or edited in any way. You shall not publish or distribute this documentation in any form or on any media, except if you distribute the documentation in a manner similar to how Oracle disseminates it (that is, electronically for download on a Web site with the

software) or on a CD-ROM or similar medium, provided however that the documentation is disseminated together with the software on the same medium. Any other use, such as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

## Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

# Changes in MySQL 8.0.46 (2026-04-21)

**Note**

These release notes were created with the assistance of MySQL HeatWave GenAI.


- Audit Log Notes
- Compilation Notes
- SQL Function and Operator Notes
- InnoDB Notes
- JSON Notes
- Optimizer Notes
- Packaging Notes
- X Plugin Notes
- Bugs Fixed


## Audit Log Notes

- Fixed an issue related to processing certain gzip files. The MySQL Server has been updated to relax the `.gz` header checks, allowing ordinary gzip files to be processed. Errors were returned similar to the following:

  ```
  Invalid audit log file content
  ```

  (Bug #38980223)

## Compilation Notes

- The included `zlib` library has been upgraded to version 1.3.2. (Bug #38987448)

## SQL Function and Operator Notes

- Fixed an issue relating to the `DEFAULT()` function. (Bug #39057054)

## InnoDB Notes

- Memory usage during FTS index construction for large tables has been optimized. (Bug #39040226)
- Fixed an issue relating to multi-value indexes. (Bug #39040128)
- Fixed an issue in the parallel reader. (Bug #39033858)
- The MySQL truncate undo operation (purge thread) did not remove the `undo_{space_number}_trunc.log` file when attempting to truncate the undo tablespace. (Bug #38871808)
- `dict_sdi_create_idx_in_mem` function added columns in a different order than expected, with the `compressed_len` and `uncompressed_len` fields swapped. (Bug #38810801)
- Under certain circumstances, running `CREATE INDEX` with a high value for `--innodb_parallel_read_threads` could cause the disk space to fill up, leading to disk space exhaustion. (Bug #38370155)
- Fixed an issue related to `TRUNCATE TABLE`. (Bug #38169053)
- Under certain circumstances, when calculating the maximum possible index record size, an assertion failure could occur. (Bug #85060, Bug #25579578)

## JSON Notes

- Fixed an issue relating to processing of JSON files. (Bug #39000847)

## Optimizer Notes

- Fixed an issue relating to optimized prepared DELETE and UPDATE statements. (Bug #39071552)
- Fixed an issue relating to query management. (Bug #38928287)
- Fixed an issue relating to memory management. (Bug #38573278)
- Queries using semi join with materialization may return incorrect results due to a missing condition.

  Our thanks to Jingqi Tian and the team at Alibaba. (Bug #38110792)
- Fixed an issue relating to the Optimizer's SQL planner. (Bug #35634700)
- In an explain expanded query, join order hints were not printed with a valid syntax.

  Our thanks to Kaiwang Chen for the contribution. (Bug #116084, Bug #37053765)
- Conditions on const tables and on previously joined tables were not consistently taken into account when building ranges in `get_quick_record_count()`. This could lead to suboptimal or incorrect range estimates, especially when range predicates depended on values from const or earlier-joined tables.

  Our thanks to Xingyu Yang for the contribution. (Bug #112737, Bug #35912840)

## Packaging Notes

- For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 3.5.5. For more information, see OpenSSL 3.5 Series Release Notes. (Bug #38867109, WL #17209)

## X Plugin Notes

- The system variable `caching_sha2_password_digest_rounds` could not be set to a non-default value using X Protocol. (Bug #36402455)

## Bugs Fixed

- Fixed an issue relating to Information Schema privileges. (Bug #35243416, Bug #37387633)
- The SQL parser consumed a large amount of memory when parsing very large queries with many large `IN` clauses.

  As of this release, the SQL parser's memory management has been improved. (Bug #105004, Bug #33390851)

# Changes in MySQL 8.0.45 (2026-01-20)

**Note**

These release notes were created with the assistance of MySQL HeatWave GenAI.


- InnoDB Notes
- Optimizer Notes
- Packaging Notes
- Bugs Fixed


## InnoDB Notes

- **InnoDB:** InnoDB redo logging error messages now include the current LSN and redo log capacity information. A new warning code `ER_IB_WRN_REDO_DISABLED_INFO` is added which includes the current LSN. A new error code `ER_IB_MSG_LOG_WRITER_WAIT_ON_NEW_LOG_FILE_INFO` is added to include the current log capacity and the current log capacity used. The `MONITOR` output has been enhanced to include redo log capacity details and current log capacity.

  The warning `ER_IB_WRN_REDO_DISABLED` was removed, as was the error `ER_IB_MSG_LOG_WRITER_WAIT_ON_NEW_LOG_FILE`. (Bug #37645185)
- Fixed an issue related to bulk inserts. (Bug #38208188)

## Optimizer Notes

- Fixed an issue relating to Common Table Expressions (CTE). (Bug #38573285)
- Fixed an issue related to executing specific SQL queries. (Bug #38448700)
- Fixed an issue relating to `SHOW CREATE TABLE`. (Bug #38298692)

- Queries which used REGEXP took longer to execute as prepared statements than as direct queries. (Bug #114056, Bug #36326728)

## Packaging Notes

- For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 3.0.18. For more information, see OpenSSL 3.0 Series Release Notes. (Bug #38632932)

## Bugs Fixed

- **InnoDB:** Under certain circumstances, when using the `que_eval_sql` interface, a race condition could occur. (Bug #118705, Bug #38310595)
- Binary log purged before persisted `binlog_expire_*` options were loaded. (Bug #38554467)
- When 2 of 6 instances in a geographically dispersed InnoDB Cluster lost connectivity, the primary server became unresponsive, causing the `COMMIT` and `INSERT` operations to become unresponsive as well. (Bug #38380392)
- Fixed several issues relating to Thread Pool blocking connections from closing. (Bug #38170188, Bug #36782728, Bug #38549372)
- GTID gaps were generated because of the `replica- skip-errors` option. (Bug #28590993)
- If server was started on Windows platforms with --skip-grant-tables, privileges were not checked and remote connections were not permitted. As a result, the server might not start. (Bug #118805, Bug #38328780)

# Changes in MySQL 8.0.44 (2025-10-21)

**Note**

These release notes were created with the assistance of MySQL HeatWave GenAI.


- Audit Log Notes
- Compilation Notes
- SQL Function and Operator Notes
- InnoDB Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Bugs Fixed


## Audit Log Notes

- Improved performance related to adding multiple `general_sql_command.str` fields to an Enterprise Audit filter. (Bug #38053242)
- Fixed an issue related to `audit_log_read` performance. (Bug #36281295, Bug #38275868)

## Compilation Notes

- OpenSSL 3 is now used on Solaris. (Bug #38193362)
- Upgraded the bundled googletest library to version 1.17. (Bug #38152017)
- Upgraded the bundled libbacktrace library to the latest version (April 2025). (Bug #38151970)
- Warnings raised by Clang 20 for non-trivially copyable types, deprecated literal operators, and incorrect `main` function declarations are no longer generated. (Bug #37785251)

## SQL Function and Operator Notes

- Fixed an issue related to executing certain UPDATE statements. (Bug #37590580)

  References: This issue is a regression of: Bug #31562881.

## InnoDB Notes

- Creating a thread for parallel scan may fail, resulting in assertion failures when it fell back to single-thread mode. Errors were returned similar to the following:

  ```
  Assertion failure: row0pread.h:306:active >= n_threads
  ```

  (Bug #38325137)
- Under certain circumstances, a virtual index rollback could fail on 32-bit builds of MySQL Server. (Bug #38167527)
- If the buffer pool was very large, it could lead to a large number of chunks required by each buffer pool instance, which could fail if the operating system was unable to allocate the required memory.

  As of this release, this allocation is checked to ensure it is allocated correctly. (Bug #37994397)
- Users could encounter an assertion failure due to a data size and bounds mismatch. Errors were returned similar to the following:

  ```
  Assertion failure: ddl0file-reader.cc:193:m_ptr + data_size <
  m_bounds.second
  ```

  (Bug #37882398)

  References: This issue is a regression of: Bug #37233273.
- Fixed an issue relating to modifying the internal FTS configuration. (Bug #37792010)
- Fixed an issue related to virtual indexes. (Bug #37602657)
- If the row size exceeded the maximum allowed size due to `innodb_strict_mode=OFF`, confusing warning messages similar to "Cannot add field" could appear in certain situations, such as selecting from a table or dropping a column.

  Error messages are improved for this scenario. (Bug #37003342, Bug #36768046, Bug #36867372)

## Optimizer Notes

- Fixed an issue relating to SQL queries involving correlated scalar subqueries and JSON tables. (Bug #36684370)

## Packaging Notes

- For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 3.0.17. For more information, see OpenSSL 3.0 Series Release Notes. (Bug #38457056)
- The RPM specification and DEB control files were missing several requirements, making it impossible to build those formats. (Bug #38310603)

## Performance Schema Notes

- Connections waiting on the connection control plugin were not displayed in the `performance_schema.processlist` table. (Bug #38043683)

## Bugs Fixed

- Logical clock handling for `CREATE ... SELECT` statements has been improved to ensure accurate logical clock values in the replica relay log. (Bug #38383106)
- The server sometimes processed table references incorrectly. (Bug #38001000)
- `PARTITION BY ... (DEFAULT (column))` was not always handled correctly. (Bug #111234, Bug #35451459)

  References: This issue is a regression of: Bug #33142135.

# Changes in MySQL 8.0.43 (2025-07-22)

**Note**

These release notes were created with the assistance of MySQL HeatWave GenAI.


- Compilation Notes
- InnoDB Notes
- Installation Notes
- Packaging Notes
- Functionality Added or Changed
- Bugs Fixed


## Compilation Notes

- **macOS:** It is now possible to compile the server on MacOS using `-DWITH_KERBEROS`.
- Upgraded the bundled `libcurl` library to version 8.14.1. (Bug #38042758)
- MySQL Server now supports CMake 4, ensuring compatibility with future CMake versions where support for versions prior to 3.10 is expected to be discontinued. (Bug #38027636)
- The included ICU library has been upgraded to version 77-1. (Bug #37870791)

- The included `zstd` library has been upgraded to version 1.5.7. (Bug #37869972)
- Disabled `clang::musttail` when building with GCC 15. (Bug #37776018)
- Implemented the standards-compliant `my_char_traits<unsigned char>` for use as a drop-in replacement for `std::char_traits<unsigned char>`, which was deprecated in Clang 18 and removed in Clang 19. (Bug #37273525, Bug #37785339)
- The linker tried to use the empty Cmake variable `${ICU_LIBRARIES}`, even though the correct library (`ext::icu`) was already linked elsewhere. (Bug #36524167)
- When compiling on Fedora 38, `grep -E` is now used in place of `egrep`. (Bug #36507549)
- Compiling on macOS looked explicitly for `openssl@1.1` but now looks for the generic `openssl` symlink instead, to allow for `openssl@3`. (Bug #35468370)

## InnoDB Notes

- Fixed an issue relating to importing tables. (Bug #37621360)

## Installation Notes

- Debian packages for installing MySQL can now be run by users other than root. This helps prevent issues for Debian or Ubuntu systems that require rootless installations. (Bug #37765153)

## Packaging Notes

- On Fedora 24, Oracle MySQL RPMs could not be installed due to package conflicts when MariaDB was already installed on the system. With this fix, the conflicts are not properly handled to allow MySQL Server to be installed successfully. (Bug #37798784)

## Functionality Added or Changed

- **Important Change:** Added the `mysql` client `--commands` option, which enables or disables most `mysql` client commands.

  This option is enabled by default. To disable it, start the client with `--commands=OFF` or `--skip-commands`.

  For a complete list of all commands affected by this option, and additional information, see mysql Client Options. (WL #16949)

  References: See also: Bug #36416568, Bug #38066040.

- **Group Replication:** Added the error `ER_GRP_RPL_APPLIER_THD_KILLED`, to distinguish when the applier thread has been terminated using SQL KILL, rather than stopped by an error. (Bug #37764717)
- Binary packages that include `curl` rather than linking to the system `curl` library have been upgraded to use `curl` 8.14.1. (Bug #37389565)

## Bugs Fixed

- **Important Note; Group Replication:** The Group Communication System (GCS) handles Group Replication communication between members, and keeps track of the group membership and connections between all group members. Membership tracking includes the membership's current and

previous two iterations. When a member leaves, the remaining members in the group keep a record of the departing member but stop communicating with it until it returns. For example: The group contains members M1, M2, and M3. M3 leaves the group; M1 and M2 stop communicating with M3. When a new member (M4) joins, it learns the previous iterations of this group's membership and attempts to communicate with all servers, including those from previous iterations (in this case, M3).

In the event that some previous servers were gone and did not return, the new member continuously tried to connect to the missing servers; in some conditions, these ongoing connection attempts could introduce network latency affecting group member communication.

In order to avoid this issue, connections to servers that belong to iterations of the group membership are now stopped after 5 minutes, which should be sufficient time to re-establish valid connections without imposing a continuing impact on group communication. (Bug #37704514)

- **NDB Cluster:** Following an upgrade from NDB 8.0 to NDB 8.4, all data nodes in the cluster underwent an unexpected simultaneous restart. This occurred when the transaction coordinator had no scan state, leading to protocol timeout; the resulting misalignment in protocol states caused data nodes to shut down unexpectedly. This is fixed by extending existing handling of an unexpected `SCAN_NEXTREQ` signal to cover the case when the scan is already stateless. (Bug #37994985)

  References: This issue is a regression of: Bug #37022901.

- **NDB Cluster:** Concurrent `ALTER TABLE` statements could cause delays of up to 100 * `TransactionDeadlockDetectionTimeout` before failing with a lock wait timeout when a client held a conflicting row lock, stalling the accompanying `get_commit_count()` call. The retry mechanism has been adjusted to identify locking issues sooner. (Bug #37955025)

- **InnoDB:** When rebuilding a primary key, the server sometimes encountered issues when duplicates were present, potentially leading to the server stopping unexpectedly.

  Our thanks to Xizhe Zhang and the team at Alibaba for the contribution. (Bug #37822992)

- **InnoDB:** Fixed an issue relating to dropping columns that were part of an index. (Bug #37726881)

- **InnoDB:** Fixed an issue relating to DELETE operations. (Bug #37478594)

- **InnoDB:** Creating a secondary index on a `VARCHAR` column could allocate more memory than configured, with the amount allocated being directly related to the value of `innodb_ddl_buffer_size`, leading to errors similar to `ERROR 1136 (21S01): Column count doesn't match value count at row 1`. (Bug #37233273)

- **InnoDB:** Fixed an issue relating to indexing spatial datatype columns. (Bug #36682518)

- **InnoDB:** A long semaphore wait crash could occur when a redo log consumer lagged behind after a failed MySQL Enterprise Backup incremental backup, preventing the redo log writer thread from advancing. Error messages were returned similar to the following:

  ```
  [Warning] [MY-013934] [InnoDB] Redo log writer is waiting for MEB redo
  log consumer which is currently reading LSN=23335640211468 preventing
  reclamation of subsequent portion of the redo log. Consider
  increasing innodb_redo_log_capacity.
  ```

  (Bug #36330455)

- **InnoDB:** Fixed an issue relating to range queries on tables. (Bug #31360522)

  References: See also: Bug #38063122.

- **Partitioning:** Truncating a partition was rejected with a duplicate entry error when the partition ID exceeded `INT_MAX`, preventing creation of new partitioned tables. To mitigate this issue, the `Table_partition_values_pk` class constructor now uses `ulonglong` instead of `int` for the object ID. (Bug #35912852)

- **Replication:** When using replication in a chain, `CREATE TABLE ... AS SELECT` sometimes resulted in inconsistent entries in the binary log, potentially causing replication to break on downstream servers. Error messages related to this issue included errors resulting from missing `START TRANSACTION` statements in the log. (Bug #37986380)

- **Replication:** During semisync replication, when the length of the binary log suffix exceeded six digits (`.999999`), so that the next log file became—for example—`mysql-bin.1000000`, the replication protocol unexpectedly changed from semisynchronous to asynchronous.

  Our thanks to Wuhao Cao and Karry Zhang and the team at Alibaba for the contribution. (Bug #115861, Bug #113813, Bug #37024069, Bug #36246779)

- **Group Replication:** In an unstable network environment, a Group Replication InnoDB Cluster with `group_replication_paxos_single_leader=ON` experienced several long-running transactions that became stuck in the `waiting for handler commit` state. As a consequence, `group_replication_set_as_primary()` was forced to wait, which in turn blocked other incoming queries and rendered the cluster unwritable.

  The issue involving long-running transactions stuck in `waiting for handler commit` occurred as follows: During an intermittent network partition, a secondary node incorrectly assumed the leader role due to outdated or inaccurate membership information. This resulted in conflicts in synode number allocation, causing transactions originating from the primary node to remain incomplete.

  We fix this by making sure that a secondary node always reflect the latest, accurate state before pushing the view message to Paxos. This ensures that outdated or inconsistent membership information does not lead to conflicts in leadership or synode number allocation. (Bug #37764970)

  References: See also: Bug #117424, Bug #37237959, Bug #37645674.

- **NDB Cluster APIs:** Excluding a `VARCHAR` column from an event definition resulted in an `Invalid schema object version` error. (Bug #37766391)

  References: See also: Bug #31848270.

- Some triggers which called stored routines did not always execute correctly. (Bug #37915445)

- A view using a CTE which contained a subquery was not always handled correctly. (Bug #37832605)

- `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()` did not always execute correctly. (Bug #37829550)

  References: See also: Bug #38063175.

- Improved a previous fix for an issue in which client connections were not always terminated correctly during shutdown. (Bug #37755594)

  References: This issue is a regression of: Bug #35854919.

- The Fedora 42 RPM installation packages have been adapted to accommodate the merged `/usr/bin` and `/usr/sbin` directories in Fedora 42 and later. (Bug #37737658)

- Performance of fulltext searches using `InnoDB`, particularly for phrase searches, has been improved. The efficiency of `doc_id` matching has been enhanced. (Bug #37682648)

- Queries against the `performance_schema.keyring_keys` table caused issues when the `keyring_okv` plugin (see Using the keyring_okv KMIP Plugin) was installed but not configured correctly. (Bug #37655299)

- The fix for Bug #30875669 was not actually included in the code for the `mysql` client when the bug was closed. The changes are now implemented. (Bug #37572191)

  References: See also: Bug #30875669.

- An init file having one or more single lines, each containing multiple SQL statements, sometimes gave rise to errors during initialization. (Bug #37559598)

- Some sequences of `CREATE TABLE` and `DROP TABLE` statements were not handled correctly. (Bug #37534068)

  References: This issue is a regression of: Bug #35721121.

- `UPDATE ... SET ...` could not always be rolled back successfully. (Bug #37489167)

- Multiple `DEFAULT` column expressions in `CREATE TABLE` statements were not always handled correctly. (Bug #37436220)

- Queries having a correlated subquery which performed aggregation were sometimes (incorrectly) rejected with duplicate key errors during execution.

  This issue was introduced by a previous fix which removed the restoration of the original reference slice (`set_ref_item_slice(REF_SLICE_SAVED_BASE)`) during `JOIN::cleanup()` execution under the assumption that this was not necessary. As a result, temporary table field references from previous executions were not cleared, leading to attempts to insert duplicate keys into temporary tables triggering the error `Can't write; duplicate key in table`.

  We fix this by restoring the original reference slice during cleanup, ensuring that any stale references are discarded. (Bug #37415167)

  References: See also: Bug #32141711. This issue is a regression of: Bug #35856247.

- Unquoted semicolon characters (`;`) within comments were not always flagged as errors, in spite of the fact that they are not allowed. (Bug #37117875)

  References: See also: Bug #38063286.

- A query using a nested aggregate function which contained a subquery was not always properly handled. (Bug #36421727)

- When attempting to transform a subquery to a derived table, certain cases were not always considered. (Bug #36421710)

  References: This issue is a regression of: Bug #36921175.

- With `optimizer_switch` set to `subquery_to_derived=on`, some queries with `ROLLUP` were not handled properly. (Bug #36314993)

- Setting `max_join_size` led to improper processing of some nested queries. (Bug #35625769)

- Events created within stored programs were not always handled correctly. (Bug #35395333, Bug #36402968, Bug #37918920)

  References: This issue is a regression of: Bug #17809, Bug #11745618.

- The transform of a scalar subquery into a join with a derived table where the subquery is in the `SELECT` list and the containing query is implicitly grouped should be allowed, but was rejected when the `subquery_to_derived` optimizer switch was enabled. (Bug #35150438)
- An error in `include/assert_grep.inc` could lead to erroneous results from any file that included it.

  Out thanks to Ke Yu for the contribution. (Bug #116239, Bug #37105430, Bug #37675340)
- The query rewrite plugin (see The Rewriter Query Rewrite Plugin) did not work properly when the server was run with `autocommit=OFF`. (Bug #115437, Bug #36784795)
- Improved comments in `sql/range_optimizer/rowid_ordered_retrieval_plan.cc`.

  Our thanks to Xiaodong Huang and the Tencent team for the contribution. (Bug #107972, Bug #34422274)

# Changes in MySQL 8.0.42 (2025-04-15)


- Audit Log Notes
- Compilation Notes
- SQL Function and Operator Notes
- INFORMATION_SCHEMA Notes
- InnoDB Notes
- Functionality Added or Changed
- Bugs Fixed


## Audit Log Notes

- `<COMMAND_CLASS>` was not populated for `<NAME>Execute</NAME>`.

  For more information, see Logging Specific Event Classes. (Bug #36686351)

## Compilation Notes

- **Group Replication:** The OpenSSL Engine interface is deprecated, and is no longer being included in OpenSSL v3 main packages by some Linux distributions, including Fedora.

  To avoid build issues, the usage of the OpenSSL Engine interface by the Group Communication System (GCS) is now restricted to OpenSSL versions previous to 1.1. (Bug #37475769)
- **Linux:** Use /usr/bin/gcc (GCC 14.2.1) when building the server on Oracle Linux 10. (Bug #37616148)
- Upgraded the bundled Curl library to version 8.12.1. (Bug #37633587)
- Abseil could not be built on FreeBSD. (Bug #37611924)
- In order to use `xxhash` functions independently from the `lz4` library (bundled or source), we compiled `xxhash.c` into our own binaries, which required using a great many CMake directives. Instead, we now build an interface library for `xxhash`, and link with that wherever such functions are used. (Bug #37417386)
- Use xxHash-0.8.2 from GitHub rather than the version bundled with lz4. (Bug #37387318)

## SQL Function and Operator Notes

- **Important Change:** When an SQL function is improved from one release to the next, it may throw SQL errors in situations in which it previously did not. If this happens in a table's constraints, default expressions, partitioning expressions, or virtual columns, the table could not be opened. This prevented both analyzing the problem (using, for example, `SHOW CREATE TABLE`) and addressing it (such as with an `ALTER TABLE ... DROP ...` statement).

  Now, on server upgrade, we scan the data dictionary for tables that use any of the features just mentioned. We then try to open such tables, and if we fail to do so, we alert the user. This patch addresses this. The `--check-table-functions` server option introduced in this release helps to address this problem by making it possible to specify the server's behavior when encountering an error with such a function. Set this option to `WARN` in order to log a warning for each table which the server could not open; setting it to `ABORT` also logs these warnings as `WARN`, but aborts the server upgrade if any issues were found.

  `ABORT` is the default; this enables the user to fix the issue using the older version of the server before upgrading to the new one. `WARN` flags the issues, but allows the user to continue in interactive mode while addressing the problem. (Bug #36890891)

  References: See also: Bug #37009318. This issue is a regression of: Bug #98950, Bug #98951, Bug #31031886, Bug #31031888.

## INFORMATION_SCHEMA Notes

- Fixed a performance issue in the `PROCESSLIST` table. (Bug #36778475)

## InnoDB Notes

- **InnoDB:** InnoDB now uses simplified APIs for saving and restoring level one progress during parallel scans. (Bug #37517125)

## Functionality Added or Changed

- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 3.0.16. For more information, see OpenSSL 3.0 Series Release Notes and OpenSSL Security Advisory (11th February 2025). (Bug #36033684)

- **Performance; Replication:** The data structure used in tracking binary log transaction dependencies has been changed from `Tree` to `ankerl::unordered_dense::map`, which uses approximately 60% less space, and which should thus contribute to better dependency tracking performance. (Bug #37008442, Bug #37529256)

- **InnoDB:** To improve debugging, the `buf_page_t` and `buf_block_t` structure's metadata is now printed to the error log. (Bug #35115629)

  References: See also: Bug #35115601.

- Increased the historical 1024-byte limit when printing the current query during signal handling to 1073741824 (1024 * 1024 * 1024). (Bug #37603354)

## Bugs Fixed

- **InnoDB:** Fixed a potential memory leak in several places in the innobase code. (Bug #37403052)

- **InnoDB:** Under certain circumstances, MySQL could crash during shutdown due to pages which were still fixed or dirty. Errors similar to the following were logged:

```
[ERROR] [MY-011908] [InnoDB] [FATAL] Page [page id: space=46, page number=75] still fixed or dirty
[ERROR] [MY-013183] [InnoDB] Assertion failure: buf0buf.cc:5889:ib::fatal triggered thread 139963705668608
```

(Bug #37391519)

References: See also: Bug #35115601.

- **InnoDB:** `CHECK TABLE` for spatial indexes did not verify the MBR against the geometry MBR stored in the clustered index record. This could result in incorrect behavior of spatial indexes.

  As of this release, `CHECK TABLE EXTENDED` verifies the MBR matches the MBR stored in the clustered index record. (Bug #37359538)

- **InnoDB:** Fixed an issue relating to pessimistic row update.

  Our thanks to Mengchu Shi and the team at Alibaba for the contribution. (Bug #37292404)

- **InnoDB:** The `CHECK TABLE` operation could incorrectly report corruption in spatial indexes. (Bug #37286473)

- **InnoDB:** Fixed an issue relating to InnoDB redo log recovery. (Bug #37061960)

- **InnoDB:** Fixed an issue relating to reading `index_id` values. (Bug #36993445, Bug #37709706)

- **InnoDB:** Fixed an issue relating to `lower_case_table_names`. (Bug #32288105)

- **InnoDB:** Partition table indexes were not checked when retrieving a record count while that table's definition was being altered by another client session. The record count was executed without error.

  As of this release, the index is checked to ensure it is usable when retrieving a record count. (Bug #117459, Bug #37617773)

- **InnoDB:** Refactored code related to BPR_PCUR_* positioning for restore operations. (Bug #117259, Bug #37505746)

  References: This issue is a regression of: Bug #37318367.

- **InnoDB:** Changes made to `innodb_spin_wait_delay` in MySQL 8.0.30 negatively impacted performance. (Bug #116463, Bug #37212019)

- **InnoDB:** Under certain circumstances, using `ALTER TABLE` with `INPLACE` to modify the size of a column could result in an index which exceeds the valid size limit (767 bytes). This occurred for tables with a row format of `Redundant` or `Compact` and the row format was not explicitly defined in the table creation.

  As of this release, a validation is performed and an error returned by any ALTER TABLE, INPLACE operation which will result in an invalid index size. (Bug #116353, Bug #37168132)

- **InnoDB:** Fixed a memory leak in the `Clone_persist_gtid` thread.

  Our thanks to Baolin Huang and the team at Alibaba for the contribution. (Bug #107991, Bug #34454572)

- **Partitioning:** When inserting `NOW()` into a column not part of the partition key of a partitioned table, all partitions were retrieved, and no pruning occurred. (Bug #37397306)

- **Replication:** In a source-replica setup, the replica encountered irregular failures of `UPDATE` and `DELETE` statements with `ER_KEY_NOT_FOUND` errors on the same table. (The replica's binary log and GTID records showed that the row required was committed, and had not been deleted or updated.) This

occurred on the replica when the row-matching algorithm used was `HASH_SCAN` and two rows in the same table had the same CRC32 value.

In the event of such a CRC32 collision, finding a matching CRC32 in the hash table does not guarantee that the correct row is being updated, so the algorithm iterates over any multiple entries having the same CRC32, and compares the full record for each of them in a loop; the problem occurred due to the fact that the logic to exit this loop was incorrect. This logic has now been corrected. (Bug #37462058, Bug #37731216)

- **Replication:** The `asynchronous_connection_failover_delete_source()` function did not always perform as expected in all cases. (Bug #36479088)
- **Replication:** In some cases, the `asynchronous_connection_failover_add_source()` function did not perform as expected. (Bug #36479083)
- **Replication:** In some cases, `MASTER_POS_WAIT()` did not perform as expected. (Bug #36421684, Bug #37709187)
- **Replication:** The `asynchronous_connection_failover_add_managed()` function in some cases did not produce the expected result. (Bug #34648589)
- **Replication:** When the server was under a heavy write load, the binary log position for `gtid_executed` as shown in the Performance Schema `log_status` table did not match that of the gtid shown in the binary log file.

  We fix this by increasing the scope of the lock on the `log_status` table when querying it to ensure that transactions in the commit pipeline are completed. This ensures that a query against the `log_status` table waits until `gtid_executed` has been fully updated, thereby guaranteeing consistency with its position in the binary log. (Bug #102175, Bug #32442772)
- **Group Replication:** When a secondary joined the group, it might happen that all group members started to grow the value of the column `COUNT_TRANSACTIONS_ROWS_VALIDATING` column of the Performance Schema `replication_group_member_stats` table indefinitely. This impacted memory consumption in all group members, eventually leading to thrashing if not mitigated by restarting the secondary group member that triggered the behavior, or in some cases, by restarting the whole group.

  Analysis pointed to issues with the Group Replication start operation, which checks whether there are partial transactions on the `group_replication_applier` channel from previous group participation; if any are found, this channel is stopped after applying all complete transactions and its relay logs purged, and then the channel is restarted. After this, distributed recovery is performed, applying any missing data from group members.

  The issues arose when the Group Replication pipeline operation for stopping the `group_replication_applier` channel incorrectly stopped a periodic task from the certifier module, which caused some periodic internal operations not to take place. One of these tasks was the periodic sending of the committed transactions; this omission prevented garbage collection for certification, which in turn caused a continuous increase in `COUNT_TRANSACTIONS_ROWS_VALIDATING` in the Performance Schema `replication_group_member_stats` table.

  To solve this problem, we have taken steps to ensure that the pipeline operation for stopping the `group_replication_applier` channel no longer interferes with the certifier module, which also stops spurious values from being added for `COUNT_TRANSACTIONS_ROWS_VALIDATING`. (Bug #37613510)
- **Group Replication:** When running Group Replication, some transactions may not have write sets, as with empty transactions with GTID_NEXT specified, or DDL statements. For such transactions, Group Replication cannot check conflicts; thus, it is not known whether they can be applied in parallel, and for

this reason, Group Replication follows a pessimistic approach, and runs them sequentially, potentially leading to an impact on performance.

While DDL must be applied sequentially, there is no actual reason to force such behavior for empty transactions, so this fix makes it possible for empty transactions to be applied concurrently with other nondependent transactions. (Bug #37597512, Bug #37569333)

- **Group Replication:** A group running group replication with a primary `i1` and two secondaries `i2` and `i3` started to have intermittent issues because of high memory usage on the primary. The secondaries began reporting the primary as unreachable then reachable again, and the primary began reporting the secondaries as intermittently reachable then reachable as well. Following a period of such instability, the secondaries expelled the original primary (`i1`) and elected a new one (`i2`).

  Under these conditions, queries against the `performance_schema.replication_group_members` table on the former primary (`i1`) reported `i1` as `ONLINE` and `PRIMARY`, `i2` as `ONLINE` and `SECONDARY`, and `i3` as `ONLINE` and `SECONDARY` for an extended period of time (12 hours or more) until the `mysqld` process was restarted on `i1`.

  The problems observed were found to have begun on the original primary (`i1`) when one of the secondaries was overloaded and began intermittently leaving and joining the group, its connections being dropped and recreated repeatedly on the primary server. During the reconnection process, the primary hung when trying to create the connection, thus blocking the single XCom thread. This was traced to the invocation of `SSL_connect()` on the XCom communication stack, which changed in MySQL 8.0.27 from asynchronous to synchronous form. When a node was overloaded, it might not respond to the `SSL_connect()` call, leaving the connecting end blocked indefinitely.

  To fix this, we now connect in a way that is non-blocking, and that returns in case of a timeout, leaving the retry attempts to the caller—in this specific case, the XCom thread when trying to reconnect to another node. (Bug #34348094, Bug #36047891)

  References: See also: Bug #37587252.

- The `fprintf_string()` function in `mysqldump` did not use the actual quote character for string escaping. (Bug #37607195)

- `EXPLAIN` did not always handle subqueries correctly. (Bug #37560280)

- If a demangled function name exceeded 512 bytes in a stack trace, the function name was truncated and a newline was not printed.

  As of this release, long strings, such as filenames and demangled functions, are written directly to the output. (Bug #37543598)

- `mysqldump` did not escape certain special characters properly in its output. With this fix, `mysqldump` now follows the rules as described in String Literals. (Bug #37540722, Bug #37709163)

- Some operations on tables having functional indexes were not handled properly. (Bug #37523857)

- Attempting to install an unknown component using `INSTALL COMPONENT` was not always handled correctly. (Bug #37437317)

- The Audit Log plugin did not handle errors correctly when writing JSON output.

  See MySQL Enterprise Audit, for more information. (Bug #37370439)

- An `UPDATE` subsequent to an `INSERT` affecting a table which had a `BEFORE INSERT` trigger was sometimes rejected with a null value error when the `INSERT` had set a `NOT NULL` column to `NULL`, even though this should have been allowed by the server `sql_mode` in effect. (Bug #37337527)

- In some cases, components could not reuse the same connection for running multiple queries. (Bug #37286895)
- Improved error handling for stored routines. (Bug #37193011)
- Stored routines were not always invoked correctly in prepared statements. (Bug #37077424, Bug #37292797)
- Increased the size of `SEL_ROOT::elements` from `uint16` to `size_t`. (Bug #36610878)
- Removed an issue with multibyte UTF8 handling. (Bug #36593253)
- An `ORDER BY` containing an aggregation was not always handled correctly. (Bug #36593244)
- An optimizer hint was ignored, unexpectedly requiring the use of `FORCE INDEX`, when querying a view that included a `UNION`. For more information, see Optimizer Hints. (Bug #36536936)
- Some subselects were not handled correctly. (Bug #36421690)
- An invalid DDL statement in certain cases was not always rejected as expected. (Bug #35721121)
- Improved the internal function `append_identifier()`. (Bug #35633084)
- Normally, a view with an unused window definition should be updatable, but when it contained a subquery, it was marked as not updatable. At update time, the window was eliminated, but this was too late to allow an update to be performed.

  We fix this by testing mergeability, by checking the presence of window functions, rather than that of window definitions; this allows the view to be updatable, and the problematic `UPDATE` to succeed. (Bug #35507777)
- In some cases, `SET` did not perform correctly in prepared statements. (Bug #35308309)
- This fix addresses the following issues:
  - `Query_expression::is_set_operation()` was not always executed properly.
  - Some sequences of DML statements could lead to an unplanned exit.
  - Some nested subselects were not always handled correctly.

  (Bug #34361287, Bug #35889583, Bug #35996409, Bug #36404149, Bug #36863048, Bug #37611264)
- On Debian, `dh_strip_nondeterminism` is no longer executed on zip and gzip files within the packages. (Bug #33791880)
- Removed an issue relating to invalid UTF8 values. (Bug #27618273, Bug #37709687)
- Addressed an issue relating to an invalid identifier. (Bug #22958632, Bug #37709664)
- A negative impact in performance was observed when using a multivalued index with `ORDER BY DESC` and `LIMIT` in a query, where the value specified by `LIMIT` was greater than the number of rows actually in the result. (Bug #117085, Bug #37436310)

  References: This issue is a regression of: Bug #104897, Bug #33334911.
- When pushing a condition down to a derived table, we clone the condition, and, if the underlying field is a view reference (that is, a field from a merged derived table), we strip off the view reference and clone the expression that it references. When the underlying expression is a constant expression from a table that is on the inner side of an outer join, it cannot be treated as a normal constant because of the need

to generate `NULL` values. When we stripped off the view reference, this information was lost, leading to wrong results.

We fix this by avoiding condition pushdown for such cases. (Bug #111355, Bug #35634714)

# Changes in MySQL 8.0.41 (2025-01-21)


- Account Management Notes
- Character Set Support
- Compilation Notes
- Component Notes
- Firewall Notes
- Optimizer Notes
- Performance Schema Notes
- Functionality Added or Changed
- Bugs Fixed


## Account Management Notes

- The database cache was not flushed properly following execution of `DROP USER`. (Bug #37132323)
- Failed password validation was not always handled correctly. (Bug #37041439)

## Character Set Support

- Zero rows resulted (where one row was expected) when selecting from a view created with its connection and client character set to `latin1`, where a query on the view used `utf8` as its connection and client character set, the view contained literal values with non-ASCII characters, and the query performed a condition pushdown into a `UNION` of query blocks of the view.

  This problem was related to a previous issue which fixed an error for a similar problem: The problem in that case was properly considering the character set of the view definition when pushing down conditions contained in the view to inner query blocks, but the fix implemented at that time did not take into account the possibility that the view might contain non-ASCII characters.

  This meant that the condition to be pushed down was written to a text string with the wrong character set. We fix this oversight by ensuring that the string is created with the correct character set. (Bug #37111452)

  References: See also: Bug #36246859.

## Compilation Notes

- **macOS:** Removed obsolete CMake code from MacOS builds. (Bug #37258036)
- **macOS:** Removed warnings of the form `ld: warning: ignoring duplicate libraries` and warnings specific to `xcodebuild`. (Bug #37065301)
- **Solaris:** The minimum required version of GCC to build MySQL on Solaris has been raised to 11.4. See Source Installation Prerequisites, for more information. (Bug #37256600)

- Starting with CMake 3.26, CMake writes the file `CMakeFiles/CMakeConfigureLog.yaml`, which supersedes `CMakeError.log`. References to `CMakeError.log` have therefore been removed. (Bug #37305289)
- The version of `libedit` used to compile MySQL was upgraded to 20240808-3.1. (Bug #37101293)
- Removed an error in `mysql_prepare_create_table()` (in the file `sql/sql_table.cc`) found when compiling MySQL with XCode 16. (Bug #37068527)
- The server could not be built on Fedora 40 (and possibly other Linux platforms) using cmake 3.11, due to an issue with TIRPC. (Bug #116164, Bug #37080195)

## Component Notes

- `INSTALL COMPONENT` issued concurrently with a `SET PERSIST` which used a subquery could sometimes lead to an unplanned exit of the server. (Bug #36559078)

  References: See also: Bug #35647759.

## Firewall Notes

- In some cases, after performing an upgrade, stored procedures relating to MySQL Enterprise Firewall were not processed correctly. (Bug #36084822)

## Optimizer Notes

- Pushing down a condition which had an aggregate function in a `WHERE` clause caused the aggregate function to be evaluated when it should not have been. (Bug #36421735)

## Performance Schema Notes

- If a user other than root ran `START REPLICA`, `PERFORMANCE_SCHEMA.PROCESSLIST` assigned that user's name to the newly spawned foreground replication threads instead of `system user`.

  As of this release, `system user` is assigned to all foreground system threads. (Bug #37357562)
- Under certain circumstances, a metadata lock can be upgraded or downgraded to a different `LOCK_TYPE`. This change was not reflected in the `PERFORMANCE_SCHEMA.METADATA_LOCKS` table.

  Our thanks to George Ma and the Alibaba team for the contribution. (Bug #116625, Bug #37271768)

## Functionality Added or Changed

- Binary packages that include `curl` rather than linking to the system `curl` library have been upgraded to use `curl` 8.11.1. (Bug #37389565)

## Bugs Fixed

- **Incompatible Change:** Corruption occurred in a spatial index when an update of a geometry with a minimal change in the minimum bounding rectangle (MBR) was followed by a delete operation.

  When upgrading to this release, it is recommended that you drop any spatial indexes beforehand, then re-create them after the upgrade is complete. Alternatively, you can drop and re-create such indexes immediately following the upgrade, but before making use of any of the tables in which they occur. You should also be aware that downgrading to any previous version reintroduces the original problem described previously.

For more information, see Creating Spatial Indexes. (Bug #36452528)

- **InnoDB:** Concurrently truncating a table while querying the Performance Schema sometimes cause MySQL to halt unexpectedly. (Bug #37271715)
- **InnoDB:** It was possible for an `ALTER TABLE` operation using the `INPLACE` algorithm on a table containing both a spatial index and an auto-increment column to cause corruption or, in debug builds, to trigger a debug assert. This was due to the auto-increment column value being overwritten in the old records of the spatial index while the new record was prepared. (Bug #37189985)
- **InnoDB:** Certain IO buffer serializations triggered an assertion in debug builds that caused the system to hang. (Bug #37139618)
- **InnoDB:** Improved `InnoDB` start up time. (Bug #36880863)

  References: This issue is a regression of: Bug #36808732.
- **InnoDB:** An assertion failure was raised when creating a `FULLTEXT` index on a table with an `FTS_DOC_ID` value greater than 4294967295. (Bug #36879147)

  References: See also: Bug #37387224.
- **InnoDB:** Dropping a primary key, and then adding a new `AUTO_INCREMENT` column as a primary key in descending order using the `INPLACE` algorithm failed.

  Our thanks to Shaohua Wang and the team at Alibaba for the contribution. (Bug #36658450)
- **InnoDB:** Extending a user tablespace produces file extension redo log records (`MLOG_FILE_EXTEND`), but they were not produced when extending the system tablespace. (Bug #36511673)
- **InnoDB:** A `DELETE` operation on a table with a self referential foreign key and full-text index could have triggered an assertion. (Bug #36234681)
- **InnoDB:** When clearing an AHI index from all buffer pool pages, the block state would potentially change to BUF_BLOCK_MEMORY before acquiring the block mutex thus causing an unexpected halt. (Bug #35037114)
- **InnoDB:** Common prefix compression for redo log inserts (`MLOG_REC_INSERT`) was disabled but is now enabled when the versions match. (Bug #34946626)

  References: This issue is a regression of: Bug #13899.
- **InnoDB:** Virtual column information for a row containing an externally stored `BLOB` was not always logged during an `UPDATE` operation, which sometimes resulted in an `Index PRIMARY is corrupted` error. (Bug #34574604)
- **InnoDB:** `ON DELETE CASCADE` with generated columns containing secondary indexes sometimes failed, due to virtual column templates not being initialized before deletion.

  Our thanks to Rahul Malik for the contribution. (Bug #33691659)
- **InnoDB:** An update operation attempted to update a virtual column while building an update node for a child table, but should not have done so since foreign key constraints cannot reference virtual columns. (Bug #33327093)
- **InnoDB:** It was possible for `ALTER TABLE`, which rebuilds `InnoDB` tables using the `INPLACE` algorithm, to be rejected with a duplicate key error due to a non-duplicate record being inserted concurrently while the rebuild was paused to release a page latch.

Our thanks to Dmitry Lenev and the team at Percona for contributing to this fix. (Bug #115511, Bug #36808088)

- **InnoDB:** The check enforcing the rule that `ALGORITHM=INSTANT` cannot be used on a column referenced by a foreign key constraint from another table did not inspect the last field of said constraint. (Bug #115457, Bug #36796094)

- **InnoDB:** CPU usage statistics did not account for a processor count over 128, which could degrade performance on these larger systems. (Bug #115399, Bug #36765223)

- **InnoDB:** Executing `ALTER TABLE` with `ADD COLUMN` or `DROP COLUMN` against an empty table now uses the `INPLACE` algorithm by default instead of `INSTANT`. This change means the row version is no longer incremented for these simple operations. (Bug #113051, Bug #36004394)

- **InnoDB:** An `ALTER TABLE` operation that rebuilt an `InnoDB` table using the `INPLACE` algorithm potentially led to losing a row of data if a purge occurred concurrently on the altered table that contained a clustered or spatial index.

  Our thanks to Dmitry Lenev and the team at Percona for contributing to this fix. (Bug #110706, Bug #113812, Bug #115608, Bug #116764, Bug #35303494, Bug #36261480, Bug #36846567, Bug #37318367)

- **InnoDB:** Queries with a descending primary key and the `index_merge` optimization sometimes yielded incorrect results such as missing rows. (Bug #106207, Bug #33767814)

- **Replication:** In an InnoDB ClusterSet setup, when `autocommit` was set to `OFF` on all nodes in the cluster, a controlled switchover using MySQL Shell was rejected with Error 1105 (`Unknown error`).

  To fix this, we now force a new transaction in `channel_change_source_connection_auto_failover()` whenever `autocommit=OFF` to prevent table access deadlocks when an info repository transaction is executed after changing `SOURCE_CONNECTION_AUTO_FAILOVER`. (Bug #37173907)

- **Replication:** While large transactions were being received and applied, and a request to stop the replication channel was made using `STOP REPLICA`, MySQL did not do so properly, and subsequently did not process any channel commands. In addition, the server shutdown process did not complete gracefully, and required either the MySQL process to be killed or the host system to be restarted. (Bug #115966, Bug #37008345)

- **Replication:** The log message written when a replica reconnects to the source (when, for example, it is stopped and restarted by issuing `STOP REPLICA` followed by `START REPLICA`) `While initializing dump thread for replica with UUID` *`uuid`*`, found a zombie dump thread with the same UUID. Source is killing the zombie dump thread(`*`thread_id`*`)` has been improved to `Upon reconnection with the replica, while initializing the dump thread for UUID` *`uuid`*`, an existing dump thread with the same UUID was detected. The source is terminating the previous dump thread (`*`thread_id`*`), which is normal and expected`. (Bug #84358, Bug #25330090)

- **Group Replication:** Removed a potential race condition between the internal functions `cs::apply::Commit_order_queue::front()` and `cs::apply::Commit_order_queue::remove()`. (Bug #37223451)

  References: See also: Bug #35206392.

- **Group Replication:** When the primary node unexpectedly left the group and quickly attempted to rejoin, the member which had been elected to remove other, faulty members tried to expel or remove the faulty

node but could not do so due to lack of a majority. When the expelled or removed node was the primary, this left the cluster without a primary, resulting in an unusable state. (Bug #36991859)

References: See also: Bug #37181867.

- **Group Replication:** In some cases, adding a new secondary caused existing secondaries to lag, leading to a deadlock which persisted with no more writes possible until the primary was restarted.

  This deadlock occurred between the ticket manager, which ensures that transactions are committed on the correct side of a view change (before or after the view change), and the commit order manager on the inbound replication channel, which ensures that transactions are committed in the same order in which they are received, when these two managers required different orders. This meant that, in some cases, adding a new secondary caused the group primary to be unable to do writes.

  We solve this issue by ignoring the commit order manager ordering and enforcing the ticket manager ordering for non-conflicting transactions when such a deadlock occurs. A consequence of this is that `replica_preserve_commit_order` may not be strictly honored near a `View_change_log_event`. In other words, `replica_preserve_commit_order` no longer provides a strict guarantee on an inbound channel on a Group Replication primary. `replica_preserve_commit_order` still ensures that transactions are ordered correctly, with the only exception being non-conflicting transactions around view change log events. (Bug #35206392)

  References: See also: Bug #37223451.

- **Group Replication:** Improved garbage collection in Group Replication by eliminating non-essential calls to `is_subset_not_equals()`. (Bug #110673, Bug #35286974)

- **Group Replication:** Removing a group member from a group in which all members were running the same version of MySQL, upgrading it to a later version (from a later release series), and then requesting it to rejoin the group caused the upgraded group member to hang in the recovering state.

- The errors `ER_DD_UPDATE_DATADIR_FLAG_FAIL`, `ER_IB_MSG_FIL_STATE_MOVED_PREV_OR_HAS_DATADIR`, `ER_RPL_KILL_OLD_DUMP_THREAD_ENCOUNTERED`, and `ER_RPL_MTA_ALLOW_COMMIT_OUT_OF_ORDER` were originally defined in MySQL 8.0, but were subsequently assigned different error code numbers (but with the same names) in MySQL 8.4. The numbers assigned in MySQL 8.0 now apply to MySQL 8.0 only; in the MySQL 8.4 and later release series, only the numbers assigned in MySQL 8.4 are used. (Bug #37284176)

- In `sql/item_cmpfunc.cc`, `Item_bool_func2::resolve_type()` made an unchecked call to `Item_bool_func::resolve_type()`; the call to `Item_bool_func::resolve_type()` ignored its return value, and execution continued even in case of an error. (Bug #37143289)

- AppArmor denied access to `/proc/$pid/task/$thread_id/mem`, a file required to generate a stack trace. (Bug #37063288)

  References: See also: Bug #37387034.

- A filter condition in a subquery was sometimes ignored when a query used the `index_subquery` join type for subquery execution, and the subquery table used materialization in the execution plan. The derived table access path replaced the filter condition, resulting a final plan without the filter layer. To fix this, in such cases, we now add the derived table access path along with the filter access path instead of replacing the latter. (Bug #36918913)

- Some `UNION` operations similar to `a UNION b UNION c ...` consumed excessive memory. To help keep this from happening, we now flatten equal set operations at the parsing level, before

contextualization occurs, which should result in reduced resource usage by such operations. (Bug #36652610)

- Improved the internal function `my_print_help()`. (Bug #36615714)

  References: See also: Bug #37387224.

- Removed incorrect code from `Acl_cache`. (Bug #36608160)

- A subquery containing an aggregate function `WITH ROLLUP` which was part of a row value comparator was not always processed correctly. (Bug #36593235)

  References: See also: Bug #37387180. This issue is a regression of: Bug #30969045, Bug #30921780, Bug #26227613, Bug #29134467, Bug #30967158.

- It was possible for errors raised when persisting variables not to be reported correctly. (Bug #36574732)

- Some subqueries using `WITH ROLLUP` were not always processed correctly. (Bug #36421704)

- Improved the output from `my_print_stacktrace()`, which prints the stack trace when MySQL terminates unexpectedly. (Bug #34904177, Bug #36027494)

- Fixed an issue relating to FTS and concurrent DDL or DML. (Bug #34633727)

- `DROP VIEW` *`name`* was rejected with `ER_BAD_TABLE_ERROR` if there existed a table with the same *`name`*. (Bug #33200087)

- Incorrect results were returned by some queries that used hash antijoins when the hash table did not fit in the join buffer and spilled to disk. (The query triggering the issue actually specified `LEFT JOIN`, but this was transformed internally from a left outer join to an antijoin.)

  The problem was that some rows in the probe table were skipped when writing the probe rows to chunk files, the skipped rows being those that had `NULL` in part of the join key. Such rows can be skipped for inner joins and semijoins, as they are known to have no match in the build table, but for outer joins and antijoins, rows in the probe table which have no matching row in the build table should be part of the join result, so they must be included in the chunk files.

  We already preserved these rows in the chunk files for outer joins. This fix extends the logic used for that purpose so that it also applies for antijoins. (Bug #116334, Bug #37161583)

- In MySQL 8.0 and later, queries of the form `SELECT DISTINCT ... FROM t1 WHERE NOT IN(SELECT ...)` were transformed into an antijoin if possible, causing the optimizer not to choose a group skip scan for table t1 whereas it would have been chosen in MySQL 5.7. This resulted in a performance degradation for such queries. Group skip scan is not chosen, since the query is now no longer a single-table query following the antijoin transformation, and this access method is enabled only for single table queries. The same behavior can be seen for queries which are transformed into semijoins as well. In such cases, group skip scan access method can still be used if the access method is used only for duplicate removal (that is, with `DISTINCT` or `GROUP BY`, but without aggregate functions).

  To fix this, we enable group skip scan when there is only one table in the original query, irrespective of the number of semijoin tables present after internal transformations as long as the query contains no aggregate functions. (Bug #112362, Bug #35842412)

- The `mysql` client did not allow using '#' or '--' inside an optimizer hint comment.

  Our thanks to Kaiwang Chen for the contribution. (Bug #98521, Bug #30875669)

# Changes in MySQL 8.0.40 (2024-10-15)


- C API Notes
- Compilation Notes
- SQL Function and Operator Notes
- Performance Schema Notes
- sys Schema Notes
- Thread Pool Notes
- Functionality Added or Changed
- Bugs Fixed


## C API Notes

- The asynchronous interface used unsafe static local variables to store state information. (Bug #115703, Bug #36891894)

## Compilation Notes

- MySQL did not compile on Fedora 41. (Bug #37046924)
- MySQL did not compile on Ubuntu 24.10. (Bug #37042308)
- Added `CONTRIBUTING.md` and `SECURITY.md` files to the MySQL sources to conform to Oracle's Open Source guidelines. (Bug #36998165)
- Aligned `CMAKE_MINIMUM_REQUIRED` with the correct required CMake version (3.14.6) as well as CMake policies for third-party libraries used by MySQL. (Bug #36978193)
- Binary packages that include curl rather than linking to the system curl library have been upgraded to use curl 8.9.1. Important issues fixed in curl version 8.9.1 are described at https://curl.se/docs/security.html. (Bug #36967379, Bug #36955197)
- The included `zlib` library has been upgraded from version 1.2.13 to version 1.3.1. (Bug #36950863)
- CMake options `BUILD_SHARED_LIBS` and `CMAKE_SKIP_INSTALL_ALL_DEPENDENCY` are now set to `OFF` in the top level of the build. (Bug #36930664)
- Upgraded the bundled `libcurl` library to version 8.9.0. (Bug #36886877)
- The bundled `lz4` library was upgraded to version 1.10.0. (Bug #36886747)
- The `-DWITHOUT_SERVER` option for CMake did not work on Enterprise Linux 7. (Bug #36824515)
- The version of `libfido` used with MySQL was upgraded to 1.15.0. (Bug #36752604)
- The version of `clang-format` used for the MySQL codebase was upgraded from 10 to 15. (Bug #36500268)
- The bundled version of TI-RPC was upgraded to 1.3.5. (Bug #115698, Bug #36886602)
- It was not possible to build MySQL using Protobuf 22 or later.

Our thanks to Gordon Wang for the contribution. (Bug #115163, Bug #36678092)

## SQL Function and Operator Notes

- The output from `DATABASE()` was truncated when this function was used as part of a `UNION` query. (Bug #36871927)

## Performance Schema Notes

- Redesigned the performance schema `data_locks` and `data_lock_waits` tables so that querying them does not require an exclusive global mutex on the transaction or lock system. It now iterates over buckets of hash tables that hold the locks to only latch the actively processed shard, when previously it iterated over the transactions. This also improves the iteration logic complexity in terms of speed and memory to decrease the impact of these queries on the rest of the system.

  Note that the query result might show an incomplete list of transaction locks if it committed, started, or otherwise changed the set of owned locks in-between visiting two buckets. This differs from previous behavior which always showed a consistent snapshot of locks held by individual transactions, although two different transactions could have been presented at different moments. In other words, the new approach gives a consistent view of a single wait queue to show conflicting locks with a waiting lock because they are always in the same bucket, while the old approach could miss some of them because they belonged to other transactions. The old approach would always show all the other locks held by a reported transaction but could miss locks of other transactions even if they were conflicting. (Bug #112035, Bug #113761, Bug #36302624, Bug #36015586, Bug #36690035, Bug #35712638, Bug #36891888)

- Running `SELECT * FROM sys.innodb_lock_waits;` on an instance which was under heavy load affected the performance of the server.

  As of this release, `SELECT * FROM sys.innodb_lock_waits;` fetches only 2 locks for each wait, instead of scanning all locks twice for each wait.

  As part of this fix, primary keys were added to `DATA_LOCKS` and `DATA_LOCK_WAITS`. (Bug #100537, Bug #31763497)

## sys Schema Notes

- The performance of the `innodb_lock_waits` view is improved in this release. (Bug #36337708)

## Thread Pool Notes

- Some concurrent connection attempts were not handled correctly. (Bug #36625082)

## Functionality Added or Changed

- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 3.0.15. For more information, see OpenSSL 3.0 Series Release Notes and OpenSSL Security Advisory [3rd September 2024]. (Bug #37021075)

- Added the `--system-command` option for the `mysql` client, which enables or disables the `system` client command.

  This option is enabled by default. To disable it, use `--system-command=OFF` or `--skip-system-command`, which causes the `system` command to be rejected with an error. (Bug #36377685, WL #16482)

References: See also: Bug #36248967.

## Bugs Fixed

- **Performance; InnoDB:** Several functions internal to `InnoDB`, which were defined as inline in MySQL 8.0.28, were found to be no longer inline in MySQL 8.0.33, due in part to refactoring which accompanied improvements made in MySQL 8.0.30 to improve the `InnoDB` adaptive hash index. This had an adverse effect on queries using joins on `InnoDB` tables. (Bug #111538, Bug #35531293)

  References: This issue is a regression of: Bug #81814, Bug #16739204, Bug #23584861.

- **InnoDB:** InnoDB did not allow updating a `REDUNDANT` table column that was altered with `NULL` as the default value using the `INSTANT` algorithm. Instead, MySQL unexpectedly halted. (Bug #36840107)

- **InnoDB:** When redo log capacity was reduced with `log_files_truncate`, under rare circumstances the `file.end_lsn` equaled `log_sys->write_lsn` thus disallowing future redo log writes. (Bug #36730830)

- **InnoDB:** It was possible for the MySQL server to halt unexpectedly when executing a `DELETE` or `UPDATE` statement after a column was dropped using the `INSTANT` algorithm. (Bug #36723117)

- **InnoDB:** FTS index optimization would sometimes not function correctly with tokens equal in collation order but different in terms of bytes. (Bug #36652127)

- **InnoDB:** Replaced `std::this_thread::sleep_for(std::chrono::seconds(0))` usage with `std::this_thread::yield()` to prevent it from being optimized out while compiled. (Bug #36522343)

- **InnoDB:** The upgrade process unexpectedly halted when the database contained a full-text index created in MySQL 5.7 with a user-created `FTS_DOC_ID` column as the primary key.

  Our thanks to Huaxiong Song and the team at Alibaba for contributing to this fix. (Bug #36496164)

- **InnoDB:** Disabled the optimizer for indexes with range conditions on multiple-value virtual columns. (Bug #36341532)

- **InnoDB:** On Windows, fixed a doublewrite buffer regression that slowed file access, and refactored `FILE_FLAG_OVERLAPPED` flag usage for opening files. (Bug #36259487)

- **InnoDB:** A schema mismatch was possible when a table was imported with a different `sql_mode` than when it was created, because strict mode influences the number of nullable columns in a table's secondary index. Now the secondary index trees are also checked for corruption after a tablespace is imported. (Bug #35799038)

- **InnoDB:** Given a table with an FTS index, when its tablespace is discarded the corresponding FTS tables are also dropped. When performing an operation that cleared out these tables, the code incorrectly assumed that the FTS tables must also exist if the FTS indexes were present in the table metadata. (Bug #35343458)

- **InnoDB:** An uninitialized buffer was being written to a temporary file when checking if the system supports a different sector size for FusionIO. This check is made when `innodb_flush_method` is set to `O_DIRECT` or `O_DIRECT_NO_FSYNC`. (Bug #115229, Bug #36705034)

- **InnoDB:** Table rebuild operations involving secondary indexes required greater file I/O for `InnoDB` temporary files as compared with MySQL 8.0.26, which degraded query performance. (Bug #114465, Bug #36444172)

- **InnoDB:** Adding indexes with the parallel index builder was much slower with the Performance Schema enabled due to issues in the `Alter_stage` class. (Bug #113505, Bug #36163502)

- **InnoDB:** Fixed an `innodb.parallel_read_kill` related unit test to better account for recent optimizer changes, and fixed an `innodb.ddl_kill` unit test and its associated macro usage.

  Our thanks to Dmitry Lenev and the team at Percona for contributing to these fixes. (Bug #113002, Bug #115416, Bug #35992036, Bug #36764973)

  References: See also: Bug #112767.

- **Partitioning:** `ALTER TABLE` did not always work correctly with partitioned tables. (Bug #36677952)

- **Group Replication:** Under certain conditions, all secondaries shut down unexpectedly following a brief period of network inactivity on the primary host. (Bug #35642087)

  References: See also: Bug #32673639, Bug #34565855.

- **Group Replication:** Memory aggregated by threads as reported did not account for all `memory/sql/Gtid_set::Interval_chunk` freed by other threads from what was allocated, leading to a incorrect ever-increasing consumption of resources by Group Replication thread `thread/group_rpl/THD_applier_module_receiver`.

  Global memory as reported by `performance_schema.memory_summary_global_by_event_name` was not affected by this issue. (Bug #34819861)

- **Group Replication:** Removed a potential memory leak from `xcom\network\xcom_network_provider_native_lib.cc`. (Bug #115162, Bug #36673883)

- The server did not always handle connections correctly when running with both the thread pool and audit log plugins. (Bug #37039422)

- Packages for Debian-based systems are now built with `WITH_ZLIB=system`, and disable `MYSQL_MAINTAINER_MODE` for debug builds. (Bug #37038213)

- Updated the Kerberos library bundled with commercial builds to version 1.21.3. For more information, see the Kerberos 5 1.21.x Release Notes. (Bug #37034600)

- Removed a heap-use-after-free warning in `regexp::Regexp_facade::~Regexp_facade()`. (Bug #36867806)

- While dumping tablespaces, `mysqldump` did not properly escape certain SQL statements in its output. In addition, the dump now encloses the following identifiers within backticks: `LOGFILE GROUP`, `TABLESPACE`, and `ENGINE`. (Bug #36816986)

- A previous fix for an issue in `sql/sql_executor.cc` checked for `const_item()` rather than `const_for_execution()`. (Bug #36804785)

  References: This issue is a regression of: Bug #34951115.

- When executing an index range scan using `IndexRangeScanIterator` the record buffer was never set due to the fact that the data type used for `IndexRangeScanIterator::m_expected_rows` had been changed from double to boolean during refactoring. This unintended change has been reverted.

  In addition, when the record buffer was enabled for index range scans a problem arose with multi-valued indexes used for covering these scans. (This is possible only when they are used as part of index merge scans, since index merge scans force covering index scans.) The source of the problem was that the

implementation of `Field_typed_array::key_cmp()` needs the value of the generated column for the indexed expression, and this column is not available in the multi-valued index, so the storage engine cannot safely evaluate the end range condition when filling the record buffer for a covering scan. To fix this, we now disable the record buffer when multi-valued indexes are used for covering index range scans. (Bug #36775910)

References: See also: Bug #36341532.

- Certain triggers and stored procedures were not handled correctly. (Bug #36775910)
- Some errors raised when referencing external tables were not handled correctly. (Bug #36758378)
- On Debian, stopped stripping `libmysqlclient.a` to increase LTO build compatibility. (Bug #36737581)
- The internal function `my_convert_internal()` sometimes returned a pointer rather than the intended value. (Bug #36684463)

  References: This issue is a regression of: Bug #36479091.
- Removed a memory issue in the server component. (Bug #36600205)
- Fixed a memory leak in the `mysql` client. (Bug #36600203)
- The `AES_ENCRYPT()` function did not always return a valid result. (Bug #36593265)
- In some cases a join involving a very large number of rows and many tables was not processed correctly. (Bug #36562979)
- `DROP INDEX` with the addition of a `FULLTEXT` index in the same transaction sometimes led to an unplanned server exit. (Bug #36559642)
- Some combinations of optimizer hints did not function correctly. (Bug #36492114)
- Following the fix for a previous issue, a `const` item that is part of a `GROUP BY` and is not found in the select list is not added as a hidden item to the field list, but this was not taken into consideration while checking for replacements for expressions involving `ROLLUP` operations while creating a temporary table. We fix this by checking whether the item that is not found in the field list is a `const` item, and if so, we use the same item as a replacement. (Bug #36444257)

  References: This issue is a regression of: Bug #34951115.
- Removed assertions found during testing of the data masking functions `gen_rnd_pan()` and `mask_ssn()`. (Bug #36397869, Bug #36398272)

  References: See also: Bug #36398221.
- Fixed an issue relating to FTS comparisons.

  Our thanks to Shaohua Wang and the team at Alibaba for the contribution. (Bug #36210202)
- Some `CREATE TABLE ... SELECT` statements were not always processed correctly. (Bug #36138460)
- RelWithDebInfo builds are no longer compiled with the "-g1" flag, thus increasing the available debugging information and generated file sizes. (Bug #36111629)

  References: See also: Bug #33664929.

- Corrected potential misbehavior when the server was run with `--skip-grant-tables`. (Bug #36043213)
- Fixed a regression in an earlier fix for a problem with references to columns from tables of outer query blocks in the `ON` condition of a join. (Bug #35854686)

  References: This issue is a regression of: Bug #96946, Bug #30350696.
- In certain rare cases, a `CREATE TABLE` statement involving the creation of a table with foreign keys was not processed correctly. (Bug #35553557, Bug #36350852)
- Fixed an issue relating to `SHOW INDEX` and generated columns. (Bug #35497623)
- In some cases, table DDL in prepared statements was not handled correctly. (Bug #35221658)
- Changed the SLES 15 target platform from openSUSE 15.5 to 15.6, which also means using GCC 13 instead of GCC 12 and building against the system's OpenSSL 3.x instead of OpenSSL 1.1.x. (Bug #115535, Bug #36934913)
- Fixed a typo in the Debian package description.

  Our thanks to Henning Pöttker for the contribution. (Bug #115363, Bug #36749142)
- With `prefer_ordering_index=off`, a query with no reference key reverted to scanning and sorting the full table even though it should have been possible to avoid the sort.

  Our thanks to Daniel Nichter for the contribution. (Bug #113699, Bug #36213938)
- A query using a greater-than (`>`) or less-than (`<`) comparison with a multi-valued index executed much more slowly than the same query using an equality (`=`) comparison with the same index. (Bug #104897, Bug #33334911)

# Changes in MySQL 8.0.39 (2024-07-23)

## Bugs Fixed

- **InnoDB:** In some cases, following the creation of a very large number of tables (8001 or more), the server could not be restarted successfully. (Bug #36808732)

  References: This issue is a regression of: Bug #33398681.
- **InnoDB:** Improved tablespace file scan performance at startup. (Bug #110402, Bug #35200385)
- **Group Replication:** Running a `CREATE TABLE ... SELECT` statement on a source coming from an asynchronous channel to Group Replication led to errors on the replica. (Bug #36784284)

# Changes in MySQL 8.0.38 (2024-07-01)

**Important**

This release is no longer available for download. It was removed due to a critical issue that could stop the server from restarting following the creation of a very large number of tables (8001 or more). Please upgrade to MySQL 8.0.39 instead.

- Audit Log Notes

- C API Notes
- Compilation Notes
- Connection Management Notes
- Data Dictionary Notes
- Performance Schema Notes
- Pluggable Authentication
- Thread Pool Notes
- Bugs Fixed


## Audit Log Notes

- Audit log pruning did not function after removing or renaming a file from the audit log. Now pruning continues in such cases, but with a warning printed in the error log stating that it was not possible to delete the missing audit log file. (Bug #35902913)

## C API Notes

- C API applications stalled while receiving results for server side prepared statements.

## Compilation Notes

- Upgraded the bundled `googletest` and `googlemock` sources to version 1.14.0. (Bug #36562482)
- Added a missing dependency on `GenError`. (Bug #36551721)
- It is now possible on Linux systems to build MySQL using a bundled `tcmalloc` library that is provided with the source by specifying `-DWITH_TCMALLOC=BUNDLED`. This is supported on Linux only. (Bug #36313839)
- The bundled `tcmalloc()` is now used when building MySQL on Enterprise Linux 8. (Bug #114844, Bug #35674008)
- Linux `aarch64` platform binaries are now built using `patchelf --page-size=65536` for compatibility with systems using either 4k or 64k for the page size. (Bug #114233, Bug #36393794)

## Connection Management Notes

- The `conn_delay/Waiting in connection_control plugin` stage was not reset after a delay introduced by the connection control plugin which resulted in incorrect monitoring information. (Bug #35205358)

## Data Dictionary Notes

- Attempting to upgrade a `MyISAM` table containing a mix of regular columns and generated columns from MySQL 5.7 to 8.0 or later led to table corruption. (Bug #105301, Bug #33503328)

## Performance Schema Notes

- Under certain conditions, a race condition could result in the amount of RAM used by `TABLE_HANDLES` increasing to a maximum of 9GB. (Bug #36170903)

- The `PROCESSLIST_INFO` column of `THREADS` was not updated when executing a prepared statement.

  Thanks to Daniel Lenski and Amazon for the contribution. (Bug #104121, Bug #33057164)

## Pluggable Authentication

- The deprecation warning issued when authenticating with the `mysql_native_password` plugin is now issued only once. (Bug #35792948)

## Thread Pool Notes

- Connecting to a thread group that had no connection handler threads stalled. We fix this by making sure that connection handler threads terminate only if there is at least one connection thread left. (Bug #36550125)

## Bugs Fixed

- **InnoDB:** MySQL unexpectedly halted on an `UPDATE` after an `ALTER TABLE` operation. (Bug #36571091)

  References: This issue is a regression of: Bug #35183686.

- **InnoDB:** The log index size calculation now accounts for column order changes. (Bug #36526369)

  References: This issue is a regression of: Bug #35183686.

- **InnoDB:** File system operations performed by `InnoDB` now consistently `fsync` the parent directory when performing directory altering tasks. (Bug #36174938)

- **InnoDB:** In debug builds, setting the `innodb_interpreter_output` debug variable would cause the server to unexpectedly halt. This is now a read-only variable. (Bug #36041032)

- **InnoDB:** For tables created with an index on a column that was too wide for the redundant row format (allowed before MySQL 5.7.35), an in-place upgrade silently imported the table but it was not accessible, which interfered with making backups. Now all operations that involve using the invalid index are rejected with `ER_INDEX_CORRUPT` until the index is dropped. An `ER_IB_INDEX_PART_TOO_LONG` error is also reported in the error log. (Bug #35869747)

  References: See also: Bug #34826861.

- **InnoDB:** An `InnoDB` assertion error referencing an invalid column index was triggered when the column index was valid. (Bug #34800754)

- **InnoDB:** With an empty `XA` transaction, shutting the server down after an `XA START` would cause the server to halt unexpectedly. (Bug #32416819)

- **InnoDB:** Shutting down the replication applier or binlog applier while processing an empty XA transaction caused the system to unexpectedly halt. (Bug #32416819)

- **InnoDB:** Removed unnecessary heap usage in the `Validate_files::check()` function.

  Our thanks to Huaxiong Song for the contribution. (Bug #115041, Bug #36626203)

- **InnoDB:** If a partition table was read with `innodb_parallel_read_threads=1`, read performance greatly decreased from any table after 256 reads. InnoDB behaved as if it reached the maximum capacity of parallel read threads despite not using any.

Our thanks to Ke Yu for the contribution. (Bug #114154, Bug #36347408)

- **InnoDB:** The result from a spatial index containing a column with a spatial reference identifier (SRID) attribute was empty. In addition, using `FORCE INDEX` to force a covering index scan on a spatial index led to an assertion. (Bug #112676, Bug #114200, Bug #35894664, Bug #36361834)

- **InnoDB:** Fixed performance issues related to querying the `data_lock` and `data_lock_waits` tables when thousands of read-only transactions were present. (Bug #109539, Bug #34951273)

- **Replication:** If a source contained a stored, generated column populated by a JSON function and `binlog_row_image` was set to `MINIMAL`, any subsequent update or deletion on the underlying column failed with the following error:

  ```
  Invalid JSON text in argument 1 to function json_extract: 'The document is empty.'
  ```

  The replica attempted to re-evaluate the generated column and failed with that error because the underlying column was unavailable. As of this release, stored, generated columns are not re-evaluated when the underlying columns are unavailable. (Bug #36515172)

- **Replication:** When running GTID-based replication with `relay_log_space_limit` enabled, a restart of the auto positioning protocol sometimes resulted in an infinite loop, leading to a deadlock in replication. This was because `relay_log_space_limit` was not honored, not only for transactions whose size exceed this limit, but when the replica could not purge previous logs as well.

  To fix this issue, we make the following changes:

  - The receiver respects `relay_log_space_limit` as set by the user, unless a transaction received by the receiver cannot fit into the purged relay log. Before queuing the received transaction, receiver now checks whether scheduling a full transaction is possible. If not, the receiver performs the following actions:

    - Sets the flag indicating that receiver is waiting
    - Rotates the relay log
    - Waits until it is notified that relay log purge was executed and that the applier has purged all available relay logs; after this, the receiver may queue a transaction without checking the limit again

  - Before moving to the next file, the coordinator checks whether the receiver is waiting for available relay log space. If so, the coordinator forcibly purges the applied logs, including the current relay log file. To purge the current relay log file safely, the coordinator must do the following:

    - Synchronize all of its workers before moving to the next file
    - Forcibly update group positions, which is necessary to allow current purging of the relay log
    - Update the variable read by the receiver which contains relay log filename to which applier was moved

    These operations are allowed because we know that receiver waits at a transaction boundary and rotates the relay log before waiting.

  (Bug #36507020)

- **Replication:** Worker jobs now contain information about the relay log file which initiated the transaction, instead of using the default defined by `relay_log`. (Bug #36395631)

- **Replication:** Handling an incident while transactions were being committed to the binary log caused MySQL to wait indefinitely. (Bug #35671897)
- **Group Replication:** Removed a memory leak from `/xcom/gcs_xcom_networking.cc`. (Bug #36532199)
- **Group Replication:** Under certain circumstances, if a primary's host experienced network inactivity of 20 seconds or more, the secondaries could stop unexpectedly. (Bug #36306144)
- **Group Replication:** Under certain circumstances, if garbage collection occurred just before a relay log rotation, it could cause the applier to stop applying new transactions on the secondary members.

  This was caused by garbage collection incrementing the relay log's `last_committed` and `sequence_number`, creating a gap in the recorded `sequence_number` after the log rotation. The applier was unaffected if the gap occurred anywhere else in the relay log.

  As of this release, only `last_committed` is updated during garbage collection. (Bug #36280130, Bug #36446250)
- **JSON:** Added missing checks for error handling to `NULLIF()`, `COALESCE()`, and the shift (`>>`) operator. (Bug #113668, Bug #35513196, Bug #36198403)

  References: See also: Bug #31358416.
- **MySQL NDB ClusterJ:** Running the ClusterJ test suite resulted in an error message saying a number of threads did not exist. That was due to some wrong handling of threads and connections, which was corrected by this patch. (Bug #36086735)
- Averages of certain numbers were not always computed correctly. (Bug #36563773)
- The following files in `strings` contained incorrect license information:
  - `mb_wc.h`
  - `ctype-uca.cc`
  - `ctype-ucs2.cc`
  - `ctype-utf8.cc`
  - `dtoa.cc`
  - `strxmov.cc`
  - `strxnmov.cc`

  (Bug #36506181)
- In certain unusual cases, the `UpdateXML()` function did not process all of its arguments correctly. (Bug #36479091)
- Explaining a query which used `FORCE INDEX` on a spatial index containing a column with SRID attributes led to an unplanned exit. (Bug #36418426)
- When incrementing the reference count for an expression, underlying expressions within this expression are not looked at. While removing an expression, after decrementing the reference count, even the underlying expressions were examined, which led to unintentional deletion of the underlying expressions. This issue manifested in `Item_ref::real_item()` as well as in an assert in `sql/`

`item.h`. We fix this by not looking at the underlying expression unless the current expression contains the only remaining reference. (Bug #36204344, Bug #36356279)

- Under certain conditions, `EXPLAIN FORMAT=JSON FOR CONNECTION` sometimes led to an unplanned exit. (Bug #36189820)
- Some `CREATE USER` statements were not handled correctly. (Bug #36022885)
- For a `SELECT` with `ORDER BY` and `LIMIT`, the optimizer first chose a full table scan with a very expensive cost, then performed another check and used the `perform_order_index` type of path, but this was not reflected by the cost in the optimizer plan. (Bug #35930969)
- Client connections were not alway terminated correctly during shutdown. (Bug #35854919)
- All internal ACL bitmask variables are now explicitly 32 bits (`uint32_t`). (Bug #35507223)
- It was not possible to add a functional index on `FIND_IN_SET()`. (Bug #35352161)
- Running two concurrent `OPTIMIZE TABLE` statements on the same table with fulltext indexes and `innodb_optimize_fulltext_only` enabled sometimes caused the server to exit. (Bug #34929814)
- Removed a memory leak observed while running `authentication_kerberos` under Valgrind. (Bug #34482788, Bug #36570929)
- The `gen_range()` function as implemented by the (deprecated) data masking plugin did not always return the correct result.

  This issue affected the data masking plugin only, and did not affect the data masking component which supersedes it. (Bug #34163992)
- Fixed an erroneous comment in `include/my_command.h`.

  Our thanks to Sho Nakazono for the contribution. (Bug #114507, Bug #36455468)
- It was possible for a deterministic stored function to return an incorrect result when the function used `JOIN ON` inside the `return` statement. If the query needed to be reprepared due to a table metadata caused by, for example, `FLUSH TABLES` between two executions, the `ON` clause was sometimes lost. (Bug #114235, Bug #36379879)

# Changes in MySQL 8.0.37 (2024-04-30)


- Audit Log Notes
- Character Set Support
- Compilation Notes
- Component Notes
- Optimizer Notes
- Performance Schema Notes
- Functionality Added or Changed
- Bugs Fixed

## Audit Log Notes

- MySQL now calls `plugin->deinit()` with a valid plugin struct as an argument regardless of the plugin's type.

  Our thanks to Martin Alderete for the contribution.

## Character Set Support

- When the `character_set_server` system variable was set using `SET PERSIST` or `SET GLOBAL`, it did not take effect for new client sessions or for a client establishing a connection to the server after the server was restarted. The only workaround was to set the corresponding command-line option when starting the server.

  To fix this, we now make sure that, at the time of server restart, the configuration data is read in the correct order so that the variable setting takes effect as expected. (Bug #35529604)

## Compilation Notes

- Warnings for unused variables are turned into compilation errors when compiling with `-DMYSQL_MAINTAINER_MODE=1`. To avoid this use `-DMYSQL_MAINTAINER_MODE=0` to disable such errors. (Bug #113662, Bug #36198423)

- Improved the cycle timer for the s390x architecture.

  Our thanks to Jonathan Albrecht for the contribution. (Bug #112845, Bug #35949958)

## Component Notes

- The values for component options set using the `--loose` prefix were not read when the component was installed. (Bug #28341329)

## Optimizer Notes

- The multi-range read (MRR) optimization did not perform as well as in previous releases. (Bug #113711, Bug #36220640)

## Performance Schema Notes

- User variables assigned decimal values were rounded up in the `user_variables_by_thread` table. (Bug #35781732)

- Executing `SELECT * from performance_schema.data_locks` on a server under heavy load could cause MySQL to consume too much memory and close unexpectedly.

  As of this release, memory used executing such a query is now instrumented with `memory/performance_schema/data_container`, enabling you to observe memory consumption. (Bug #35240825)

- Executing `SELECT * from performance_schema.data_locks` on a server under heavy load could cause a deadlock in InnoDB. (Bug #35068461)

  References: See also: Bug #35240825.

## Functionality Added or Changed

- **Important Change; Group Replication:** When issued with `group_replication_consistency` set to `BEFORE_ON_PRIMARY_FAILOVER`, the MySQL `KILL` statement now ignores any consistency guarantees, with any interrupted transactions now being rolled back.
- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 3.0.13. Issues fixed in OpenSSL version 3.0.13 are described at https://openssl-library.org/news/openssl-3.0-notes/. (Bug #36261675)
- **Packaging:** Added support for Fedora 40 and Ubuntu 24.04.
- **Microsoft Windows:** MySQL Windows binary files (`.exe` and `.dll` files) now display additional information when their properties are viewed. (Bug #36379291)
- Clone plugin version requirements have been relaxed to allow cloning between different point releases in the same series. In other words, only the major and minor version numbers must match when previously the release number also had to match.

  For example, cloning of MySQL 8.0.37 to (a future) MySQL 8.0.42, or the reverse, is supported. Previous restrictions still apply to versions older than 8.0.37, for example, so cloning of MySQL 8.0.36 to any other MySQL release (or from a different MySQL release to 8.0.36) remains unsupported.

  For more information, see The Clone Plugin. (Bug #36293529, WL #15989)

## Bugs Fixed

- **Important Change:** The Robin Hood hashing library has been replaced with `unordered_dense`. (Bug #36158022)
- **InnoDB; Microsoft Windows:** Improved redo log performance on Windows by opening redo log files in overlapped mode. (Bug #36154818)

  References: This issue is a regression of: Bug #12527.
- **InnoDB:** The log writer calls functions that temporarily release `log.writer_mutex`; when `innodb_log_writer_threads=OFF`, this potentially led to other threads writing to the redo log in between these times. (Bug #36425219)
- **InnoDB:** Some FTS operations on tables with FTS indexes led to inconsistent results. For example, if the server terminated while synchronizing the FTS cache or when synchronization occurred concurrently with another FTS operation.

  Our thanks to Yin Peng and the Tencent team for the contribution. (Bug #36347647)
- **InnoDB:** When creating an index on a table containing data, `valgrind` occasionally reported reads of uninitialized memory from `ddl::Builder::bulk_add_row()`. (Bug #36342792)
- **InnoDB:** Fixed an issue relating to the merge sort buffer. (Bug #36319083)
- **InnoDB:** On Windows, keeping a file open without a shared write lock and attempting to acquire the `fil_shard` mutex caused a deadlock with another thread that had acquired the `fil_shard` mutex and was attempting to access the same file. (Bug #36159317)

  References: See also: Bug #32808809.
- **InnoDB:** Fixed a potential redo log rotation issue that could emit a "Found existing redo log files, but at least one is missing" error during recovery. (Bug #36124625)

- **InnoDB:** Improved `os_innodb_umask` handling, and made it read-only. (Bug #35932118)

  References: This issue is a regression of: Bug #29472125.

- **InnoDB:** Found and fixed an assertion failure related to full-text indexes. (Bug #35836581)

- **InnoDB:** Improved buffer handling during the tablespace deletion process, a situation that could have potentially caused an assertion failure. (Bug #35676106, Bug #36343647)

- **InnoDB:** If a MySQL table in a system schema had an INSTANT ADD column that was added before 8.0.29 (they are not allowed as of that version), and after MySQL was upgraded to a version greater than 8.0.29, DMLs on these tables would result in the server unexpectedly closing.

  Our thanks to Richard Dang for the contribution. (Bug #35625510, Bug #35981565, Bug #36180360)

- **InnoDB:** The redo log would potentially not log a column order change with instant DDL, which could cause an incorrect log replay during recovery. (Bug #35183686)

- **InnoDB:** Results for `SHOW ENGINE INNODB STATUS` showed an invalid value (`NULL`) as the name of the first thread in the `File IO` section. (Bug #113819, Bug #36118112)

  References: This issue is a regression of: Bug #34992157.

- **InnoDB:** With `innodb_parallel_read_threads` set to a value greater than 1, `InnoDB` unnecessarily disabled read-ahead heuristics which resulted in stalls when pages were not already in the buffer pool. (Bug #113482, Bug #36142806)

- **InnoDB:** Running a query that used a unique hash index with the TempTable storage engine could take significantly more time compared to running the query with the MEMORY engine.

  Our thanks to xiaoyang chen for the contribution. (Bug #113178, Bug #36037224, Bug #36224958)

- **InnoDB:** In debug builds, there was an assertion failure in InnoDB's background when a transaction it wanted to acquire an MDL lock on was no longer active.

  This fix is based on a patch from Genze Wu with Alibaba, thank you for the contribution. (Bug #112424, Bug #35835864)

  References: This issue is a regression of: Bug #33700835.

- **InnoDB:** With `innodb_parallel_read_threads` set to a value greater than 1, InnoDB would unnecessarily request asynchronous reads which required more synchronization during I/O completion and created a bottleneck due to the limited number of available threads (`innodb_read_io_threads`) for handling I/O operations. Now this performs synchronous instead of asynchronous reads. (Bug #112137, Bug #35740866)

- **InnoDB:** A trx would unexpectedly halt after encountering an incorrect trx->in_innodb value.

  Our thanks to Shaohua Wang for the contribution. (Bug #110652, Bug #35277407)

- **InnoDB:** `SELECT ... GROUP BY` queries were at least twice as slow with the TempTable engine than the Memory engine. (Bug #107700, Bug #34338001)

- **InnoDB:** MySQL no longer ignores the optimizer hint to use a secondary index scan, which instead forced a clustered (parallel) index scan. (Bug #100597, Bug #112767, Bug #31791868, Bug #35952353)

- **Group Replication:** Problems arose when members `M1` and `M2` were in a group, with `M1` using `u1` as its recovery user and `M2` using `u2` as its own recovery user, and both users `u1` and `u2` existing on `M1` and

M2 with all necessary privileges, and when a new member M3 joined the group using u2 as its recovery user. M3 knew only of user u2, but did not know of user u1, leading START GROUP REPLICATION on M2 to be rejected since M1 was unable to connect to M2. This also generated a new view_id listing the group members as M1 and M2, but M1 nevertheless continued trying to connect to M3, with M1 logging Error in establishing mysql connection and M3 logging Access denied errors for the connection attempts from M1.

By design, XCom stores the last three known configurations, including references to physical connections shared among all past and present configurations. This is done to facilitate quick reconnections by nodes rejoining the group, explicitly or implicitly, and that were already present in any of those configurations.

A side effect of this was that we might keep attempting to connect to a node that was currently not in the group. To solve this problem, we inhibit error logging if the node is not in the current configuration, in order to avoid false negatives which might lead a DBA or an operator to think mistakenly that there is a problem in the system. (Bug #36210988)

References: See also: Bug #32592027.

- **Group Replication:** Improved handling of GTID sets. (Bug #36093405)
- **Group Replication:** Two cases were found in which a member exited the group and moved to the ERROR state, but did not honor the action specified by group_replication_exit_state_action; these are listed here:
  - When an error occurred while enabling super_read_only
  - When member join recovery was not possible, due to missing binary logs and clone groups on group members

  *Example*: When the value of group_replication_exit_state_action was OFFLINE_MODE and one of these events took place, offline mode was not enabled as expected. (Bug #36076308)
- **Group Replication:** After successfully setting a new primary, group_replication_set_as_primary() in some cases waited indefinitely for the operation to complete. (Bug #36059098)
- **JSON:** JOIN and GROUP BY handled some JSON column values differently. (Bug #101048, Bug #31969607)
- **MySQL NDB ClusterJ:** The setLimits() method can now be chained to deletePersistentAll() to limit the number of items to delete. See the description of deletePersistentAll() for details. (Bug #36049906)
- Updated BuildRequire rules to align with versions now required for CMake and Bison. (Bug #36343254)
- SET GLOBAL offline_mode=ON did not always perform correctly when issued under high loads. (Bug #36275182)

  References: See also: Bug #36405894.
- Upgraded curl to version 8.6.0. (Bug #36267545)
- mysqldump did not always interpret the server version correctly. (Bug #36248967)

  References: See also: Bug #36405879.

- Condition pushdown to a view was rejected with a collation mismatch if the view was created with a different character set than the character set used when querying the view. (Bug #36246859)
- Use of the deprecated `exec_program()` command has been replaced by `execute_process()` to provide compatibility with CMake 3.28.1 and later. (Bug #36220656)
- Upgraded the `protobuf` library to version 25.1. (Bug #36108397)
- For building Enterprise Linux RPMs, the build scripts now point to a newer `strip` command (under `/opt/rh/gcc-toolset-12`), and they now check that the corresponding `dwz` tool is available. Previously this was only implemented for EL8. (Bug #36090069)
- We now look for `gcc-ar` and `gcc-ranlib` when building on Oracle Linux with link-time optimization. (Bug #36089900)
- The server sometimes terminated unexpectedly in response to a specific query. (Bug #35957627)
- In certain cases, a lateral join was not handled correctly. (Bug #35945239)

  References: See also: Bug #107700, Bug #34338001. This issue is a regression of: Bug #32644631.
- Improved view and trigger definer handling by view and table DDL. (Bug #35942937)
- The server did not always return metadata to the client correctly for certain queries. (Bug #35904044, Bug #36521376)
- When running queries against a table with a multi-value index, the server sometimes exited unexpectedly, often while executing a complex `SELECT` query which used this index. (Bug #35789759)
- Improved code in `sql/item_subselect.cc`. (Bug #35733778, Bug #35738531, Bug #35779012)
- `CREATE USER IF EXISTS` was not always logged correctly. (Bug #35530823)
- The server did not disallow subqueries in partition expressions properly. These are invalid, and should cause a syntax error. (Bug #35476172)
- Some `RANK() ... OVER()` queries raised an assertion in `sql/sql_executor.cc`. (Bug #35228083)
- When successive `ALTER TABLE ... ALGORITHM=COPY` statements were issued within 10 seconds of one another, the `n_rows` value became 0. (Bug #35127747)
- Updated the URL used for downloading the Boost C++ libraries. (Bug #113576, Bug #36164514)
- The fix for a previous issue, first addressed in MySQL 8.0.30, was incomplete.

  Our thanks to Hao Lu for the contribution. (Bug #113174, Bug #36035044)

  References: This issue is a regression of: Bug #110801, Bug #35328028.
- On s390x, we now compile the FMA test with `-O2` to avoid overoptimization.

  Our thanks to Jonathan Albrecht for the contribution. (Bug #113096, Bug #36016140)
- Although s390x is a big-endian platform, the little-endian ICU data directory was used for compiling.

  Our thanks to Jonathan Albrecht for the contribution. (Bug #113095, Bug #36016141)
- Incorrect results were sometimes obtained from a query that used a group by loose index scan. (Bug #112541, Bug #35854362)

- An assertion failed in debug builds when inserting data with a zero-length column, such as `CHAR(0)` or `BINARY(0)`, into a table. Now, a less strict assertion more accurately fails only if it detects that a non-zero number of bytes copied from a source is identical to the target. (Bug #111450, Bug #35507763)
- A `VALUES` statement in a correlated lateral or (other) dependent subquery yielded an incorrect result. (Bug #109252, Bug #110076, Bug #34852090, Bug #35087820)
- `IN` should return `NULL` not only if the expression on the left hand side is `NULL`, but when no match is found in the list and one of the expressions in the list is `NULL` as well. In some cases, it was found that adding rows to a table caused a subsequent prepared `SELECT NULL IN (SELECT ...)` query to return a different result when the result should have remained the same. (Bug #94857, Bug #29602835)

# Changes in MySQL 8.0.36 (2024-01-16)


- Audit Log Notes
- Compilation Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Pluggable Authentication
- Functionality Added or Changed
- Bugs Fixed


## Audit Log Notes

- In some cases, calling `audit_log_read( audit_log_read_bookmark() )` led to an `Out of memory` error. (Bug #35957453)

## Compilation Notes

- **Microsoft Windows:** MySQL did not compile correctly using Visual Studio 2022. (Bug #35967676)
- Improved the `-DWITH_ZLIB=system` check. (Bug #35968195)
- For compiling on Linux, changed the `no-error=deprecated-declarations` flag to `no-deprecated-declarations` for the OpenSSL 3 library.

  Our thanks to karry zhang for the contribution. (Bug #112209, Bug #35755328)

## Optimizer Notes

- The hashing algorithm employed yielded poor performance when using a `HASH` field to check for uniqueness. (Bug #109548, Bug #34959356)

## Packaging Notes

- **Important Change:** The GnuPG build key (`A8D3785C`) used to sign MySQL downloadable packages has been updated. The previous GnuPG build key (`3A79BD29`) expired on 2023-12-14. For information

about verifying the integrity and authenticity of MySQL downloadable packages using GnuPG signature checking, or to obtain a copy of our public GnuPG build key, see Signature Checking Using GnuPG.

Due to the GnuPG key update, systems configured to use `repo.mysql.com` may report a signature verification error when upgrading to MySQL 8.0.36 and higher or to MySQL 8.3.0 and higher using `apt` or `yum`. Use one of the following methods to resolve this issue:

1. Manually reinstall the MySQL APT or YUM repository setup package from https://dev.mysql.com/downloads/.
2. Download the MySQL GnuPG public key and add it your system GPG keyring.

## Performance Schema Notes

- When executing a stored program, the Performance Schema instrumentation caused some unnecessary overhead.

  As of this release, all stored procedure micro instructions (`statement/sp/%`), except `statement/sp/stmt`, are disabled by default. (Bug #27934653)

- The performance of the Performance Schema statement instrumentation has been improved. Specifically, collecting `MESSAGE_TEXT` data is now more efficient. (Bug #112621, Bug #35916912)

## Pluggable Authentication

- Beginning with this release, the behavior of the `AUTHENTICATION_PAM_LOG` environment variable used in debugging the PAM authentication plugin is changed as follows:

  - Setting `AUTHENTICATION_PAM_LOG` to an arbitrary value (except as noted in the next item) no longer includes passwords in its diagnostic messages.
  - To include passwords in the diagnostic messages, set `AUTHENTICATION_PAM_LOG=PAM_LOG_WITH_SECRET_INFO`.

  For more information, see PAM Authentication Debugging. (Bug #74313, Bug #20042010)

## Functionality Added or Changed

- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 3.0.12. Issues fixed in OpenSSL version 3.0.12 are described at https://openssl-library.org/news/openssl-3.0-notes/. (Bug #36033684)

## Bugs Fixed

- **Performance; InnoDB:** The hashing function used by the `InnoDB` adaptive hash index became more expensive as an unintended result of refactoring done in MySQL 8.0.30. (Bug #111538, Bug #35531293)

  References: This issue is a regression of: Bug #81814, Bug #16739204, Bug #23584861.

- **InnoDB:** The hash function used by the adaptive hash index (AHI) was improved to increase performance. (Bug #35449386)

- **InnoDB:** If change buffer entries are present during startup, a disabled `innodb_validate_tablespace_paths` option will no longer be enforced and instead the MySQL server will proceed to validate all tablespaces. Otherwise, secondary indexes could end up corrupted. (Bug #35208990)

- **InnoDB:** During concurrent DDL and DML operations, DDL could fail if the online log grew too large. Buffer handling was improved to prevent this issue. (Bug #35115601)
- **Replication:** An issue with calculating the current number of bytes used for `Log_event` events in Performance Schema memory instrumentation made it appear as though the `sql/replica_sql` thread on the replica grew endlessly and never decreased in size. (Bug #35546877)
- **Replication:** Stopping replication while replicating `CREATE TABLE AS SELECT` caused the server to exit. (Bug #33934013)
- **Group Replication:** A forced `START GROUP_REPLICATION` while a replication channel was in an error state could lead to an unplanned server exit. (Bug #34724344)
- For building Enterprise Linux 8 RPMs, the build scripts now point to a newer `strip` command (under `/opt/rh/gcc-toolset-12`), and they now check that the corresponding `dwz` tool is available. (Bug #36086236)
- In some cases, calling a loadable function installed by an improperly initialized plugin caused an unplanned shutdown. (Bug #35889261)
- Found and fixed an assertion failure at `handler::ha_index_end()` in `handler.cc`. (Bug #35877600)
- When the `MYSQL_FIREWALL` plugin failed to initialize properly during server startup, this led to errors in the server log and sometimes unplanned termination of `mysqld`. (Bug #35853298)
- Some nested queries with `GROUP BY` were not handled correctly. (Bug #35846402, Bug #35945822)

  References: This issue is a regression of: Bug #32918400.
- In limited cases, passing data to the `MD5()` encryption function could halt the server. (Bug #35764496)
- Some subselects from views were not always handled correctly. (Bug #35738548)
- While performing an operation such as the bulk renaming of many tables, simultaneously executing a data definition statement similar to `CREATE TABLE ... SELECT` could stop the server unexpectedly. (Bug #35735937)
- `UPDATE HISTOGRAM` did not behave as expected in all cases.

  UPDATE HISTOGRAM did not behave as expected in all cases. (Bug #35710404)
- `EXPLAIN ANALYZE` did not always produce the expected result. (Bug #35710383)
- An error occurred during subquery resolution. (Bug #35710373)

  References: This issue is a regression of: Bug #35184353.
- Refreshing of used table information is now postponed to the start of the next execution, just after tables have been opened, when we know that all table objects are in a proper state. (Bug #35710213, Bug #36030073)
- Some `HAVING` queries did not produce expected results. (Bug #35710183)
- Some recursive CTEs did not function as expected. (Bug #35654240)
- Some queries using `OVER (PARTITION ...)` were not always executed successfully. (Bug #35627798)
- Some subqueries with `ROLLUP` were not always handled correctly. (Bug #35621842, Bug #35804794)

- Removed the `CPACK_COMPONENT_GROUP_INFO_DISPLAY_NAME` configuration option from the Windows installation MSI interface. Now the `INFO_BIN` and `INFO_SRC` files are always installed. (Bug #35529968)
- Some queries using windowing functions were not always handled correctly. (Bug #35471471)
- In debug builds, a case-altered column name could cause the server to exit. (Bug #35449266)
- MySQL Server installation packages contained two copies of the `INFO_SRC` file. (Bug #35400142)
- A `SELECT` statement within a prepared statement unexpectedly returned different results on successive executions. (Bug #35340987, Bug #35846585, Bug #35846873)

  References: This issue is a regression of: Bug #35060385.
- Some `SELECT DISTINCT` queries were not always handled correctly. (Bug #33725447)
- Removed an assertion failure in `sql/field.cc`. (Bug #112503, Bug #35846221)
- Sme queries having the form `SELECT AVG(...) OVER (PARTITION BY ...)` were not always handled correctly. (Bug #112460, Bug #35710179, Bug #35845413)
- Upgrading MySQL using an official MySQL Yum or SUSE repository always enables the MySQL service. Now it enables the service only after installing, and preserves (and does not edit) the existing value while upgrading. (Bug #112382, Bug #35823558)
- For a query with a derived condition pushdown where a column in the condition needs to be replaced, a matching item could not found, even when known to be present, when the replacement item was wrapped in a `ROLLUP` while the matching item was not. (Bug #111665, Bug #35498378, Bug #35570065, Bug #35826171)

  References: This issue is a regression of: Bug #33349994.
- Ordering a nested block where the `ORDER BY` contained a window function raised an assert in `sql/sql_executor.cc`. (Bug #111306, Bug #34856256, Bug #35471522)
- Performing an arithmetic operation on the result over a window function in a stored procedure gave the correct result the first time the procedure was executed, but returned an incorrect result on all subsequent invocations. (Bug #110983, Bug #35380604)

  References: See also: Bug #110847, Bug #35340987.
- MySQL did not build correctly using the `musl` version of `libc`.

  Our thanks to Sam James for the contribution. (Bug #110808, Bug #35330950)
- In some cases, selecting from a view leaked a small amount of memory. (Bug #103133, Bug #32764586)

# Changes in MySQL 8.0.35 (2023-10-25)


- Audit Log Notes
- Authentication Notes
- C API Notes
- Compilation Notes
- Deprecation and Removal Notes

- SQL Function and Operator Notes
- Packaging Notes
- Security Notes
- Functionality Added or Changed
- Bugs Fixed

## Audit Log Notes

- Added the new `audit_log_filter_uninstall.sql` script to simplify removing MySQL Enterprise Audit. For information about this plugin, see MySQL Enterprise Audit. (Bug #35611072)

## Authentication Notes

- The pluggable FIDO authentication method is deprecated and subject to removal in a future MySQL release. (WL #15006)

## C API Notes

- Earlier distributions, such as MySQL 5.1 and MySQL 5.0, included servers that did not advertise pluggable authentication. When a newer client using any pluggable authentication method attempted to connect to one of these older servers, the server emitted a “bad handshake” error message. This fix now ensures that client-side authentication (specifically, the `--default-auth` option) works properly for all servers, including those that do not support pluggable authentication. In addition, `mysql_native_password` now replaces `caching_sha2_password` as the default authentication method when the server does not support pluggable authentication. (Bug #90994, Bug #28082093)
- The `mysql_ssl_set()` C API function is deprecated and subject to removal in a future MySQL release. There are equivalent `mysql_options()` TLS options for all `mysql_ssl_set()` parameters. (WL #11091)

## Compilation Notes

- Fixed the `WITH_DEVELOPER_ENTITLEMENTS` macOS CMake option; it did not activate due to a typo. (Bug #35374026)
- Fixed a string concatenation warning produced when compiling with Clang 12. (Bug #111614, Bug #35549962)
- Made additional improvements to `WITH_ZLIB` functionality.

  Thanks to Nikolai Kostrigin for the contribution. (Bug #111549, Bug #35534309)
- Building with WITH_PROTOBUF=system failed with Protobuf 22 or newer due to Protobuf 22 adding the Abseil dependency. (Bug #111469, Bug #111623, Bug #35546459, Bug #35550389)
- Building with `WITH_ZLIB`="system" would break the MySQL build as it failed to find ZLIB.

  Our thanks to Meng-Hsiu Chiang and the team at Amazon for contributing to this fix. (Bug #111467, Bug #35511210)

  References: This issue is a regression of: Bug #35057542.
- On EL7 aarch64-based platforms, fixed an issue related to how fetching the CPU cache line size returned 0 that caused the MySQL server to unexpectedly halt. (Bug #110752, Bug #35479763)

References: See also: Bug #107081, Bug #34095278.

- The C++ Standardization Committee's Library Working Group recently resolved an issue (LWG-3865 *Sorting a range of pairs*) which changes how the comparison operators are defined for `std::pair`. This fix updates the equality operator used in two files in `sql/auth` to align with this change.

  Based on a suggestion by the Microsoft Visual Studio team. (Bug #110254, Bug #35137978)

## Deprecation and Removal Notes

- **Replication:** The `binlog_transaction_dependency_tracking` server system variable is now deprecated, and subject to removal in a future version of MySQL. Setting or retrieving the value of this variable now triggers a warning, as does using the equivalent startup option `--binlog-transaction-dependency-tracking`. Applications depending on this variable (or option) should begin migrating away from it as soon as possible. No replacement for this variable or its functionality is planned or needed. (WL #13964)

- The `old` and `new` server system variables are now deprecated, and a warning is now issued whenever either of these variables is set or read. Expect these variables are to be removed in a future version of MySQL; applications which rely on them should migrate away from such dependencies as soon as possible. (WL #8609)

- The `--character-set-client-handshake` server option, originally intended for use with upgrades from very old versions of MySQL, is now deprecated, and a warning is issued whenever it is used. You should expect this option to be removed in a future version of MySQL; applications depending on this option should begin migration away from it as soon as possible. (WL #13220)

- Two changes in this release affect granting of privileges in MySQL. One of these changes impacts database grants; the other concerns host names used in grants. These changes are listed here:

  1. **Database-level grants.** The use of the characters `%` and `_` as wildcards in database-level grants is now deprecated, and you should expect for this wildcard functionality to be removed in a future MySQL release. The intent is for these characters always to be treated as literals, as they are already whenever the value of the `partial_revokes` server system variable is `ON`.

     For example, with `partial_revokes` set to `ON`, `GRANT SELECT ON db_.* TO user1` allows `user1` to select from any table in a database named `db_`, but does *not* allow `user1` to select from tables in databases `db1`, `db2`, `dbx`, `dby`, and so on. You should expect this always to be the case once support for wildcards in database names in grants is removed.

  2. **Host name '%' in grants.** The treatment of `%` by the server as a synonym for `localhost` when checking privileges is now also deprecated and thus also subject to removal in a future version of MySQL.

     Currently, privileges granted to `'user1'@'%'` are also granted to `'user1'@'localhost'`. This automatic assignment is now deprecated; once the behavior is removed, it will be necessary always to grant privileges to `'user1'@'localhost'` explicitly, using the `localhost` host name.

     **Note**

     In both of the cases listed, no warnings or errors are raised in the user session or logs, since doing so could very quickly produce a great many such messages.

  (WL #14280, WL #15676)

- `INFORMATION_SCHEMA.PROCESSLIST` is deprecated and subject to removal in a future MySQL release. As such, the implementation of `SHOW PROCESSLIST` which uses that table is also deprecated.

  It is recommended to use the Performance Schema implementation of `SHOW PROCESSLIST` instead. As such, `performance_schema_show_processlist` is also deprecated and subject to removal in a future MySQL release. (WL #15915)

- Support for the `TLS_AES_128_CCM_8_SHA256` ciphersuite now is deprecated and subject to removal in a future version of MySQL. Any attempt to set it using either the `tls_ciphersuites` or `admin_tls_ciphersuites` system variable now returns a warning. By default, both system variables accept these ciphersuites:

  - `TLS_AES_128_GCM_SHA256`
  - `TLS_AES_256_GCM_SHA384`
  - `TLS_CHACHA20_POLY1305_SHA256`
  - `TLS_AES_128_CCM_SHA256`

  Several previously supported ciphers now are deprecated and subject to removal in a future version of MySQL. If the deprecated ciphers are specified using either the `ssl_cipher` or `admin_ssl_cipher` system variable, then a warning now is issued. The following ciphers continue to be supported without emitting a deprecation warning:

  - `ECDHE-ECDSA-AES128-GCM-SHA256`
  - `ECDHE-ECDSA-AES256-GCM-SHA384`
  - `ECDHE-RSA-AES128-GCM-SHA256`
  - `ECDHE-RSA-AES256-GCM-SHA384`
  - `ECDHE-ECDSA-CHACHA20-POLY1305`
  - `ECDHE-RSA-CHACHA20-POLY1305`
  - `ECDHE-ECDSA-AES256-CCM`
  - `ECDHE-ECDSA-AES128-CCM`
  - `DHE-RSA-AES128-GCM-SHA256`
  - `DHE-RSA-AES256-GCM-SHA384`
  - `DHE-RSA-AES256-CCM`
  - `DHE-RSA-AES128-CCM`
  - `DHE-RSA-CHACHA20-POLY1305`

  (WL #15800)

## SQL Function and Operator Notes

- The `STR_TO_DATE()` function did not perform complete range checking on the string to be converted, so that it was possible to pass to it a string which would yield an invalid date, such as `'2021-11-31'`. (Bug #108782, Bug #34704094)

## Packaging Notes

- On Windows, the MSI package definition files were updated to work with the Windows Installer XML (WiX) toolset version 4. Note that they can no longer be used with previous versions of the toolset. (Bug #35613791)
- The bundled `libedit` library was upgraded to version 20221030-3.1. (Bug #35489173)

## Security Notes

- Binary packages that include curl rather than linking to the system curl library have been upgraded to use curl 8.4.0. Important issues fixed in curl version 8.4.0 are described at https://curl.se/docs/security.html. (Bug #35897778)

  References: See also: Bug #35709229.

## Functionality Added or Changed

- **Incompatible Change:** Upgraded the bundled libfido2 library to libfido2 v1.13.0. This libfido2 version requires OpenSSL 1.1.1 or higher, which means the authentication_fido and authentication_fido_client authentication plugins are no longer available on the following platforms: Enterprise Linux 6 and 7, Solaris 11, and SLES 12. (Bug #35685515)
- **Important Change; Group Replication:** Version requirements for groups are relaxed beginning with this release, as follows:
  - A server running any MySQL 8.0 release of version 8.0.35 *or newer* can join the same group.
  - A group member running any MySQL 8.0 release *newer than* 8.0.35 can be downgraded in place to the previous release.

  This change does not affect the requirement that, when selected, the primary must be running the lowest MySQL version present in the group; this remains true whether this is automatic, or is performed by executing `group_replication_set_as_primary()`. (Bug #35397276)
- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 3.0.10. Issues fixed in OpenSSL version 3.0.10 are described at https://openssl-library.org/news/openssl-3.0-notes/. (Bug #35702863, Bug #35732474)

## Bugs Fixed

- **Performance:** `records_in_range` performed an excessive number of disk reads for insert operations.

  Our thanks to Facebook for the contribution. (Bug #109595, Bug #34976138)
- **InnoDB:** Fixed processing of single character tokens by a FTS parser plugin.

  Our thanks to Shaohua Wang for the contribution. (Bug #35432973)
- **InnoDB:** On Windows, the redo log file was not flushed even with `innodb_flush_log_at_trx_commit=1`. (Bug #112078, Bug #35713721)

- **InnoDB:** Fixed a potential transaction rollback issue stemming from the ALTER TABLE EXCHANGE PARTITIONS statement. (Bug #110869, Bug #35352168)

  References: This issue is a regression of: Bug #33700835.

- **InnoDB:** When creating full-text indexes spanning multiple columns, index creation could fail with an error similar to "ERROR 1062 (23000): Duplicate entry 'NULL-NULL'." (Bug #109242, Bug #33542939, Bug #34846823)

- **InnoDB:** The last detected deadlock section of the engine status log was only showing 1024 characters for the combined thread and query information. Fixing by removing the printed query string limit. (Bug #80927, Bug #23036096)

- **Replication:** Memory used by binary log compression was not instrumented. (Bug #35290223)

- **Replication:** To detect dependencies and conflicts among transactions, MySQL Replication uses writesets which are extracted from keys that are `NOT NULL` (primary keys by definition, as well as unique and foreign keys which are `NOT NULL`) whenever `binlog_format` is `ROW` and `transaction_write_set_extraction` is enabled (the default in both cases). In the case of multi-column unique keys using only the leading parts of column values (see Column Prefix Key Parts), the writesets were not properly generated, which sometimes led to spurious duplicate key errors; this was due to the fact that the whole values of referenced key columns were added to the writeset in such cases, even though the keys themselves referenced only the column prefix. (Bug #111027, Bug #35404584)

- Following a change made in MySQL 8.0.23, in some cases where `GROUP_CONCAT()` had previously returned `NULL`, it did so no longer. (Bug #35730982)

  References: This issue is a regression of: Bug #32053547.

- When no rows matched the `WHERE` condition of a query, the results of the query as prepared (using `PREPARE`) differed from those of the same query when run directly. (Bug #35689804)

- Some queries using `INTERSECT` were not always processed correctly. (Bug #35504625)

  References: See also: Bug #35362424.

- In some cases, the `QUOTE()` function returned binary data rather than the expected `NULL`. (Bug #35499232)

- Previously, executing `audit_log_rotate()` manually could cause the function to appear unresponsive when the output performance was low, the workload on the server was heavy, and the `audit_log_strategy` system variable had a write strategy of `ASYNCHRONOUS` or `PERFORMANCE`. (Bug #35397216)

- In some cases the `QUOTE()` function returned `NULL` even though it had previously been resolved as non-nullable. (Bug #35381715)

  References: See also: Bug #35145246, Bug #35195181.

- A previous fix in MySQL 8.0.30 for a stored program that was not executed correctly following the first invocation did not cover the case where it contained a `SET` statement.

  Our thanks to Hao Lu for the contribution. (Bug #35328028)

  References: This issue is a regression of: Bug #33754993.

- An issue with stored functions was found to be a regression from work done previously to improve internal functions that are used in copying values between columns. (Bug #35150382)

References: This issue is a regression of: Bug #32742537.

- Some cached queries were not always handled correctly.

  Thanks to Nicholas Othieno and Amazon for the contribution. (Bug #35105404, Bug #35511208)

- Updated the Kerberos library bundled with commercial builds to version 1.21.2. (Bug #35001935)

- Some inserts on tables with triggers were not always handled correctly. (Bug #34920120)

  References: See also: Bug #35178672, Bug #35195079.

- Some `DESCRIBE` statements using `FORMAT = JSON` were not always processed correctly, and sometimes raised assertions in debug builds. (Bug #34909766)

- Some complex queries using multiple common table expressions were not always handled correctly. (Bug #34900334)

  References: This issue is a regression of: Bug #34377854.

- Some queries with window functions were not pushed down correctly. (Bug #34778435)

- Some complex queries using multiple common table expressions were not always handled correctly. (Bug #112021, Bug #35284734, Bug #35694546)

  References: See also: Bug #111994, Bug #35686058. This issue is a regression of: Bug #34377854.

- For Enterprise Linux, changed the krb5-devel build requirement from commercial to all builds. (Bug #111596, Bug #35546449)

- `CREATE ROLE` and `DROP ROLE` statements were not handled correctly when checking for orphaned SQL objects. (Bug #111303, Bug #35471453)

- `Clone_Snapshot::extend_and_flush_files()` always created files of type `OS_CLONE_DATA_FILE`. This function uses `flush_redo()` to handle redo log files, which must be of type `OS_CLONE_LOG_FILE`, which could sometimes lead to errors in `os_file_set_size()`.

  This problem is now fixed by ensuring that the type is set correctly when creating the new file.

  Our thanks to Tencent for the contribution. (Bug #110569, Bug #35240055)

- The low limit heuristic did not work well for `ORDER BY DESC` due to choosing the wrong index. (Bug #107626, Bug #34306497)

# Changes in MySQL 8.0.34 (2023-07-18)


- Account Management Notes
- Audit Log Notes
- Binary Logging
- C API Notes
- Compilation Notes
- Deprecation and Removal Notes

- Performance Schema Notes
- SQL Syntax Notes
- Functionality Added or Changed
- Bugs Fixed

## Account Management Notes

- A new password-validation system variable now permits the configuration and enforcement of a minimum number of characters that users must change when attempting to replace their own MySQL account passwords. This new verification setting is a percentage of the total characters in the current password. For example, if `validate_password.changed_characters_percentage` has a value of 50, at least half of the characters in the replacement account password must not be present in the current password, or the password is rejected.

  This new capability is one several that provide DBAs more complete control over password management. For more information, see Password Management. (WL #15751)

## Audit Log Notes

- In MySQL 8.0.33, the `audit_log` plugin added support for choosing which database to use to store JSON filter tables. It is now possible to specify an alternative to the default system database, `mysql`, when running the plugin installation script. Use the `audit_log_database` server system variable (or `-D database_name`) on the command line together with the alternative database name, for example:

  ```
  $> mysql -u root -D database_name -p < audit_log_filter_linux_install.sql
  ```

  For additional information about using `audit_log` plugin installation scripts, see Installing or Uninstalling MySQL Enterprise Audit. (Bug #35252268)

- MySQL Enterprise Audit now supports using the scheduler component to configure and execute a recurring task to flush the in-memory cache. For setup instructions, see Enabling the Audit Log Flush Task. (WL #15567)

## Binary Logging

- Several functions now are added to the `libmysqlclient.so` shared library that enable developers to access a MySQL server binary log: `mysql_binlog_open()`, `mysql_binlog_fetch()`, and `mysql_binlog_close()`.

  Our thanks to Yura Sorokin for the contribution. (Bug #110658, Bug #35282154)

## C API Notes

- In the calling function, `len` is initialized to 0 and never changed if `net->vio` is null. This fix adds a check of `net` before dereferencing `vio`.

  Our thanks to Meta for the contribution. (Bug #30809590)

- A variable in the `async` client was uninitialized in certain code paths. It is fixed by always initializing the variable.

  Our thanks to Meta for the contribution. (Bug #30809568)

## Compilation Notes

- **Microsoft Windows:** For Windows, improved `MSVC_CPPCHECK` support; and check for MSVC warnings similar to "maintainer" mode. For example, check after all third party configurations are complete. (Bug #35283401)

  References: See also: Bug #34828882.

- **Microsoft Windows:** For Windows builds, improved WIN_DEBUG_NO_INLINE=1 support; usage would exceed the library limit of 65535 objects. (Bug #35259704)

- Upgraded the bundled robin-hood-hashing from v3.8.1 to v3.11.5. (Bug #35448980)

- Removed the unused `extra/libcbor/doc/` directory as `extra/libcbor/doc/source/requirements.txt` inspired bogus pull requests on GitHub. (Bug #35433370)

- Updated the bundled ICU files from version 69.1 to version 73 for the `icu-data-files` package. (Bug #35353708)

- ZSTD sources bundled in the source tree were upgraded to ZSTD 1.5.5 from 1.5.0. (Bug #35353698)

- Initialize the internal `MEM_ROOT` class memory with garbage using the TRASH macro to make easier to reproduce bugs caused by reading initialized memory allocated from MEM_ROOT. (Bug #35277644)

- We now determine stack direction at runtime rather than at configure time. (Bug #35181008)

- Added the `OPTIMIZE_SANITIZER_BUILDS` CMake option that adds `-O1 -fno-inline` to sanitizer builds. It defaults to ON. (Bug #35158758)

- Changed the minimum Bison version requirement from v2.1 to v3.0.4. For macOS, this may require installing Bison via a package manager such as Homebrew. (Bug #35154645, Bug #35191333)

- MySQL now sets LANG=C in the environment when executing `readelf` to avoid problems with non-ASCII output.

  Our thanks to Kento Takeuchi for the contribution. (Bug #111190, Bug #35442825)

- On macOS, MySQL would not compile if rapidjson was installed via Homebrew. The workaround was to `brew unlink rapidjson`. (Bug #110736, Bug #35311140)

  References: This issue is a regression of: Bug #35006191.

- MySQL would not build with `-DWITH_ZLIB=system`; it'd complain about not finding the system zlib library despite finding it. (Bug #110727, Bug #110745, Bug #35307674, Bug #35312227)

## Deprecation and Removal Notes

- **Important Change:** Since MySQL provides other means of performing database dumps and backups with the same or additional functionality, including `mysqldump` and MySQL Shell Utilities, the `mysqlpump` client utility program has become redundant, and is now deprecated. Invocation of this program now produces a warning. You should keep in mind that `mysqlpump` is subject to removal in a future version of MySQL, and move applications depending on it to another solution, such as those mentioned previously. (WL #15652)

- **Replication:** The `sync_relay_log_info` server system variable is deprecated in this release, and getting or setting this variable or its equivalent startup option `--sync-relay-log-info` now raises a warning.

Expect this variable to be removed in a future version of MySQL; applications which make use of it should be rewritten not to depend on it before this happens. (Bug #35367005, WL #13968)

- **Replication:** The `binlog_format` server system variable is now deprecated, and subject to removal in a future version of MySQL. The functionality associated with this variable, that of changing the binary logging format, is also deprecated.

  The implication of this change is that, when `binlog_format` is removed, only row-based binary logging, already the default in MySQL 8.0, will be supported by the MySQL server. For this reason, new installations should use only row-based binary logging, and existing ones using the statement-based or mixed logging format should be migrated to the row-based format. See Replication Formats, for more information.

  The system variables `log_bin_trust_function_creators` and `log_statements_unsafe_for_binlog`, being useful only in the context of statement-based logging, are now also deprecated, and are thus also subject to removal in a future release of MySQL.

  Setting or selecting the values of any of the variables just mentioned now raises a warning. (WL #13966, WL #15669)

- **Group Replication:** The `group_replication_recovery_complete_at` server system variable is now deprecated, and setting it produces a warning. You should expect its removal in a future release of MySQL. (WL #15460)

- The `mysql_native_password` authentication plugin now is deprecated and subject to removal in a future version of MySQL. `CREATE USER`, `ALTER USER`, and `SET PASSWORD` operations now insert a deprecation warning into the server error log if an account attempts to authenticate using `mysql_native_password` as an authentication method. (Bug #35336317)

- Previously, if the `audit_log` plugin was installed without the accompanying audit tables and functions needed for rule-based filtering, the plugin operated in legacy filtering mode. Now, legacy filtering mode is deprecated. New deprecation warnings are emitted for legacy audit log filtering system variables. These deprecated variables are either read-only or dynamic.

  (Read-only) `audit_log_policy` now writes a warning message to the MySQL server error log during server startup when the value is not `ALL` (default value).

  (Dynamic) `audit_log_include_accounts`, `audit_log_exclude_accounts`, `audit_log_statement_policy`, and `audit_log_connection_policy`. Dynamic variables print a warning message based on usage:

  - Passing in a non-NULL value to `audit_log_include_accounts` or `audit_log_exclude_accounts` during MySQL server startup now writes a warning message to the server error log.
  - Passing in a non-default value to `audit_log_statement_policy` or `audit_log_connection_policy` during MySQL server startup now writes a warning message to the server error log. `ALL` is the default value for both variables.
  - Changing an existing value using `SET` syntax during a MySQL client session now writes a warning message to the client log.
  - Persisting a variable using `SET PERSIST` syntax during a MySQL client session now writes a warning message to the client log.

  (WL #11248)

- MySQL enables control of FIPS mode on the server side and the client side using a system variable and client option. Application programs can use the `MYSQL_OPT_SSL_FIPS_MODE` option to `mysql_options()` to enable FIPS mode on the client. Alternatively, it is possible to handle FIPS mode directly through OpenSSL configuration files rather than using the current server-side system variable and client-side options. When MySQL is compiled using OpenSSL 3.0, and an OpenSSL library and FIPS Object Module are available at runtime, the server reads the OpenSSL configuration file and respects the preference to use a FIPS provider, if one is set. OpenSSL 3.0 is certified for use with FIPS.

  To favor the OpenSSL alternative, the `ssl_fips_mode` server system variable, `--ssl-fips-mode` client option, and the `MYSQL_OPT_SSL_FIPS_MODE` option now are deprecated and subject to removal in a future version of MySQL. A deprecation warning prints to standard error output when an application uses the `MYSQL_OPT_SSL_FIPS_MODE` option or when a client user specifies the `--ssl-fips-mode` option on the command line, through option files, or both.

  Prior to being deprecated, the `ssl_fips_mode` server-side system variable was dynamically settable. It is now a read-only variable (accepts `SET PERSIST_ONLY`, but not `SET PERSIST` or `SET GLOBAL`). When specified on the command line or in the `mysqld-auto.cnf` option file (with `SET PERSIST_ONLY`) a deprecation warning prints to the server error log. (WL #15631)

- The `mysql_ssl_rsa_setup` program originally provided a simple way for community users to generate certificates manually, if OpenSSL was installed on the system. Now, `mysql_ssl_rsa_setup` is deprecated because MySQL Community Edition no longer supports using yaSSL as the SSL library, and source distributions no longer include yaSSL. Instead, use MySQL server to generate missing SSL and RSA files automatically at startup (see Automatic SSL and RSA File Generation). (WL #15668)

- The `keyring_file` and `keyring_encrypted_file` plugins now are deprecated. These keyring plugins are superseded by the `component_keyring_file` and `component_keyring_encrypted_file` components. For a concise comparison of keyring components and plugins, see Keyring Components Versus Keyring Plugins. (WL #15659)

- Previously, the MySQL server processed a version-specific comment without regard as to whether any whitespace followed the MySQL version number contained within it. For example, the comments `/*!80034KEY_BLOCK_SIZE=1024*/` and `/*!80034 KEY_BLOCK_SIZE=1024*/` were handled identically. Beginning with this release, when the next character following the version number in such a comment is neither a whitespace character nor the end of the comment, the server issues a warning: `Immediately starting the version comment after the version number is deprecated and may change behavior in a future release. Please insert a whitespace character after the version number`.

  You should expect the whitespace requirement for version-specific comments to become strictly enforced in a future version of MySQL.

  See Comments, for more information. (WL #15686)

- The MySQL client library currently supports performing an automatic reconnection to the server if it finds that the connection is down and an application attempts to send a statement to the server to be executed. Now, this feature is deprecated and subject to removal in a future release of MySQL.

  The related `MYSQL_OPT_RECONNECT` option is still available but it is also deprecated. C API functions `mysql_get_option()` and `mysql_options()` now write a deprecation warning to the standard error output when an application specifies `MYSQL_OPT_RECONNECT`. (WL #15766)

## Performance Schema Notes

- The type used for the Performance Schema `clone_status` table's `gtid_executed` column has been changed from `VARCHAR(4096)` to `LONGTEXT`. (Bug #109171, Bug #34828542)

## SQL Syntax Notes

- `CURRENT_USER()` can now be used as a default value for `VARCHAR` and `TEXT` columns in `CREATE TABLE` and `ALTER TABLE ... ADD COLUMN` statements.

  The functions `SESSION_USER()`, `USER()`, and `SYSTEM_USER()` are also supported in all of the cases just mentioned. By way of example, the following sequence of statements now works similarly to what is shown here, with the precise output dependent on your environment:

```
mysql> SELECT CURRENT_USER();
+-------------------+
| CURRENT_USER()    |
+-------------------+
|  sakila@localhost |
+-------------------+
1 row in set (0.00 sec)

mysql> CREATE TABLE t (
     >  c1 VARCHAR(288) DEFAULT (USER()),
     >  c2 VARCHAR(288) DEFAULT (CURRENT_USER()),
     >  c3 VARCHAR(288) DEFAULT (SESSION_USER()),
     >  c4 VARCHAR(288) DEFAULT (SYSTEM_USER())
     > );
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO t VALUES ROW();
Query OK, 1 row affected (0.01 sec)

mysql> TABLE t;
+-------------------+-------------------+-------------------+-------------------+
| c1                | c2                | c3                | c4                |
+-------------------+-------------------+-------------------+-------------------+
|  sakila@localhost |  sakila@localhost |  sakila@localhost |  sakila@localhost |
+-------------------+-------------------+-------------------+-------------------+
1 row in set (0.00 sec)
```

  When used in this way, these functions are also included in the output of `SHOW CREATE TABLE` and `SHOW COLUMNS`, and referenced in the `COLUMN_DEFAULT` column of the Information Schema `COLUMNS` table where applicable.

  If you need to insure that values having the maximum possible length can be stored in such a column, you should make sure that the column can accommodate at least 288 characters (255 for the user name and 32 for the host name, plus 1 for the separator `@`). For this reason—while it is possible to use one of these functions as the default for a `CHAR` column, it is not recommended due to the risk of errors or truncation of values. (Bug #17809, Bug #11745618)

## Functionality Added or Changed

- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated from OpenSSL 1.1.1 to OpenSSL 3.0. The exact version is now 3.0.9. More information on changes from 1.1.1 to 3.0 can be found at https://www.openssl.org/docs/man3.0/man7/migration_guide.html. (Bug #35475140, WL #15614)

- **Packaging:** Beginning with this release, binary archive packages previously named using `-el7-x86_64.tar` and `-el7-x86_64.tar.gz` are now named using `-linux-glibc2.17-x86_64.tar` and `-linux-glibc2.17-x86_64.tar.gz`, respectively. For example, if you used `mysql-8.0.33-el7-x86_64.tar` to install or upgrade to MySQL 8.0.33, you should use `mysql-8.0.34-linux-glibc2.17-x86_64.tar` for installing or upgrading to MySQL 8.0.34.

The naming change reflects the fact that these binaries are intended to be compatible with any Linux system using `glibc2.17` or newer; this includes Enterprise Linux 7, EL8, and EL9.

- Binary packages that include curl rather than linking to the system curl library have been upgraded to use curl 8.1.2. (Bug #35329529)

## Bugs Fixed

- **Important Change:** The default value of the `connection_memory_chunk_size` server system variable, when introduced in MySQL 8.0.28, was mistakenly set at 8912. This fix changes the default to 8192, which is the value originally intended. (Bug #35218020)
- **NDB Cluster:** The fix for a previous issue introduced a slight possibility of unequal string values comparing as equal, if any Unicode 9.0 collations were in use, and the collation hash methods calculated identical hash keys for two unequal strings. (Bug #35168702)

  References: See also: Bug #27522732. This issue is a regression of: Bug #30884622.
- **InnoDB:** Purging a large number of system threads sometimes led to congestion. (Bug #35289390, Bug #110685)
- **InnoDB:** `ddl::Aligned_buffer()` now uses the standard memory allocator rather than kernel memory management. (Bug #35194732, Bug #110411)
- **InnoDB:** An upgrade from MySQL 5.7 to MySQL 8.0.32 or 8.0.33 sometimes failed due to use of the deprecated system variables `innodb_log_file_size`, `innodb_log_files_in_group`, or both. To work around this problem prior to upgrading to MySQL 8.0.34, start the server with `--innodb-redo-log-capacity=206158430208`. (Bug #35155280)
- **InnoDB:** Changes in hashing functions made in MySQL 8.0.30 had an adverse effect on performance. (Bug #34870256)
- **InnoDB:** The rules used for aggregating entries in the redo log have been improved. (Bug #34752625, Bug #108944)
- **InnoDB:** In some cases, tablespace deletion caused issues with the buffer pool. (Bug #34330475, Bug #107689)
- **Packaging; Group Replication:** The group replication plugin from the Generic Linux packages did not load on some platforms that lacked a compatible version of `tirpc`. (Bug #35323208)
- **Replication:** Changes in `session_track_gtids` were not always propagated correctly. (Bug #35401212)
- **Replication:** By design, all DDL operations (including binary log operations such as purging the binary log) acquire a shared lock on the `BACKUP_LOCK` object, which helps to prevent simultaneous backup and DDL operations. For binary log operations, we checked whether any locks existed on `BACKUP_LOCK` but did not check the types of any such locks. This caused problems due to the fact that binary log operations should be prevented only when an exclusive lock is held on the `BACKUP_LOCK` object, that is, only when a backup is actually in progress, and backups should be prevented when purging the binary log.

  Now in such cases, instead of checking for locks held on the `BACKUP_LOCK` object, we acquire a shared lock on `BACKUP_LOCK` while purging the binary log. (Bug #35342521)
- **Replication:** In all cases except one, when `mysqlbinlog` encountered an error while reading an event, it wrote an error message and returned a nonzero exit code, the exception being for the active binary log

file (or any binary log where the format_description_log_event had the `LOG_EVENT_BINLOG_IN_USE_F` flag set), in which case it did not write a message, and returned exit code 0, thus hiding the error.

Now `mysqlbinlog` suppresses only those errors which are related to truncated events, and when doing so, prints a comment rather than an error message. This fix also improves the help text for the `--force-if-open` option. (Bug #35083373)

- **Replication:** Compressed binary log event handling was improved. (Bug #33666652)
- **Replication:** A transaction consisting of events each smaller than 1 GiB, but whose total size was larger than 1 GiB, and where compression did not make it smaller than 1 GiB, was still written to the binary log as one event bigger than 1 GiB. This made the binary log unusable; in effect, it was corrupted since neither the server nor other tools such as `mysqlbinlog` could read it.

  Now, when the compressed data grows larger than 1 GiB, we fall back to processing the transaction without any compression. (Bug #33588473)
- **Group Replication:** In a group replication setup, when there was a source of transactions other than the applier channel, the following sequence of events was possible:

  1. Several transactions being applied locally were already certified, and so were associated with a ticket, which we refer to as Ticket 2, but had not yet been committed. These could be local or nonlocal transactions.
  2. A view is created with Ticket 3, and must wait on transactions from Ticket 2.
  3. The view change (VC1) entered the GR applier channel applier and waited for the ticket to change to 3.
  4. Another group change, and another view change (VC2), occurred while the transactions from Ticket 2 were still completing.

  This gave rise to the following issue: There was a window wherein the last transaction from Ticket 2 had already marked itself as being executed but had not yet popped the ticket; VC2 popped the ticket instead but never notified any of the participants. This meant that VC1 continued to wait indefinitely for the ticket to change, and with the additional effect that the worker could not be killed.

  We fix this by checking for the need to break each second so that this loop is responsive to changes in the loop condition; we also register a new stage, so that the loop is more responsive to kill signals. (Bug #35392640)

  References: See also: Bug #35206392, Bug #35374425.
- **Group Replication:** Removed a memory leak discovered in `Network_provider_manager::open_xcom_connection()`. (Bug #34991101)
- **Group Replication:** When a group action was sent to the group and the connection was killed on the coordinator, group members were in different states, with members which received the coordinated action waiting for the member that executed it, and the member which started execution having nothing to process, which caused problems with coordination of the group.

  Now in such cases, we prevent this issue from occurring by causing group actions to wait until all members have completed the action. (Bug #34815537)
- **Group Replication:** Cleanup of resources used by OpenSSL connections created indirectly by group replication was not carried out as expected at all times. We fix this by adding cleanup functionality that can be called at any time such connections are created by group replication. (Bug #34727136)

- **JSON:** When the result of `JSON_VALUE()` was an empty string and was assigned to a user variable, the user variable could in some cases be set to `NULL` instead, as shown here:

```
mysql> SELECT JSON_VALUE('{"fname": "Joe", "lname": ""}', '$.lname') INTO @myvar;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT @myvar = '', @myvar IS NULL;
+-------------+----------------+
| @myvar = '' | @myvar IS NULL |
+-------------+----------------+
|        NULL |              1 |
+-------------+----------------+
1 row in set (0.00 sec)
```

  With this fix, the query just shown now returns `(1, 0)`, as expected. (Bug #35206138)

- **JSON:** Some JSON schemas were not always processed correctly by `JSON_SCHEMA_VALID()`. (Bug #109296, Bug #34867398)

- In rare cases, MySQL server could exit rather than emit an error message as expected. (Bug #35442407)

- The internal resource-group enhancement added in MySQL 8.0.31 and refactored in MySQL 8.0.32 is now reverted. (Bug #35434219)

  References: Reverted patches: Bug #34702833.

- An in-place upgrade from MySQL 5.7 to MySQL 8.0, without a server restart, could result in unexpected errors when executing queries on tables. This fix eliminates the need to restart the server between the upgrade and queries. (Bug #35410528)

- A fix in MySQL 8.0.33 made a change for `ORDER BY` items already resolved so as not to resolve them again (as is usually the case when a derived table is merged), but this did not handle the case in which an `ORDER BY` item was itself a reference. (Bug #35410465)

  References: This issue is a regression of: Bug #34890862.

- Changes in `session_track_gtids` were not always handled correctly. (Bug #35401212)

- Some pointers were not always released following statement execution. (Bug #35395965)

- Some instances of subqueries within stored routines were not always handled correctly. (Bug #35377192)

- Fortified parsing of the network packet data sent by the server to the client. (Bug #35374491)

- Encryption enhancements now strengthen compliance and remove the use of deprecated APIs. (Bug #35339886)

- When a column reference given by table name and column name was looked up in the function `find_item_in_list()`, we ignored that the item searched for might not have a table name, as it was not yet resolved. We fix this by making an explicit check for a null table name in the sought-after item. (Bug #35338776)

- Deprecated the `lz4_decompress` and `zlib_decompress` command-line utilities that exist to support the deprecated `mysqlpump` command-line utility. (Bug #35328235)

- Queries using `LIKE '%...%'` ran more poorly than in previous versions of MySQL. (Bug #35296563)

- In `Bounded_queue::push()`, when `Key_generator::make_sortkey()` returns `UINT_MAX` (error), then no key has been produced; now when this occurs, we no longer update the internal queue.

As part of this fix, `push()` now returns true on error. (Bug #35237721)

- The `authentication_oci` plugin is fixed to allow federated and provisioned users to connect to a DB System as a mapped Proxy User using an ephemeral key-pair generated through the OCI CLI. (Bug #35232697)
- Some queries using common table expressions were not always processed correctly. (Bug #35231475)
- The internal function `compare_pair_for_nulls()` did not always set an explicit return value. (Bug #35217471)
- Removed the clang-tidy checks that clash with the MySQL coding style. (Bug #35208735)
- Some subqueries using `EXISTS` in both the inner and outer parts of the query were not handled correctly. (Bug #35201901)
- Rotated audit log files now always reset the ID value of the bookmark to zero, rather than continuing the value from the previous file. (Bug #35200070)
- Errors were not always propagated correctly when evaluating items to be sorted by filesort. (Bug #35195181)

  References: See also: Bug #35145246.
- The fix for a previous issue with `ROLLUP` led to a premature server exit in debug builds. (Bug #35168639)

  References: This issue is a regression of: Bug #33830659.
- Simplified the implementation of `Item_func_make_set::val_str()` to make sure that we never try to reuse any of the input arguments, always using the local string buffer instead. (Bug #35154335, Bug #35158340)
- When transforming subqueries to a join with derived tables, with the containing query being grouped, we created an extra derived table in which to do the grouping. This process moved the initial select list items from the containing query into the extra derived table, replacing all of the original select list items (other than subqueries, which get their own derived tables) with columns from the extra derived table.

  This logic did not handle `DEFAULT` correctly due to the manner in which default values were modelled internally. This fix adds support for `DEFAULT(`*`expression`*`)` in queries undergoing the transform previously mentioned. This fix also solves an issue with item names in metadata whereby two occurrences of the same column in the select list were given the same item name as a result of this same transform. (Bug #35150085, Bug #35101169)
- A query of the form `SELECT * FROM t1 WHERE (SELECT a FROM t2 WHERE t2.a=t1.a + ABS(t2.b)) > 0` should be rejected with `Subquery returns more than 1 row`, but when the `subquery_to_derived` optimization was enabled, the transform was erroneously applied and the query returned an incorrect result. (Bug #35101630)
- Handling of certain potentially conflicting `GRANT` statements has been improved. (Bug #35089304)
- A query using both `MEMBER OF()` and `ORDER BY DESC` returned only a partial result set following the creation of a multi-valued index on a JSON column. This is similar to an issue fixed in MySQL 8.0.30, but with the addition of the `ORDER BY DESC` clause to the prblematic query. (Bug #35012146)

  References: See also: Bug #106621, Bug #33917625.

- The debug server asserted on certain operations involving `DECIMAL` values. (Bug #34973932)
- All instances of adding and replacing expressions in the select list when transforming subqueries to use derived tables and joins have been changed so that their reference counts are maintained properly. (Bug #34927110)
- Index Merge (see Index Merge Optimization) should favor ROR-union plans (that is, using RowID Ordered Retrieval) over sort-union plans if they have similar costs, since sort-union requires additionally sorting of the rows by row ID whereas ROR-union does not.

  For each part of a `WHERE` clause containing an `OR` condition, the range optimizer gets the best range scan possible and uses all these range scans to build an index merge scan (that is, a sort-union scan). If it finds that all the best range scans are also ROR-scans, the range optimizer always proposes a ROR-union scan because it is always cheaper than a sort-union scan. Problems arose when the best range scan for any one part of an `OR` condition is not a ROR-scan, in which case, the range optimizer always chose sort-union. This was true even in cases, where it might be advantageous to choose a ROR-scan (even though it might not be the best range scan to handle one part of the OR condition), since this would eleminate any need to sort the rows by row ID.

  Now, in such cases, when determining the best range scan, the range optimizer also detects whether there is any possible ROR-scan, and uses this information to see whether each part of the `OR` condition has at least one possible ROR-scan. If so, we rerun the range optimizer to obtain the best ROR-scan for handling each part of the `OR` condition, and to make a ROR-union path. We then compare this cost with the cost of a sort-union when proposing the final plan. (Bug #34826692, Bug #35302794)
- Selecting from a view sometimes raised the error `Illegal mix of collations ... for operation '='` when the collation used in the table or tables from which the view definition selected did not match the current session value of `collation_connection`. (Bug #34801210)
- Valid MySQL commands (`use` and `status`) and C API functions (`mysql_refresh`, `mysql_stat`, `mysql_dump_debug_info`, `mysql_ping`, `mysql_set_server_option`, `mysql_list_processes`, and `mysql_reset_connection`) could write an error message to the audit log, even though running the command or calling the function emitted no such error. (Bug #33966181)
- Increased the maximum fixed array size to 8192 instead of 512. This fixes an issue with `mysqladmin` extended status requests, which can exceed 512 entries.

  Our thanks to Meta for the contribution. (Bug #30810617)
- The `mysqldump --column-statistics` option attempted to select from `information_schema.column_statistics` against MySQL versions before 8.0.2, but this now generates the `warning column statistics not supported by the server` and sets the option to false.

  Our thanks to Meta for the contribution. (Bug #28782417)
- The function used by MySQL to get the length of a directory name was enhanced. (Bug #28047376)
- Executing a query with an implicit aggregation should return exactly one row, unless the query has a `HAVING` clause that filters out the row, but a query with a `HAVING` clause which evaluated to `FALSE` sometimes ignored this, and returned a row regardless. (Bug #14272020)
- The presence of an unused window function in a query, along with an `ORDER BY` that could have been eliminated, led to an unplanned server exit. (Bug #111585, Bug #35168639, Bug #35204224, Bug #35545377)

  References: This issue is a regression of: Bug #35118579.

- `ORDER BY RANDOM_BYTES()` had no effect on query output. (Bug #111252, Bug #35148945, Bug #35457136)
- Fixed an issue which could occur when loading user-defined functions. (Bug #110576, Bug #35242734)
- Concurrent execution of `FLUSH STATUS`, `COM_CHANGE_USER`, and `SELECT FROM I_S.PROCESSLIST` could result in a deadlock. A similar issue was observed for concurrent execution of `COM_STATISTICS`, `COM_CHANGE_USER`, and `SHOW PROCESSLIST`.

  Our thanks to Dmitry Lenev for the contribution. (Bug #110494, Bug #35218030)
- The `mysqldump` utility could generate invalid INSERT statements for generated columns. (Bug #110462, Bug #35208605)
- During optimization, range-select tree creation uses logic which differs based on the left-hand side of the `IN()` predicate. For a field item, each value on the right-hand side is added to an OR tree to create the necessary expression. In the case of a row item comparison (example: `WHERE (a,b) IN ((n1,m1), (n2, m2), ...)`), an expression in disjunctive normal form (DNF) is needed. A DNF expression is created by adding an `AND` tree with column values to an `OR` tree for each set of RHS values, but instead the `OR` tree was added to the `AND` tree causing the tree merge to require exponential time due to $O(n^2)$ runtime complexity. (Bug #108963, Bug #34758905)
- When using `SELECT` to create a table and the statement has an expression of type `GEOMETRY`, MySQL could generate an empty string as the column value by default. To resolve this issue, MySQL no longer generates default values for columns of type `GEOMETRY` under these circumstances. Our thanks to Tencent for the contribution. (Bug #107996, Bug #34426943)
- For index skip scans, the first range read set an end-of-range value to indicate the end of the first range, but the next range read did not clear the stale end-of-range value and applies this stale value to the current range. Since the indicated end-of-range boundary had already been crossed in the previous range read, this caused the reads to stop, causing multiple rows to be missed in the result.

  We fix this by making sure in such cases that the old end-of-range value is cleared. (Bug #107460, Bug #34235624, Bug #34982949)

# Changes in MySQL 8.0.33 (2023-04-18)


- Audit Log Notes
- Compilation Notes
- Component Notes
- Deprecation and Removal Notes
- MySQL Enterprise Notes
- Performance Schema Notes
- Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Audit Log Notes

- MySQL Enterprise Audit previously used tables in the `mysql` system database for persistent storage of filter and user account data. For enhanced flexibility, the new `audit_log_database` server system

variable now permits specifying other databases in the global schema namespace at server startup. The `mysql` system database is the default setting for table storage. (WL #15500)

## Compilation Notes

- **Microsoft Windows:** Added MSVC Code Analysis support for Visual Studio 2017 and higher. This adds a new MSVC_CPPCHECK (defaults to OFF) CMake option that either enables or disables this analysis on the current directory and its subdirectories. (Bug #34828882)
- Downgraded curl deprecation warnings to *-Wno-error* for curl versions greater than 7.86 when MySQL is built with a GNU compiler or clang. (Bug #35111625)
- On macOS, added *-framework CoreFoundation* and *-framework SystemConfiguration* when linking the curl interface to link with shared system libraries as needed. (Bug #35104962)
- Replaced the `MY_INCLUDE_SYSTEM_DIRECTORIES` macro with library interfaces. (Bug #35018072, Bug #35028089, Bug #35072295)
- Improved CMake code to support alternative linkers. (Bug #34963568)
- Removed the deprecated `Docs/mysql.info` file from the build system. (Bug #34960126)
- Added a top-level `.clang-tidy` file and associated `.clang.tidy` files in the `strings/` and `mysys/` directories. Also enabled *compdb* support to enable clang-tidy usage on header files. (Bug #34917075)
- Removed several unmaintained or unused C++ source files for functionality such as uca-dump and uctypedump. (Bug #34898978)
- Added a CMake build option to enable colorized compiler output for GCC and Clang when compiling on the command line. To enable, pass `-DFORCE_COLORED_OUTPUT=1` to CMake. (Bug #34897192)
- On Windows, also install .pdb files for associated .dll files if they are found for 3rd-party libraries. (Bug #34863555)
- Enterprise Linux 8 and Enterprise Linux 9 builds now use GCC 12 instead of GCC 11. (Bug #34829151)
- Building with -static-libgcc -static-libstdc++ now also builds the bundled protobuf with static libraries, as required.

  Thanks to Alex Xing for the contribution. (Bug #110216, Bug #35123848)

## Component Notes

- `INSTALL COMPONENT` now includes the `SET` clause, which sets the values of component system variables while installing one or more components. The new clause reduces the inconvenience and limitations associated with the other ways of assigning variable values. For usage information, see INSTALL COMPONENT Statement. (WL #10916)

## Deprecation and Removal Notes

- User-defined collations (see Adding a Collation to a Character Set) are now deprecated. Either of the following now causes a warning to be written to the log:
  - Any occurrence of `COLLATE` followed by the name of a user-defined collation in an SQL statement
  - Use of the name of a user-defined collation as the value of `collation_server`, `collation_database`, or `collation_connection`

You should expect support for user-defined collations to be removed in a future version of MySQL. (WL #14277)

## MySQL Enterprise Notes

- MySQL Enterprise Edition now provides data masking and de-identification capabilities based on components, rather than being based on a plugin library that was introduced in MySQL 8.0.13. The component implementation provides dedicated privileges to manage dictionaries and extends the list of specific types to include:

  - Canada Social Insurance Number
  - United Kingdom National Insurance Number
  - International Bank Account Number
  - Universally Unique Identifier (UUID)

  An improved table-based dictionary registry replaces the file-based dictionary used by the plugin. For a summary of the differences between the component and plugin implementations, see Data-Masking Components Versus the Data-Masking Plugin. Existing plugin users should uninstall the server-side plugin and drop its loadable functions before installing the new MySQL Enterprise Data Masking and De-Identification components. (Bug #33851601, WL #12641)

## Performance Schema Notes

- The Performance Schema Server Telemetry Traces service is added in this release. An interface which provides plugins and components a way to retrieve notifications related to SQL statements' lifetime.

  For more information on this interface, see the *Server telemetry traces service* section in the MySQL Source Code documentation.

  The following were added:

  - Status variable `Telemetry_traces_supported`. Whether server telemetry traces is supported. (Boolean)
  - `TELEMETRY_ACTIVE` column was added to the `threads` table. Indicates whether the thread has an active telemetry session attached.

  (WL #15059)

## Plugin Notes

- The length of a digest for the query rewrite plugin should be 32 bytes, but this was not always enforced. (Bug #34991760)

## Functionality Added or Changed

- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.1.1t. Issues fixed in OpenSSL version 1.1.1t are described at https://www.openssl.org/news/cl111.txt. (Bug #35092429)
- **Replication:** As part of ongoing work to change old terminology used in MySQL products, the terms “master”, “slave”, and “MTS” have been replaced in error messages relating to MySQL Replication by “source”, “replica”, and “MTA”, respectively. This includes all error messages listed in

`messages_to_clients.txt` and `messages_to_error_log.txt` relating to replication; the present task does not perform this replacement for messages used in other contexts.

See the MySQL 8.0 Error Message Reference, for more information. (Bug #108422, Bug #34594819, WL #14191)

- **Replication:** `mysqlbinlog --start-position` now accepts values up to 18446744073709551615, unless the `--read-from-remote-server` or `--read-from-remote-source` option is also used, in which case the maximum is 4294967295. (Bug #77818, Bug #21498994)
- Binary packages that include curl rather than linking to the system curl library have been upgraded to use curl 7.88.1. (Bug #34828111)
- The use of a generated column with `DEFAULT(col_name)` to specify the default value for a named column is not permitted and now emits an error message. (Bug #34463652, Bug #34369580)
- The new `TELEMETRY_LOG_ADMIN` privilege now enables telemetry log configuration. This privilege is defined by the `telemetry_log` plugin, which is deployed exclusively through MySQL HeatWave on AWS. (Bug #111395, Bug #35494180)
- It is now possible to return the Statement ID to the client for successfully executed statements. To enable this per session, add `statement_id` to the value of `session_track_system_variables`, or set `session_track_system_variables` to the special value `*` (asterisk). (WL #15418)

## Bugs Fixed

- **NDB Cluster:** Occasional temporary errors which could occur when opening a table from the `NDB` dictionary while repeatedly performing concurrent schema operations were not retried. (Bug #34843889)
- **NDB Cluster:** During iteration, ordered index scans retain a cursor position within each concurrently scanned ordered index fragment. Ordered index fragments are modified and balanced as a result of committing DML transactions, which can require scan cursors to be moved within the tree. When running with query threads configured (`AutomaticThreadConfig` set to 1), multiple threads can access the same index fragment tree structure, and the scans of multiple threads can have their cursors present in the same structure.

  The current issue arose due to an assumption in the logic for moving scan cursors when committing DML operations that all scan cursors belonged to the LDM thread owning the index fragment, which did not allow for the possibility that such fragments might belong to query threads. (Bug #33379702)

  References: See also: Bug #32257063.
- **InnoDB:** Error messages related to `innodb_doublewrite` now appear in the error log. (Bug #34883045, Bug #109330)
- **InnoDB:** Prevent online DDL operations from accessing out-of-bounds memory. (Bug #34750489, Bug #108925)
- **InnoDB:** `ALTER TABLE ... AUTO_INCREMENT` could be set to less than MAX + 1 and not forced to MAX + 1. (Bug #33419246, Bug #105092)
- **InnoDB:** `Innodb_data_pending_fsyncs` could show extremely high inaccurate values because of a variable overflow. (Bug #30133150)
- **Partitioning:** Some `IN()` queries on partitioned tables were not always handled correctly. (Bug #34801284)

  References: This issue is a regression of: Bug #32311183.

- **Partitioning:** Queries using the `INDEX_MERGE` optimizer hint was not handled correctly in all cases. (Bug #34797257)

- **Replication:** XA transactions whose XIDs contained null bytes could not be recovered. (Bug #34918985)

- **Replication:** When `binlog_order_commits` was set equal to `1`, for any two transactions and for any sub-step of the commit phase, the transaction that was written to the binary log first did not always execute the sub-step first, as expected. (Bug #34703698)

- **Replication:** Some binary log events were not always handled correctly. (Bug #34617506)

- **Replication:** The binary log recovery process did not report all possible error states. (Bug #33658850)

- **Replication:** Following `CHANGE REPLICATION SOURCE TO SOURCE_CONNECTION_AUTO_FAILOVER=1`, failover generated a number of misleading warnings in the log that implied there were problems when in fact conditions were those expected for such a failover. These log messages have been updated accordingly. (Bug #32135376)

- **Replication:** When a transaction failed, as a side effect, extraneous error messages relating the replication data repositories were written to the log. Now in such cases, we suppress such error messages, which are not directly related to the issue of the failed transaction or its cause. (Bug #19820134)

- **Replication:** Setting `binlog_order_commits` to `OFF` could lead to a missed GTID in the next binary log file's `Previous_gtids` event.

  Our thanks to Yewei Xu and the Tencent team for the contribution. (Bug #109485, Bug #34930969)

- **Replication:** Corrected the SQL statements suggested in the error message text for `ER_RPL_REPLICA_ERROR_RUNNING_QUERY`.

  Our thanks to Dan McCombs for the contribution. (Bug #109154, Bug #34822612)

- **Replication:** A hash scan builds a hash of changes, scans the target table or index, and applies any matching change for the current entry. In the build phase, it uses only the before image, and skips any after image. Problems arose in some cases because generated columns were computed for the (skipped) after image, leading to replication errors. This is fixed by not computing generated columns any longer for seek-only calls such as hash scans.

  Our thanks to dc huang for the contribution. (Bug #107366, Bug #34198907)

- **Replication:** In certain rare cases, it was possible to set `gtid_mode=OFF` for one session while another session, after `WAIT_FOR_EXECUTED_GTID_SET()` was issued by a user in this second session, was still waiting for the next GTID set from the first session. This could result in the second session waiting indefinitely for the function to return. (Bug #99921, Bug #31505993)

- **Group Replication:** Accessing the Performance Schema `replication_group_communication_information` and `replication_group_member_stats` tables in parallel sometimes caused subsequent group replication operations to hang. (Bug #34870181)

- **Group Replication:** In certain cases, the group replication secondary node unexpectedly shut down while purging the relay log. (Bug #34397106)

- **Group Replication:** When shutting down the Group Replication plugin, the order in which the associated events were reported the error log sometimes led to confusion. To remove any doubts, we now make sure that `Plugin group_replication reported: 'Plugin 'group_replication'`

`has been stopped.` is in fact the last log message relating to the shutdown, written only when all other events associated with shutting down the plugin have been logged. (Bug #109345, Bug #34887491)

- **Microsoft Windows:** The `authentication_fido_client` plugin stopped responding during the authentication process if it was unable to find a FIDO device on the Windows client host. (Bug #34918044)

- In certain cases, `CONVERT(utf8mb3_column USING UTF16)` was rejected with the error `Cannot convert string '\x--...' from binary to utf16`. (Bug #35129361)

- When joining two tables on a string column, and the column from one of the tables has an additional predicate comparing it with a temporal literal, constant propagation in some cases incorrectly caused the join condition to be modified such that it used temporal rather than string semantics when comparing the strings. This caused incorrect results to be returned from the join. (Bug #35115909)

- Error messages returned after calling the `mysql_reset_connection()` C API function in a prepared statement did not identify the function name properly. (Bug #35107280)

- Fixed a regression in a previous fix for an issue with windowing functions.

  Our thanks to Dmitry Lenev for the contribution. (Bug #35061924)

  References: This issue is a regression of: Bug #34572136.

- When replacing subqueries in transforms, the internal flag showing whether a given query block contains any subqueries (`PROP_SUBQUERY`) was not updated afterwards. (Bug #35060385)

- A client setting the character set to an impermissible client character set (`ucs2`, `utf16`, `utf16le`, or `utf32`) could cause unexpected behavior when the client used an authentication plugin. (Bug #35054579)

- `EXPLAIN ANALYZE` displayed 0 when the average number of rows was less than 1. To fix this, we now format numbers in the output of `EXPLAIN ANALYZE` and `EXPLAIN FORMAT=TREE` such that numbers in the range 0.001-999999.5 are printed as decimal numbers, and numbers outside this range are printed using engineering notation (for example: `1.23e+9`, `934e-6`). In addition, trailing zeroes are no longer printed, and numbers less than `1e-12` are printed as `0`.

  This helps ensure consistent precision regardless of the number's value and improve readability, while producing minimal rounding errors. (Bug #34990948)

- The `NTILE()` function did not work correctly in all cases. (Bug #34986665)

- Some joins on views did not perform correctly. (Bug #34985359)

- Transforming a correlated scalar subquery to a derived table led to a wrong result for `InnoDB` tables when the subquery included duplicate predicates. An example of a query which could be affected by this issue is shown here:

```
SELECT * FROM t1
WHERE (
  SELECT t2.a FROM t2
  WHERE t2.a = t1.a AND t2.a = t1.a
    ) > 0;
```

  (Bug #34973220)

- Fixed an assert in `sql/item_strfunc.cc` that could potentially lead to issues with the `SPACE()` function. (Bug #34962821)

- Using `ROW_COUNT()` as the *`length`* argument to `LPAD()` or `RPAD()` did not perform as expected. (Bug #34961236)
- A query with a window function having an expression with a `CASE` function in its `ORDER BY` clause could lead to a server exit. (Bug #34933045)
- The fix for a previous issue introduced an assertion in debug builds when optimizing a `HAVING` clause. (Bug #34923792)

  References: This issue is a regression of: Bug #33725415.
- When using `mysqld_multi`, the system that obscures "--password" usage as "--password=*****" would also match "--password-history" and "--password-require-current" definitions as "--password", but now explicitly checks for "--password=" instead. (Bug #34918740)
- In some cases, calling the `mysql_bind_param()` C API function could cause the server to become unresponsive. (Bug #34869076)
- The `authentication_oci_client` plugin was unable to open a valid configuration file if any of its entries contained an equals sign character separated by spaces (for example, `key_file = /home/user/.oci/oci_api_key.pem`). Now, both '*`key`*=*`value`*' and '*`key`* = *`value`*' entry formats are supported. (Bug #34864078)
- Incorrect results were returned when the result of an `INTERSECT` or `EXCEPT` operation was joined with another table. This issue affected these operations in such cases when used with either `DISTINCT` or `ALL`. (Bug #34843764)
- When preparing a view query, the operation used the system character set (instead of the character set stored in data dictionary) and then reported an invalid character-string error. (Bug #34800905)
- Prepared statements that operate on derived tables, including views, could stop unexpectedly due to problems with the code for reopening tables after an error. (Bug #34798403)
- Removed an assertion raised in certain cases by the `RANDOM_BYTES()` function in debug builds. (Bug #34781507)
- There was an issue in how persisted variables were set on startup, causing certain variables not to get properly set to their persisted value. (Bug #34751419)
- The `MAKETIME()` function did not perform correctly in all cases. (Bug #34745241)
- Some functions with multiple arguments did not produce the expected results. (Bug #34741801)
- A table reference in an `ORDER BY` outside the parenthesized query block in which the table was used, and which query block had no `LIMIT` or `ORDER BY` of its own, raised an error. (Bug #34704011)

  References: This issue is a regression of: Bug #103954, Bug #32980249.
- A left join with an impossible condition as part of an `ON` clause was not optimized as in MySQL 5.7, so that, in later versions, the query executed more quickly without the impossible condition than with it. An example of such a query, impossible condition included, is `SELECT * FROM t1 JOIN t2 ON t1.c1=t2.c1 AND 1=2`. (Bug #34668756)
- When a user defined function was part of a derived table that was merged into the outer query block, or was part of a subquery converted to a semi-join, knowledge of whether this UDF was deterministic (or not) was lost during processing. (Bug #34666531)
- With JSON logging enabled and an event subclass specified in the audit log filter definition, an empty item (`"" : { }`) was appended to the end of the logged event. (Bug #34659904)

- Some subqueries did not execute properly. (Bug #34616553)
- After the `asymmetric_encrypt()` component function in a `SELECT` query encountered a NULL field to decrypt, it could return NULL values for other non-NULL encrypted fields. (Bug #34598912)
- The server did not always shut down cleanly after uninstalling the audit log plugin. (Bug #34594035)
- Certain antijoins were not handled correctly by the server. (Bug #34370673)

  References: This issue is a regression of: Bug #30573446.
- When the MySQL 5.7 Optimizer has 2 choices for an index to filter rows, one primary and one secondary, it picks a range scan on the secondary index because the range scan uses more key parts. MySQL 8.0 did not use this logic, instead choosing the primary index to filter rows with `WHERE` clause filtering. Primary key use is not suitable in such cases due to the presence of `LIMIT`, and due to the nature of data distribution. The secondary index was not considered while resolving order by due to constant elimination. This resulted in much different query plans in MySQL 5.7 and MySQL 8.0 for the same query.

  We solve this issue in MySQL 8.0 by skipping the constant key parts of the index during order-by evaluation only if the query is constant-optimized, which can be done at this time, but not during `LIMIT` analysis. (Bug #34291261)
- The MySQL data dictionary caches failed lookups of `se_private_id` values (IDs which are not found), which speeds up execution of code specific to `InnoDB`, relying on the fact that `InnoDB` does not reuse these IDs. This assumption does not necessarily hold for other storage engines, most notably `NDB`, where this problem was resolved previously by not using this cache.

  We extend the previous fix made for `NDB` so that the cache lookup is now employed only when the table uses the `InnoDB` storage engine. (Bug #34145006)

  References: See also: Bug #33824058.
- Unexpected results were seen in some queries using `DENSE_RANK()`, possibly with the addition of `WITH ROLLUP`. (Bug #34099408)
- Fixed an assert raised in `sql/sql_tmp_table.cc` following work done previously to reimplement `ROLLUP` processing. (Bug #33830659)

  References: This issue is a regression of: Bug #30969045.
- Some CTEs that did not use any tables were not always handled correctly. (Bug #33725542)

  References: This issue is a regression of: Bug #27062031.
- Accessing rows from a window frame of a window function call present only in the query's `ORDER BY` list raised an error. (Bug #33069747)
- `PERCENT_RANK()` used with `ORDER BY` *`column`* did not return the correct result. (Bug #33064174)

  References: This issue is a regression of: Bug #27484133.
- The `--exclude-tables` and `--include-tables` `mysqlpump` options did not handle views. (Bug #21303549)
- Changed the MySQL systemd service unit configuration from `After=network-online.target` to `Wants=network-online.target` to ensure that all configured network devices are available and have an IP address assigned before the service is started. (Bug #109996, Bug #35068274)

- `AVG(...) OVER (ROWS BETWEEN 1 FOLLOWING AND UNBOUNDED FOLLOWING)` did not return the correct result. (Bug #109725, Bug #35013880)

  References: This issue is a regression of: Bug #108008, Bug #34431996.

- A query of the form `SELECT 1 FROM t1 WHERE NOT EXISTS (VALUES ROW(1), ROW(2))` caused an assert in debug builds when the `subquery_to_derived` optimizer switch was enabled. (Bug #109723, Bug #35014318)

  References: See also: Bug #108910, Bug #34746261.

- `mysqlimport` did not escape reserved word table names when used with the `--delete` option. (Bug #109711, Bug #34999015)

- When cloning a condition to push down to a derived table, characters in strings representing conditions were converted to `utfmb4` correctly only for values less than 128 (the ASCII subset), and code points outside the ASCII subset were converted to invalid characters, causing the resulting character strings to become invalid. For derived tables without `UNION`, this led to problems when a column name from the derived table used characters outside the ASCII subset, and was used in the `WHERE` condition. For derived tables with `UNION`, it created problems when a character outside the ASCII subset was present in a `WHERE` condition.

  We fix these issues by initializing the string used for representing the condition in such cases to the connection character set. (Bug #109699, Bug #34996488)

- Using `--single-transaction` with `mysqldump` version 8.0.32 required either the `RELOAD` or `FLUSH_TABLES` privilege. This requirement now applies only when both `gtid_mode=ON` (default `OFF`) and with `--set-gtid-purged = ON|AUTO` (default `AUTO`). (Bug #109685, Bug #109701, Bug #34993824, Bug #34998910, Bug #35020512)

  References: This issue is a regression of: Bug #105761, Bug #33630199.

- Fixed a number of issues present in the internal documentation for the scramble generator algorithm in `sha256_scramble_generator.cc` and `sha2_password_common.cc`.

  Our thanks to Niklas Keller for the contribution. (Bug #109576, Bug #34967141)

- `CREATE USER IF NOT EXISTS` added a password history entry even when the user already existed and the password was not updated. This caused a subsequent `ALTER USER` statement to be rejected. (Bug #109415, Bug #34906592)

- Many joins using `eq_ref` access did not perform as well as in previous versions. This issue was first reported in MySQL 8.0.29. (Bug #109361, Bug #34891365)

- A hash outer join sometimes incorrectly matched `NULL` with a decimal zero or an empty string that used a non-padding collation, leading to erroneous results. (Bug #109211, Bug #34837464)

  References: This issue is a regression of: Bug #33794977.

- An object used internally by `ALTER INSTANCE RELOAD TLS` was not freed until the number of readers reached 0, under the assumption is that the number of readers should reach 0 fairly frequently. The read lock held during an SSL handshake is generally an expensive operation, with network calls, so when round trips between the client and the server took excessively long, the lock was held for a relatively long amount of time. This meant that, when changing the value of this object and there were a sufficient number of incoming SSL connections being made, the number of readers might not reach 0 in a reasonable length of time, leaving the thread holding the lock using 100% of the CPU until the lock was released.

We fix this by adding a wait after setting the pointer to this object to a new value, but before releasing the old object.

Our thanks to Sinisa Milivojevic for the contribution. (Bug #107567, Bug #34284186)

- If `mysqldump` or `mysqlpump` could not convert a field's default value to UTF-8 (for instance, if the field was of type `BINARY` and the default value did not coincide with valid UTF-8), the operation produced results that were not valid to import. Further, using the `--hex-blob` option did not resolve the issue. We now convert the default value to the system character set. If this fails, the server sends the value as hexadecimal instead to make it more human-readable. (Bug #104840, Bug #33322551)

- A connection using the C API (`libmysqlclient`) client library could fail with the `FUTURE` crypto policy. (Bug #104778, Bug #33316709)

- While cloning a temporary table for a common table expression which used shared materialization, the cloned temp table was not marked as using hash deduplication, leading to wrong results. We now set the hash field for the cloned temporary table correctly, and update the hidden field count to take this into account. (Bug #102251, Bug #32383712)

  References: See also: Bug #103052, Bug #32659629.

- `CREATE EVENT` and `ALTER EVENT` assumed that all values passed to them (other than in a `DO` clause) resolved as scalars without actually checking the values. This led to assertions when any such values actually rows.

  We now perform an explicit check for the number of columns when resolving such items, and report an error when one produces a row and not a scalar value. (Bug #57552, Bug #11764690)

- A view reference whose underlying field is a constant is not marked as constant when the reference is part of an inner table of an outer join. It was found that, when pushing a condition down to a derived table, the reference was stripped off and only the underlying field was cloned, which made it a constant, and led to wrong results.

  To fix this problem, we ensure that we do not push such a condition down to the derived table by adding a check to see first whether the table used by the condition matches the derived table or is a constant expression; only when it is one or the other of these do we actually push the condition down. (Bug #34661, Bug #11747971)

# Changes in MySQL 8.0.32 (2023-01-17)

- Authentication Notes
- C API Notes
- Compilation Notes
- Deprecation and Removal Notes
- Generated Invisible Primary Keys (GIPKs)
- Keyring Notes
- Pluggable Authentication
- Spatial Data Support
- SQL Syntax Notes

- Thread Pool Notes
- Functionality Added or Changed
- Bugs Fixed

## Authentication Notes

- The server could return `LDAP_OPERATIONS_ERROR` for LDAP authentication failures, rather than only for actual LDAP server errors such as when an AD domain is not accessible. Now, the server returns `LDAP_AUTHENTICATION_ERROR`, a MySQL-specific error code, to indicate authentication errors. (Bug #100333, Bug #31680279)

## C API Notes

- The `sha256_password_auth_client_nonblocking()` function always returned an error, even when the public key was available. (See the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html.) Our thanks to Facebook for the several fixes in this patch. (Bug #34556764)

## Compilation Notes

- **Microsoft Windows:** The `authentication_ldap_sasl` server plugin is no longer built for Windows as only the client is supported for SASL-based LDAP authentication. (Bug #34448155)
- On Windows, compiling MySQL server using VS 2022 would emit an error about two projects named "parser-t" if tests and the NDB storage engine were enabled. The tests were renamed to avoid conflict on case-insensitive operating systems. (Bug #34790413)
- On MacOS, silenced deprecation warnings generated by Xcode 14; this includes suggestions to use `snprintf(3)` instead of `sprintf(3)`, and warnings about possible loss of precision when converting from 64-bit to 32-bit integers. (Bug #34776172)
- Removed the boost library usage from the plugins. (Bug #34694419)
- Removed all 3rd party files named '`Makefile`' as they were not used. (Bug #34648199)
- Added clang 15 support. (Bug #34638573)
- Located and removed unused code; located it using fastcov. (Bug #34583577)
- Improved code related to building the ndbcluster plugin by fixing warnings generated with 'gcc 11.2.0 RelWithDebInfo on Ubuntu 22.04' and 'gcc 8.3.1 on el6'. (Bug #34384889)
- Now use full file paths for the Bison and Flex source files to help simplify debugging and gcov reports. (Bug #109022, Bug #34776151)
- Building MySQL would fail if the building user lacked access to the `mysqld` temporary directory. Now `--no-defaults` is used when creating the `INFO_BIN` file. (Bug #108947, Bug #34756282)

## Deprecation and Removal Notes

- Use of the dollar sign (`$`) as the first character of an unquoted identifier is now deprecated, raises a warning (`ER_WARN_DEPRECATED_SYNTAX_NO_REPLACEMENT`), and is subject to removal in a future release.

  This affects statements using such an identifier for the name of any database, table, view, column, stored program, or alias. Identifiers beginning with a dollar sign are still permitted only when they are

quoted—that is, delimited by single or double quote marks (' or "), or by backtick characters (`), depending on the server SQL mode. *Example*:

```
mysql> TABLE $t;  # Unquoted, produces warning
+------+
| a    |
+------+
|    1 |
|    2 |
+------+
2 rows in set, 1 warning (0.00 sec)

mysql> SHOW WARNINGS\G
*************************** 1. row ***************************
  Level: Warning
   Code: 1681
Message: '$ as the first character of an unquoted identifier' is deprecated and
will be removed in a future release.
1 row in set (0.00 sec)

mysql> TABLE `$t`;  # Quoted, no warning
+------+
| a    |
+------+
|    1 |
|    2 |
+------+
2 rows in set (0.00 sec)
```

User variables are not affected by this change. For example, the statement `SELECT 1 INTO @$x` does not produce a warning.

See Schema Object Names, for more information. (Bug #34785775, WL #15422)

References: See also: Bug #34684193.

- The `CLIENT_NO_SCHEMA` flag is deprecated. Client programs that specify `CLIENT_NO_SCHEMA` as the `client_flag` argument to `mysql_real_connect()` can now omit the flag and the `db` argument to have the connection set the database value to the current (or default) database. The `libmysqlclient` library now prints a warning on standard error when `mysql_real_connect()` is called with `CLIENT_NO_SCHEMA`. In addition, the server adds a deprecation warning for every non-prepared query executed, if the connection has `CLIENT_NO_SCHEMA`. (WL #13128)

- Previously, legacy compression-control parameters were deprecated and replaced with new configuration parameters for greater control over the use of compression in connections to the server. The new and deprecated parameters are:

  - The `--compression-algorithms` client option enables deprecation of the legacy `--compress` client option and the `Compression` status variable.

  - The `MYSQL_OPT_COMPRESSION_ALGORITHMS` C API option enables deprecation of the legacy `MYSQL_OPT_COMPRESS` C API option.

  - The `MASTER_COMPRESSION_ALGORITHMS` option for the `CHANGE MASTER TO` statement enables deprecation of the legacy `slave_compressed_protocol` system variable.

  The deprecated parameters will be removed in a future MySQL version.

  Now, the following client programs print a deprecation warning to standard error when a client user invokes one of the programs with `--compress` (or `-C`, if applicable): `mysqlpump`, `mysqlcheck`,

`mysql`, `mysqladmin`, `mysqlbinlog`, `mysqldump`, `mysqlimport`, `mysqlshow`, `mysqlslap`, `mysql_upgrade`, and mysqltest.

The `mysqlbackup --compress` option has different capabilities and is not deprecated. (WL #13292)

## Generated Invisible Primary Keys (GIPKs)

- **Replication:** It is now possible to cause a replica to add a generated invisible primary key to any `InnoDB` table that otherwise, as replicated, has no primary key. This is implemented by adding `GENERATE` as a possible value for the `CHANGE REPLICATION SOURCE TO` statement's `REQUIRE_TABLE_PRIMARY_KEY_CHECK` option.

  `REQUIRE_TABLE_PRIMARY_KEY_CHECK = GENERATE` can be used on a per-channel basis. Suppose the replica is configured with two replication channels, named `ch1` and `ch2`, and you execute the statements shown here:

  ```
  STOP REPLICA;

  SET @@sql_require_primary_key = ON;

  CHANGE REPLICA SOURCE TO
    REQUIRE_TABLE_PRIMARY_KEY_CHECK = GENERATE
    FOR CHANNEL ch1;

  START REPLICA;
  ```

  The effect of these statements is that the replica now adds an invisible primary key for tables whose creation is replicated by channel `ch1`, but does not do so for any keyless table that is created in the context of `ch2`.

  A replica ignores any setting for `sql_generate_invisible_primary_key` that has been made on the source, and this variable is not replicated. (This behavior is unchanged from previous releases.)

  `REQUIRE_TABLE_PRIMARY_KEY_CHECK = GENERATE` is not compatible with MySQL Group Replication, where `ON`, `OFF`, and `STREAM` are the only supported values for this option.

  See CHANGE REPLICATION SOURCE TO Statement, and Generated Invisible Primary Keys, for further information. (WL #15419)

## Keyring Notes

- Host names of endpoints specified in `component_keyring_oci` configuration files, and obtained from the Oracle Cloud Infrastructure Console or by querying the Oracle Cloud Infrastructure API, now can retain the `https://` prefix that previously had to be removed when generating a MySQL configuration for the Oracle Cloud Infrastructure Vault keyring component. (Bug #34636297)

## Pluggable Authentication

- On Windows, the client-side Kerberos authentication plugin now supports GSSAPI through the MIT Kerberos library. It is possible to choose between SSPI and GSSAPI at runtime using a new plugin option supported by the `authentication_kerberos_client` authentication plugin on Windows. Client users invoke `mysql` or `mysqldump` with the `--plugin-authentication-kerberos-client-mode` command-line option to set the mode to GSSAPI. The default mode of the `authentication_kerberos_client` plugin is SSPI, previously the only authentication method on Windows.

  For more information, see Connection Commands for Windows Clients in GSSAPI Mode. (WL #15336)

## Spatial Data Support

- The MySQL `ST_Transform()` function now supports all Cartesian projections, with the exceptions of EPSG 1042 (Krovak Modified), EPSG 1043 (Krovak Modified (North Orientated)), EPSG 9816 (Tunisia Mining Grid), and EPSG 9826 (Lambert Conic Conformal (West Orientated)). (Bug #27272733, Bug #34495023, WL #15164)

## SQL Syntax Notes

- **Important Change:** Previously, MySQL supported the use of “full” as the name of a table, column, view, stored procedure, or stored function, as well as for the alias of a table, view, or column. Beginning with this release, using “full” (regardless of letter case) in this fashion as an unquoted identifier is now deprecated, and raises a warning. This is to align more closely with the SQL standard, in which `FULL` is reserved as a keyword.

  For example, the following `CREATE TABLE` and `DROP TABLE` statements now raise warnings, as shown here:

```
mysql> CREATE TABLE full (c1 INT, c2 INT);
Query OK, 0 rows affected, 1 warning (0.03 sec)

mysql> SHOW WARNINGS\G
*************************** 1. row ***************************
  Level: Warning
   Code: 4119
Message: Using FULL as unquoted identifier is deprecated, please use quotes or
rename the identifier.
1 row in set (0.00 sec)

mysql> DROP TABLE full;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> SHOW WARNINGS\G
*************************** 1. row ***************************
  Level: Warning
   Code: 4119
Message: Using FULL as unquoted identifier is deprecated, please use quotes or
rename the identifier.
1 row in set (0.00 sec)
```

  To execute the statements without the warnings, encase the table name in each of them with backtick characters (`), like this:

```
mysql> CREATE TABLE `full` (c1 INT, c2 INT);
Query OK, 0 rows affected (0.03 sec)

mysql> DROP TABLE `full`;
Query OK, 0 rows affected (0.02 sec)
```

  The use of `FULL` in the right hand side of a value assignment is *not* affected by this change. For example, the `SET` statement shown here remains valid:

```
mysql> SHOW VARIABLES LIKE '%metadata%';
+---------------------------------------+---------+
| Variable_name                         | Value   |
+---------------------------------------+---------+
| binlog_row_metadata                   | MINIMAL |
| innodb_stats_on_metadata              | OFF     |
| performance_schema_max_metadata_locks | -1      |
| resultset_metadata                    | FULL    |
+---------------------------------------+---------+
4 rows in set (0.00 sec)
```

```
mysql> SET @@global.binlog_row_metadata=FULL;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW VARIABLES LIKE '%metadata%';
+---------------------------------------+-------+
| Variable_name                         | Value |
+---------------------------------------+-------+
| binlog_row_metadata                   | FULL  |
| innodb_stats_on_metadata              | OFF   |
| performance_schema_max_metadata_locks | -1    |
| resultset_metadata                    | FULL  |
+---------------------------------------+-------+
4 rows in set (0.00 sec)
```

For more information, see Keywords and Reserved Words. (WL #15241)

- It is now possible to set the default format for the output of any `EXPLAIN` statement which obtains a query execution plan, and which has no `FORMAT` option, using the `explain_format` system variable added in this release. Like the `FORMAT` option, this variable can take any of the values `TRADITIONAL`, `JSON`, or `TREE`. `DEFAULT` is also supported as a synonym for `TRADITIONAL`. (`DEFAULT` is *not* supported with the `FORMAT` option for `EXPLAIN`.) Suppose the value of `explain_format` is `TREE`; in this case, the output from any such `EXPLAIN` statement uses the tree-based format, as though `FORMAT=TREE` had been specified as part of the `EXPLAIN` statement.

  Any value set for `explain_format` is overridden by a `FORMAT` option. This means that, if *`explain_format`* is set to *`TREE`*, supplying `FORMAT=JSON` when invoking `EXPLAIN` causes the value of `explain_format` to be ignored, and the result is displayed using the `JSON` format.

  `explain_format` also affects the behavior of `EXPLAIN ANALYZE`; since this statement supports only the `TREE` format, if the value of explain_format is not `TREE`, this means that any `EXPLAIN ANALYZE` statement that does not specify the `TREE` format explicitly raises the error `This version of MySQL doesn't yet support 'EXPLAIN ANALYZE with format format'`.

  The new system variable has both global and session scope, can be persisted, and can be set from the command line (as `--explain-format`) or in a `my.cnf` option file.

  See the description of `explain_format`. See also Obtaining Execution Plan Information, and Obtaining Information with EXPLAIN ANALYZE, for further information and examples. (WL #15040)

  References: See also: Bug #33629360.

## Thread Pool Notes

- Whenever a connection was terminated due to inactivity, the thread pool plugin printed only a generic message about connections timing out; this often made analysis of such timeouts more difficult than necessary. A new `INFO_LEVEL` message makes it clear that a connection has been terminated due to inactivity in the thread pool, as well as which timeout value was used to make this determination. (Bug #34767607)

- Two columns added to the Performance Schema `tp_thread_state` table in this release make it possible to identify a thread's type, and to map threads in this table to those in the Performance Schema `threads` table. The type of thread is now shown in the `tp_thread_state` table's `TP_THREAD_TYPE` column, and the thread's unique ID in the `THREAD_ID` column. For more information, see The tp_thread_state Table. (Bug #34020058)

## Functionality Added or Changed

- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.1.1s. Issues fixed in OpenSSL version 1.1.1s are described at https://www.openssl.org/news/cl111.txt. (Bug #34828308)

- Binary packages that include curl rather than linking to the system curl library have been upgraded to use curl 7.86.0. (Bug #34828111)

- The internal resource-group enhancement added in MySQL 8.0.31 is refactored, but it continues to support the `Resource_group_supported` status variable. (Bug #34702833, Bug #34699751)

  References: Reverted patches: Bug #34264356.

## Bugs Fixed

- **Important Change:** The implementation of the `max_join_size` system variable, although documented as a maximum number of rows or disk seeks, did not check the number of rows or disk seeks directly, but instead treated `max_join_size` as the maximum estimated cost to permit. While cost and row count are correlated, they are not the same, and this could lead to unexpected results when some large queries were allowed to proceed.

  In this release, we change how `max_join_size` is used, so that it now actually limits the maximum number of row accesses in base tables. If the estimate indicates that a greater number of rows must be read from the base tables, an error is raised. This makes the actual behavior better reflect what is documented. (Bug #83885, Bug #25118903)

- **InnoDB:** Undetectable problems after upgrade from 8.0.28, crash and corruption.

  Any new row inserted after upgrade will have all columns added with `ALGORITHM=INSTANT` materialized and have version=0.

  In the new implementation, a column added with `ALGORITHM=INSTANT` will fail if the maximum possible size of a row exceeds the row size limit. So new rows with materialized `ALGORITHM=INSTANT` columns will always be within row size limit. (Bug #34558510)

- **InnoDB:** No more garbled UTF characters in SHOW ENGINE INNODB STATUS (Bug #34486877, Bug #108111)

- **InnoDB:** Alter handler adjusted so that adding more than 1024 columns with `ALGORITHM=INSTANT` no longer causes a crash. (Bug #34378513, Bug #107854)

- **InnoDB:** After a column added with `ALGORITHM=INSTANT`, an online rebuild DDL no longer crashes. (Bug #33788578, Bug #106279)

- **InnoDB:** Several adaptive hash index (AHI) code optimizations and improvements were implemented, addressing various issues including potential race conditions. (Bug #33601434)

- **Replication:** When `SOURCE_HEARTBEAT_PERIOD` was set to a very small value (such as 1 microsecond) on the server using `CHANGE REPLICATION SOURCE TO`, and the `mysqlbinlog` client program was started with `--read-from-remote-server` and `--stop-never=1`, it was possible for the binary log dump thread to send an `EOF` packet to the client before all events had been sent. (Bug #34860923)

- **Replication:** Removed an assert from `sql/rpl_group_replication.cc` which triggered a false error in testing. (Bug #34619134)

- **Replication:** After MySQL was started with `--server-id=0`, trying to change the server ID by using `SET PERSIST server_id=N` (where N is an integer greater than zero) and restarting the server had the following results:

  - `SELECT @@server_id` returned *`N`*.
  - Any replication SQL statement such as `START REPLICA` was rejected with `ER_SLAVE_CONFIGURATION`.

  To fix this problem, we now ensure that such checks use the value of the server variable rather than the value passed to the startup option. (Bug #34412816)

- **Replication:** When replicating compressed binary log events generated by the `NDB` binary log injector, relay log positions were not updated in the multithreaded applier, thus causing replication to hang. (Bug #33889030)

  References: See also: Bug #33784241.

- **Replication:** Issuing `STOP REPLICA SQL_THREAD` while the SQL thread was handling a transaction caused replication to stop immediately, instead of waiting 60 seconds for the event group to complete before shutting down the SQL thread as expected.

  The root cause of this issue was due to the internal variable storing the last event start time not being reset after the SQL thread was restarted.

  We fix this by resetting the variable holding the last event start time whenever the SQL thread is started. (Bug #33646899)

- **Replication:** While their wording might imply otherwise, the log messages `Setting super_read_only=ON` (`ER_GRP_RPL_SUPER_READ_ON`) and `Setting super_read_only=OFF` (`ER_GRP_RPL_SUPER_READ_OFF`) were written only after the operations were attempted, and not beforehand, or while the operations were ongoing. This sometimes led to confusion when setting the variable was rejected, and this was logged prior to the set attempt itself being logged. To keep this from happening, these messages are now logged just prior to attempting the operation. (Bug #108843, Bug #34728079)

- **Replication:** The `relay_log_space_limit` system variable is a 64-bit value, but its valid maximum was specified internally as that of a 32-bit value. (Bug #106323, Bug #33799840)

- **Replication:** Eliminated an unnecessary update of the `gtid_executed` table which was performed when rotating the binary logs. (Bug #106116, Bug #33759477)

- **Group Replication:** The `WRITE_CONSENSUS_SINGLE_LEADER_CAPABLE` column of the MySQL Performance Schema's `replication_group_communication_information` table reflects the runtime value of a Paxos Single Leader setup in a group, letting users know what the value of `group_replication_paxos_single_leader` must be on joining members.

  A group that was bootstrapped with single-leader enabled but with its protocol version downgraded to one that did not support it reported `WRITE_CONSENSUS_SINGLE_LEADER_CAPABLE` equal to `0`, as expected, but attempting to join an instance to the group using `group_replication_paxos_single_leader = 0` was not possible.

  To solve this problem, we change the behaviour and make the value of `group_replication_paxos_single_leader` consistent with the communication version that the

group is running. Since this variable was introduced in MySQL 8.0.27, it is not known or used in any previous version, and so we now enforce the following rules:

- When a node tries to join a group that is running MySQL 8.0.26 or earlier and we are version 8.0.27 or later, we reject the attempt with an error stating that `group_replication_paxos_single_leader` must be `OFF` before joining the group
- When we try to use `group_replication_set_communication_protocol()` to set a version less than 8.0.27 and we are of version 8.0.27 or later, we reject the function call if `group_replication_paxos_single_leader` is not `OFF`.

In addition, we also change the value checked to determine whether changing the group leader is allowed after running `group_replication_set_communication_protocol()`. Previously, this was the runtime value of `group_replication_paxos_single_leader`, which takes effect only after a group reboot. Instead, when we run `group_replication_set_communication_protocol()`, we now use the value shown by the `replication_group_communication_information` table's `WRITE_CONSENSUS_SINGLE_LEADER_CAPABLE` column, described previously. (Bug #34555045, Bug #34828311)

- **Group Replication:** In a 3 node cluster, all nodes were killed due to running out of memory. Subsequently, after all nodes were restarted successfully, attempting to bring the cluster back online caused the node that had been the primary to hang.

  For more information, see Rebooting a Cluster from a Major Outage. (Bug #108339, Bug #34564856)

- **Group Replication:** When a group was run with `group_replication_consistency = AFTER` and a secondary failed due to external conditions such as an unstable network, the secondary could sometimes encounter the error `Transaction 'GTID' does not exist on Group Replication consistency manager while receiving remote transaction prepare`.

  The root cause of this issue was that the primary might log out of order the `View_change_log_event` with which the secondary rejoined; when the secondary used the primary as the group donor, this could cause the secondary to catch up with the group improperly and, eventually, generate incorrect GTIDs for the group transactions. The group replication primary ensures that the `View_change_log_event` is logged after all preceding transactions, but there was a window during which transactions ordered after the `View_change_log_event` on the group global order could be logged before the event.

  To solve this issue, we now make sure that transactions ordered before a view are always logged before the `View_change_log_event`, and that transactions ordered after a view are always logged after this event. This is now done by the binary log ticket manager, which guarantees the order in which transactions in the binary log group commit are committed. (Bug #104980, Bug #33405699)

  References: See also: Bug #34746357.

- **Microsoft Windows:** When compiling MySQL on Windows platforms, the CMake `-DWITH_WIN_JEMALLOC` option was not always handled correctly. (Bug #108341, Bug #34698376)

- **JSON:** While saving the result of `JSON_ARRAYAGG()` or `JSON_OBJECTAGG()` in a column, the data type information was lost due to the result being an item of type `SUM_FUNC_ITEM`. To fix this, we remove the type check and this way retain the original type information. (Bug #108326, Bug #34548259)

- Some remote connections to the server were not handled correctly. This issue arose as the result of a previous fix for an issue with `require_secure_transport`. (Bug #34857411)

  References: This issue is a regression of: Bug #34094706.

- Some query plans were not stable due to nondeterministic sorting of `Key_use_array` in `sql_optimizer.cc`; now we sort it with `std::stable_sort()` instead of `std::sort()`. (Bug #34823952)

  References: This issue is a regression of: Bug #25965593.

- Binary packages that include OpenLDAP rather than linking to the system OpenLDAP library were upgraded to use version 2.5.13. (Bug #34815046)

- In some cases, an unexpected packet sent by the server to a MySQL client program during authentication could result in an infinite loop. (Bug #34805922)

- GIS data was not always handled correctly in windowing functions. (Bug #34778646)

- A thread remained bound to the CPU of a dropped resource group even after it was assigned to the user default resource group (`USR_default`). `USR_default` has 0 CPU priority and no CPU affinity, so with this fix, the thread now is able to run any CPU with `USR_default`. (Bug #34748973)

- With JSON logging enabled, calling the `audit_log_rotate()` function did not rotate the file as expected. A rotated file name consists of the timestamp from the last event logged into the file. When the file is empty, last timestamp is the same as the timestamp in the already created file. To fix this issue, the function now uses the current time to name the file if the file is empty. (Bug #34733508)

- Some queries having multiple lateral derived tables did not produce the expected result. (Bug #34716246)

- The bundled zlib library has been upgraded to zlib 1.2.13; zlib 1.2.13 is now the minimum zlib version supported. (Bug #34711762, Bug #34711758)

- Certain `INTERSECT` queries were not handled correctly. (Bug #34642435)

- Using the `MAX_EXECUTION_TIME` optimizer hint with a value greater than the stated maximum kept an upgrade to MySQL 8.0.30 from completing; this caused the server to report a warning which was interpreted by the upgrade process as an unrecoverable error. (Bug #34607401)

- In certain cases, evaluation of window functions was not performed correctly. (Bug #34572136)

  References: This issue is a regression of: Bug #32644631, Bug #32802301.

- Some CTEs were not processed correctly. (Bug #34572040, Bug #34634469)

  References: This issue is a regression of: Bug #33856374.

- Values returned from functions or operators that convert a value to `FLOAT` (`CAST(... AS FLOAT)`, `CONVERT(..., FLOAT)`, `JSON_VALUE(... RETURNING FLOAT))` may have extra precision in their internal representation, since they are stored in double precision internally. This sometimes caused unexpected results when checking such values for equality, such as `SELECT DISTINCT` returning duplicates and comparison operators incorrectly reporting two equal values as unequal.

  We fix this problem by stripping off the extra double precision from the values before returning them, and by making any conversion from float to string in these conversion operators use float format instead of double format. (Bug #34554755)

- Removed an assertion in `query_expression::assert_not_fully_clean()`. (Bug #34526104)

- Upgrading from MySQL 5.7 to MySQL 8.0 with a very large number of tables in a single database caused the server to consume excessive memory. It was found that, during the process of checking whether tables could be upgraded, we fetched all the data dictionary `Table` objects upfront, processing

each and fetching its name, then performed `CHECK TABLE ... FOR UPGRADE` on the list. Fetching all objects beforehand was not necessary in this case, and contributed greatly to memory consumption.

To correct this problem, we now fetch one `Table` object at a time in such cases, performing any required checks, fetching its name, and releasing the object, before proceeding with the next one. (Bug #34526001)

- When creating natural join columns, a hidden column added as part of a materialized derived table is used in constructing the join condition, which is later used to check whether the join eligible for pushdown to derived tables. The current issue arose when this column was not retrieved from the derived table due to it being hidden; this occurred even when the condition pushdown optimization was not enabled. We solve this problem by rejecting all hidden columns added internally, and not merely for hidden columns added for functional indexes. (Bug #34523627)

- Types were not derived consistently from user variables. This could be seen, for example, by executing the following statements repeatedly:

```
CREATE TABLE t AS SELECT @max_error_count UNION SELECT 'a';

SHOW CREATE TABLE t;
```

  In this particular case, the output of the `SHOW CREATE TABLE` statement showed `` `@max_error_count` text `` the first time, and `` `@max_error_count` mediumblob `` in successive iterations. (The second is correct.) (Bug #34523475)

- Following work done in MySQL 8.0.23 to improve host resolution for user accounts, the time needed for `CREATE USER` to complete increased significantly, particularly when running many such statements in close succession.

  Prior to upgrading to this release, you can work around this issue when issuing many such statements in succession by preceding them with a single `CREATE USER 'fakeuser' ACCOUNT LOCK` (you can use any user name that does not conflict with existing ones for this). When finished, you can (and should) clean up by issuing the following statements:

```
DROP USER 'fakeuser';
FLUSH PRIVILEGES;
```

  For more information, see Access Control, Stage 1: Connection Verification. (Bug #34449016)

- The `data_masking` server-side plugin could emit a runtime error and halt unexpectedly. (Bug #34445632)

- Stack overruns could be erroneously reported when using the thread pool under heavy load. The race condition causing this in the thread pool has been avoided with an additional thread group ID check. (Bug #34414959)

- When merging a derived table, a nested join condition is added to the derived table and the underlying tables are added to this join nest. In addition, the join condition is associated with the derived table.

  Evaluation for range access is skipped if the table is an inner table of an outer join or if the table is an inner table and the join is not a semijoin. For a derived table, the underlying base table was treated as being of the latter kind, range analysis was skipped, and the range access method was thus unavailable.

  To fix this problem, we now evaluate for range access when the embedding table is a derived table, and ensure that the join condition associated with the derived table is used for the range optimization. (Bug #34347116)

- A `LOAD DATA INFILE` statement issued with a subquery could cause the server to return an incorrect warning (`Subquery returns more than 1 row`). (Bug #34336033)

- Improved handling of resource allocation for internal temporary tables. (Bug #34174001)

- A specific column added after a drop using the `INSTANT` algorithm could cause a data error and a server exit. (Bug #34122122)

- A query such as `SELECT 1 AS one FROM t WHERE 1=(SELECT 1 UNION SELECT 2)` is transformed to this:

  ```
  SELECT 1 AS one
  FROM t
  JOIN ( SELECT 1 AS col1 UNION SELECT 2) derived
  WHERE 1 = derived.col1;
  ```

  The optimizer in this case pushed down `1 = derived.col1` into the union, removing the contribution from `SELECT 2`, which led to an erroneous result. We now no longer push the condition down in such cases. (Bug #33910786)

- Some parenthesized query expressions with either or both of `ORDER BY` and `LIMIT` were not always handled correctly. (Bug #33725530)

- Reimplemented retention of item trees having multiple references, by using a reference count in every `Item` object. Also removed old code no longer needed due to this change. (Bug #33725415)

- An assert occurred in `InnoDB` during upgrade of the data dictionary when the definition of the `innodb_ddl_log` table changed, even when such changes were effectively null operations, such as updating `utf8` and `utf8_bin` in table and column definitions to `utf8mb3` and `utf8mb3_bin`, respectively. (Bug #33688509)

  References: This issue is a regression of: Bug #33787300.

- Data and GTIDs backed up by `mysqldump` were inconsistent when the options `--single-transaction` and `--set-gtid-purged=ON` were both used. This was because, between the start of the transaction by `mysqldump` and the fetching of `GTID_EXECUTED`, GTIDs on the server could have increased already. With this fix, a `FLUSH TABLES WITH READ LOCK` is performed before fetching `GTID_EXECUTED`, to ensure that its value is consistent with the snapshot taken by `mysqldump`.

  Our thanks to Marcelo Altmann for the contribution.

  *Limitation*: This fix adds a requirement for the `RELOAD` privilege when using `--single-transaction` to execute `FLUSH TABLES WITH READ LOCK`; the MySQL team is investigating a solution. (Bug #33630199, Bug #105761)

- `SET PERSIST` accepted dot-separated names of variables registered by components as well as `MyISAM` multiple key cache variables, but `RESET PERSIST` rejected the same names with a syntax error. To fix this discrepancy, we add support in this to `RESET PERSIST` for variable names containing a dot character (`.`). (Bug #33417357)

- Some grouped queries were not always handled correctly. (Bug #33294005, Bug #33349994)

- When multifactor authentication used the `auth_socket` authentication plugin for the first factor, the server executed the wrong code during the second-factor authentication workflow and returned an error message. The second factor could be any authentication plugin. (Bug #33192223)

- Use of a wild card as a column identifier in an `INSERT` statement was allowed by the parser even though the syntax is not supported, which led to an assert in debug builds and a silent rejection of the statement

in release builds. This construction has been removed as a possibility from the grammar, and is now handled strictly as a syntax error. (Bug #33142665)

References: This issue is a regression of: Bug #30528450.

- In prepared statements, some types of subqueries could cause a server exit. (Bug #33100586)
- Some floating-point literals were not always handled correctly. (Bug #32824429)
- A `DELETE` statement with a table alias could result in an intermittent server exit. (Bug #32762229)
- Moved INFO_SRC and INFO_BIN from the mysql-common package to the mysql-community-server-core package, the same package as mysqld and more consistent with RPM packaging. (Bug #32752147)
- Some queries making use of `MATCH()` within a `HAVING` clause were not handled correctly. (Bug #32616816, Bug #32934558, Bug #34782389)
- A `CREATE VIEW` statement that contained a subquery sometimes led to an assertion in debug builds. (Bug #108783, Bug #34703610)
- The deduction of data types for dynamic parameters passed as parameters to a user-defined SQL function was correct only for a single parameter; with more than one parameter, no such deduction was performed for the second and following parameters, with the result that their types were always reported erroneously to clients as `MYSQL_TYPE_INVALID`. (Bug #108545, Bug #34629157)
- The internal function `clone_os_copy_file_to_buf()` did not advance the buffer position in the event of a partial read.

  Our thanks to Laurynas Biveinis for the contribution. (Bug #108317, Bug #34543194)
- Views that access system views could encounter an access-denied error during normal use if the pushdown condition included expressions that used native functions from the system view. (Bug #108202, Bug #34515868)
- When using window functions, the current row could reevaluate itself based on the wrong record in some cases. (Bug #108008, Bug #34431996)
- A condition pushdown into a `UNION` of queries having `LIKE` clauses did not preserve the correct character set, leading to an (erroneous) empty result.

  We solve this problem in two parts:

  1. By refactoring resolution of `LIKE` expressions, in which character set determination and propagation were previously performed in two separate blocks of the code that were not always consistent with one another.
  2. By adding, in the internal `parse_expression()` function, a character set prefix to any literal character string that is cloned.

  (Bug #107787, Bug #34359297, Bug #34589153)
- Some multiply nested queries were not performed correctly. (Bug #107660, Bug #34321147, Bug #34377854)
- The `audit_log` server-side plugin always logged an entire multiple query, rather than logging only the specific part of the query that was executed. Changing when the query length is set resolves the issue. (Bug #107390, Bug #34207811)

- Following an upgrade to MySQL 8.0.27 a specific query started consuming comparatively high amounts of memory whenever it was run within a stored procedure. (Bug #107327, Bug #34190122)
- When a `mysqld` startup option was used with the `maximum-` prefix, the upper bound for the corresponding system variable was set but its current value was not checked against or adjusted according to the new limit and thus could in some cases be greater than the stated maximum. We fix this by adjusting the current value if it is larger than the new user defined maximum value. (Bug #99029, Bug #31072098)
- The `mysql_stmt_close()` C API function could stop responding after a prepared statement was canceled using `KILL QUERY`. (Bug #84470, Bug #25584097)

# Changes in MySQL 8.0.31 (2022-10-11)


- Audit Log Notes
- Compilation Notes
- Component Notes
- Deprecation and Removal Notes
- Keyring Notes
- Optimizer Notes
- Performance Schema Notes
- SQL Syntax Notes
- Functionality Added or Changed
- Bugs Fixed


## Audit Log Notes

- The `audit_log_rotate()` function simplifies audit log file rotation. Previously, audit log file rotation required renaming the file manually and setting `audit_log_flush = ON` to close the file and open a new log file with the original name. The `audit_log_rotate()` function renames the current file and creates a new one. Manually renaming the audit log file is no longer necessary.

  The `audit_log_flush` variable is deprecated as of MySQL 8.0.31; expect support for it to be removed in a future version of MySQL. (WL #15249)

## Compilation Notes

- **Microsoft Windows:** Fixed all C++ deprecation warnings by not using deprecated C++ STL features. (Bug #33985941)
- Renamed internal Performance Schema functions from _utf8* to _utf8mb4* as they've used utf8mb4 since v8.0.11. (Bug #34351301)

  References: See also: Bug #27407745.
- Fixed linker flags for clang/ubsan to work around a LLVM/Clang related integer issue as described in LLVM bug #16404. (Bug #34311325)

- Updated helper scripts to use 'utf8mb3' for charsets/collations instead of 'utf8' for future compatibility.

  Renamed functions and data structures used to handle utf8mb3 character sets and collations to help clarify their utf8mb3 usage. (Bug #34263480, Bug #34025412)

- Cleaned up CMake code, and simplified INFO_BIN and INFO_SRC. (Bug #34139084)

## Component Notes

- New component services now enable server components and plugins to query within the local server. The new MySQL command services are similar to C API functions in `libmysql`, except they do not expose protocol internals to the client. For information about these services, see the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html. Search for:

  - `s_mysql_mysql_command_factory`
  - `s_mysql_mysql_command_options`
  - `s_mysql_mysql_command_query`
  - `s_mysql_mysql_command_query_result`
  - `s_mysql_mysql_command_field_info`
  - `s_mysql_mysql_command_error_info`

  (WL #14293)

## Deprecation and Removal Notes

- The `keyring_oci` plugin is deprecated and subject to removal in a future release of MySQL. Instead, consider using the `component_keyring_oci` component for storing keyring data (see Using the Oracle Cloud Infrastructure Vault Keyring Component). (WL #14943)

## Keyring Notes

- MySQL Keyring previously implemented Oracle Cloud Infrastructure Vault Keyring keystore capabilities using a server plugin, but now is transitioning to use the MySQL component infrastructure. The new keyring component has similarities to the existing `keyring_oci` plugin, but is configured differently and can access keys that were created using the `keyring_oci` plugin when a similar set of configuration options are used to initialize the new component. For configuration details, see Using the Oracle Cloud Infrastructure Vault Keyring Component.

  `component_keyring_oci` is not loaded using the `--early-plugin-load` server option during startup or configured during startup or at runtime using system variables:

  - During startup, the server determines which keyring component to load using a manifest file, and the loaded component consults its own configuration file when it initializes. See Keyring Component Installation.
  - At runtime, the `ALTER INSTANCE RELOAD KEYRING` statement enables reconfiguring an installed keyring component after changes to its configuration file. See ALTER INSTANCE Statement.

  If a keyring component is installed, the Performance Schema `keyring_component_status` table provides status information about it. See The keyring_component_status Table. (WL #14943)

## Optimizer Notes

- **Important Change:** The MySQL Optimizer's internal management of set operations has been improved, with the following effects:

  - Bodies of parenthesized query expressions can now be nested to in combination with `UNION`. For example, the query shown here, previously rejected with error `ER_NOT_SUPPORTED_YET`, is now allowed:

    ```
    (
      (SELECT a, b, c FROM t ORDER BY a LIMIT 3) ORDER BY b LIMIT 2
    ) ORDER BY c LIMIT 1;
    ```

  - When collapsing parenthesized expression bodies, MySQL now follows the semantics specified in the SQL standard, so that a higher outer limit cannot override an inner lower one. This means that the expression `(SELECT ... LIMIT 3) LIMIT 5` can return at most three rows.

  - `UNION DISTINCT` and `UNION ALL` can now be nested in arbitrary combinations.

  The maximum nesting level supported in all the cases just listed is 63; this follows any simplifications or merging performed by the parser. See Parenthesized Query Expressions, for more information. (Bug #103954, WL #11350)

- An outer reference in a derived table pointing to a table in a `JOIN` operation that was not available for the subquery to be reported (incorrectly) as valid, whereas such a reference without a derived table was correctly reported as invalid.

  In a related issue, an outer reference to a derived table placed inside the `VALUES` clause of an `INSERT` statement was not recognized as valid, whereas a similar reference without a derived table was so recognized. Now such a reference to a derived table is accepted within `VALUES`.

  In both cases, we fix the problem by making sure that the correct name resolution context is used for the outer reference. (Bug #32678303, Bug #34131822)

## Performance Schema Notes

- The Performance and sys schemas now expose metrics for the global and session memory limits introduced in MySQL 8.0.28.

  The following columns were added to tables of the Performance Schema:

  - `SETUP_INSTRUMENTS`:

    - `FLAGS`

  - `THREADS`:

    - `CONTROLLED_MEMORY`
    - `MAX_CONTROLLED_MEMORY`
    - `TOTAL_MEMORY`
    - `MAX_TOTAL_MEMORY`

  - `EVENTS_STATEMENTS_CURRENT`, `EVENTS_STATEMENTS_HISTORY`, and `EVENTS_STATEMENTS_HISTORY_LONG`:

    - MAX_CONTROLLED_MEMORY
    - MAX_TOTAL_MEMORY
  - Statement Summary Tables:
    - MAX_CONTROLLED_MEMORY
    - MAX_TOTAL_MEMORY
  - Performance Schema Connection Tables:
    - MAX_SESSION_CONTROLLED_MEMORY
    - MAX_SESSION_TOTAL_MEMORY
  - PREPARED_STATEMENTS_INSTANCES:
    - MAX_CONTROLLED_MEMORY
    - MAX_TOTAL_MEMORY

  The following columns were added to the sys schema STATEMENT_ANALYSIS and X$STATEMENT_ANALYSIS views:

  - MAX_CONTROLLED_MEMORY
  - MAX_TOTAL_MEMORY

  The PROPERTIES column of the SETUP_INSTRUMENTS table was updated with the controlled_by_default flag.

  You can also add, or remove, non-global memory instruments to the set of controlled-memory instruments by setting the value of the FLAGS column of SETUP_INSTRUMENTS.For example:

```
SQL> UPDATE PERFORMANCE_SCHEMA.SETUP_INTRUMENTS SET FLAGS="controlled"
     WHERE NAME='memory/sql/NET::buff';
```

  (WL #14432)

## SQL Syntax Notes

- It is now possible to set a column histogram to a user-specified JSON value. This can be useful when sampling leaves out important values. This also means that a secondary (replica) MySQL server can assume the work of sampling the data and building the histogram, which can then be used on the primary (source) without impacting its performance.

  This capability is provided through an extension to ANALYZE TABLE ... UPDATE HISTOGRAM. The following statement sets the histogram for column *col_name* of table *tbl_name* to the histogram's JSON representation *json_data*:

```
ANALYZE TABLE tbl_name
  UPDATE HISTOGRAM ON col_name
  USING DATA 'json_data'
```

You can confirm afterwards this by checking the value of the `HISTOGRAM` column of the Information Schema `COLUMN_STATISTICS` table.

See Histogram Statistics Analysis, for more information and examples.

Our thanks to Kaiwang Chen for this contribution to MySQL. (Bug #104040, Bug #33012389, WL #15123)

- In this release MySQL adds support for the SQL standard `INTERSECT` and `EXCEPT` table operators.

  *`query_a`* `INTERSECT` *`query_b`* includes only rows appearing in both result sets.

  *`query_a`* `EXCEPT` *`query_b`* returns any rows from the result set of *`query_a`* which are *not* in the result of *`query_b`*.

  `INTERSECT` and `EXCEPT` both support `DISTINCT` and `ALL`, with `DISTINCT` the default in both cases. (This is the same as for `UNION`).

  `INTERSECT` groups before `EXCEPT` or `UNION`, so `TABLE r EXCEPT TABLE s INTERSECT TABLE t` is evaluated as `TABLE r EXCEPT (TABLE s INTERSECT TABLE t)`.

  See INTERSECT Clause, and EXCEPT Clause, for additional information and examples. (Bug #1309, Bug #31336, Bug #11747209, Bug #11744757, WL #349)

## Functionality Added or Changed

- **Important Change; Replication:** The `Rewriter` plugin rewrites SQL queries according to a table of rewriter rules. Previously, all queries were subject to rewrites, including those from any user regardless of privileges, as well as internal system queries like those performed by the replication applier thread, by bootstrapping the MySQL Server, and by storage engines. This MySQL release provides a mechanism for allowing the plugin to skip rewrites for certain queries, such as those executed by system threads, and those from designated users.

  We do this by implementing a new privilege `SKIP_QUERY_REWRITE`, which indicates a user whose queries are not rewritten. A new system variable `rewriter_enabled_for_threads_without_privilege_checks` controls whether rewrites are skipped for threads executing with privilege checks disabled. Such threads include upgrade threads, initialisation threads, and replication threads which have `PRIVILEGE_CHECKS_USER` as `NULL`. To preserve backward compatibility, the default setting for this variable is variable is `ON`, although the recommended setting is `OFF`.

  Both `SKIP_QUERY_REWRITE` and `rewriter_enabled_for_threads_without_privilege_checks` are available only if the `Rewriter` plugin is installed.

  For more information, see The Rewriter Query Rewrite Plugin. (WL #14527)

- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.1.1q. Issues fixed in OpenSSL version 1.1.1q are described at https://www.openssl.org/news/cl111.txt and https://www.openssl.org/news/vulnerabilities.html. (Bug #34414695)

- **InnoDB:** Two new status variables are provided for monitoring online buffer pool resizing operations. The `Innodb_buffer_pool_resize_status_code` status variable reports a status code indicating the stage of an online buffer pool resizing operation. The

`Innodb_buffer_pool_resize_status_progress` status variable reports a percentage value indicating the progress of each stage.

For more information, see Configuring InnoDB Buffer Pool Size. (WL #15175)

- **Replication:** When replication filtering is in use, a replica no longer raises replication errors related to privilege checks or `require_row_format` validation for events which are filtered out. Previously, all privileges were checked in the applier, with some being checked before applying filters and others not until after; with this release, privilege checks are now deferred until after all replication filters have been applied. In addition, prior to this release, checks for `require_row_format` equal to `1` took place on both the receiver and the applier; now the applier alone performs this check, before any filters are evaluated.

  This makes it possible to filter out any transactions that fail validation, using existing replication filters, and for the replica to accept only that part of the database to which a given user has been granted access, as long as updates to this part of the database are replicated only in row-based format. Another instance in which this might be desirable is when migrating to MySQL Database Service from on-premise or cloud services which use administration tables to which the inbound replication user does not have access.

  Privilege checks protect against updating critical tables on the replica. Checking `require_row_format` protects against replicating unsafe operations; this is used in conjunction with privilege checks, because the guaranteed absence of unsafe operations removes the need for some session context initialization and cleanup actions, which in turn removes the need to grant privileges for those operations.

  Other behavior changes associated with this work are noted in the following list:

  - The receiver thread no longer errors out when an event that violates require row format validation is received; the applier thread no longer errors out when an event that violates require row format validation is handled and subsequently filtered out due to replication filters.
  - The applier thread no longer errors out for `Query` log events that are filtered out due to lack of privileges for setting any of the system variables listed here:
    - `pseudo_thread_id`
    - `sql_require_primary_key`
    - `default_table_encryption`

    This is because setting the variable and checking whether the applier is authorized to set it are now performed only after filtering takes place. If an event is filtered out, the applier does not try to set the variable, and thus never fails a check whether it has the necessary privileges to do so.
  - The applier thread no longer errors out on privilege checks when an `Execute_load_query` event is handled and ends up being filtered out due to replication filters. This is also the case when an `Append_block` or `Begin_load_query` event is applied, and a subsequent `Execute_load_query` event is handled and then filtered out.
  - The applier thread no longer errors out on privilege checks or `require_row_format` checks for `Delete_file` events. These events are not filterable, because they do not contain the table name or database name, so we assume they are filtered out. We assume this to be safe because Delete_file events do not alter database state.
  - The applier thread no longer creates or deletes files in the filesystem during the handling of `Append_block` or `Delete_file` events, when there is a user account configured by

`PRIVILEGE_CHECKS_USER` and it does not have the `FILE` privilege, nor if `require_row_format` is enabled.

- For a replication event altered by `--replicate-rewrite-db`, any privilege checks apply to the rewritten database.

See How Servers Evaluate Replication Filtering Rules, and Replica Errors During Replication, for more information. (Bug #33704306, Bug #33733677, WL #15032)

- The new `Resource_group_supported` status variable is added to indicate whether a server supports the resource group feature. (Bug #34264356)
- Binary packages that include curl rather than linking to the system curl library have been upgraded to use curl 7.85.0. (Bug #34138733, Bug #34614578)
- The new `thread_pool_transaction_delay` system variable permits specifying a delay period, in milliseconds, before worker threads start executing a new transaction. The maximum value is 300000 (5 minutes).

  A transaction delay can be used in cases where parallel transactions affect the performance of other operations due to resource contention. For example, if parallel transactions affect index creation or an online buffer pool resizing operation, you can configure a transaction delay to reduce resource contention while those operations are running. (WL #15277)
- The new `thread_pool_query_threads_per_group` system variable lets you increase the number of query threads in a thread group, from the default of a single query thread. When you are using the high concurrency thread pool algorithm (`thread_pool_algorithm=1`), consider increasing the value if you experience slower response times due to long-running transactions. It cannot exceed the value of `thread_pool_max_transactions_limit`. (WL #15330)
- Thread pool plugin features previously only available with MySQL Database Service, are now available in MySQL Enterprise Edition.

  - The `thread_pool_max_transactions_limit` variable defines the maximum number of transactions permitted by the thread pool plugin. Defining a transaction limit binds a thread to a transaction until it commits, which helps stabilize throughput during high concurrency.
  - The `thread_pool_dedicated_listeners` variable dedicates a listener thread in each thread group to listen for incoming statements from connections assigned to the group. This variable is only useful when a transaction limit is defined by `thread_pool_max_transactions_limit`. Having an additional dedicated listener thread in each thread group consumes more resources and affects thread pool performance. `thread_pool_dedicated_listeners` should therefore be used cautiously.
  - The `TP_CONNECTION_ADMIN` privilege enables a user to access the server with a privileged connection. When the limit defined by `thread_pool_max_transactions_limit` has been reached, new connections cannot access the server. A privileged connection ignores the limit defined by `thread_pool_max_transactions_limit` and permits connecting to the server to increase the transaction limit, remove the limit, or kill running transactions.

  (WL #15293)
- The new field `WRITE_CONSENSUS_SINGLE_LEADER_CAPABLE` in the Performance Schema table `replication_group_communication_information` shows whether the replication group supports the use of a single consensus leader, even if `group_replication_paxos_single_leader` is currently set to `OFF` on the queried member. The field is set to 1 if the group was started with `group_replication_paxos_single_leader` set to `ON`, and its communication protocol version is MySQL 8.0.27 or above. (WL #15203)

- Activating MySQL Server's offline mode, which is done by changing the value of the `offline_mode` system variable to `ON`, now requires the `CONNECTION_ADMIN` privilege in addition to the `SYSTEM_VARIABLES_ADMIN` privilege (or the deprecated `SUPER` privilege, which covers both these privileges). Sessions whose users have the `CONNECTION_ADMIN` privilege remain connected when offline mode is activated, and users with that privilege can make new connections to an offline server instance. The new requirement means that administrators who can activate offline mode cannot inadvertently lock themselves out by doing so. (WL #13400)

- Added the read-only `build_id` system variable. On Linux systems, a 160-bit `SHA1` signature is generated at compile time; this value, converted to a hexadecimal string, provides a unique identifier for the build. This value is written in the server log at startup.

  This variable is not supported on platforms other than Linux. (WL #15161)

- Added Ubuntu 22.10 support.

## Bugs Fixed

- **Incompatible Change:** The service `pfs_plugin_table`, deprecated since MySQL 8.0.14, is removed in this release.

  Plugins or components using this service must be updated to use `pfs_plugin_table_v1` and `pfs_plugin_column_*` instead. (Bug #34361827)

- **Important Change; Replication:** Query log events filtered by the default database whenever `--replicate-do-db` or `--replicate-ignore-db` was used included `XA START`, `XA END`, `XA COMMIT`, and `XA ROLLBACK` (but not `XA PREPARE` or `XA COMMIT ONE_PHASE`), regardless of the binary log format.

  This can lead to one of the issues listed here:

  - When `XA START` or `XA END` was filtered out, the sequence of XA statements within the first part of the transaction were rendered invalid, which made the replica stop with an error.

  - When `XA START` and `XA END` were preserved while `XA COMMIT` or `XA ROLLBACK` was filtered out, then a transaction could remain in the prepared state indefinitely on the replica.

  To prevent either of these problems from occurring, we no longer filter `XA START`, `XA END`, `XA COMMIT`, or `XA ROLLBACK` statements by the default database with `--replicate-do-db` or `--replica-ignore-db`. (Bug #106201, Bug #33764808)

- **InnoDB:** After upgrading to a release that supports row versions for columns added or dropped using `ALGORITHM-INSTANT`, a failure occurred during an instant `ADD COLUMN` operation on a table with a nullable column and an instantly added column. (Bug #34488482)

- **InnoDB:** Adding a virtual column and dropping a column in the same `ALTER TABLE` statement raised an invalid debug assertion failure. (Bug #34467592)

- **InnoDB:** The physical position of columns were not set correctly after dropping a column and renaming an existing column to the name of the dropped column. (Bug #34463089)

- **InnoDB:** A Valgrind error detected in `mtr_t::start()` was corrected. (Bug #34327575)

- **InnoDB:** A DDL operation on a corrupted partitioned table raised an assertion failure. (Bug #34293555)

- **InnoDB:** An index block latch order violation during histogram sampling blocked concurrent inserts and could cause a deadlock failure. (Bug #34282448, Bug #34174927, Bug #107299)

- **InnoDB:** An `ALTER TABLE` operation executed while a data load operation was in progress raised an assertion failure. (Bug #34267618)
- **InnoDB:** `InnoDB` memory leaks during MySQL server initialization, identified by Address Sanitizer (ASAN) builds, were fixed. (Bug #34156050)
- **InnoDB:** During recovery, a tablespace object associated with encrypted undo tablespace pages that were fetched from disk did not contain the encryption keys required to decrypt the pages, resulting in a failure. (Bug #34148143)
- **InnoDB:** In debug builds, a descending b-tree scan raised a debug assertion failure. (Bug #34144951)
- **InnoDB:** The `innodb_redo_log_consumer_advance()` function failed to handle an invalid argument. (Bug #34052884)
- **InnoDB:** A column added using `ALGORITHM=INSTANT` was visible in read views created before the DDL operation that added the column. (Bug #33937504)
- **InnoDB:** A failure occurred while upgrading a MySQL instance with a MySQL 5.6 data directory containing user-created table with a particular table ID. Assignment of that table ID resulted in assignment of conflicting data dictionary table IDs while upgrading from MySQL 5.7 to MySQL 8.0.

  Thanks to Rahul Malik for the contribution. (Bug #33919635)
- **InnoDB:** A buffer block containing intrinsic temporary table page was relocated during page traversal, causing an assertion failure. (Bug #33715694)
- **InnoDB:** Dropping a table with a discarded tablespace caused an assertion failure. (Bug #33232978)
- **InnoDB:** Page I/O reads not completed before shutdown reached the flush phase caused an assertion failure in cases where pages had change buffer merges to be completed after the I/O read. (Bug #33192496)
- **InnoDB:** An index latch order violation in `dict_table_x_lock_indexes()` caused an assertion failure. (Bug #32912868)
- **InnoDB:** A `TRUNCATE TABLE` operation failed to free an acquired mutex in specific cases. (Bug #107858, Bug #34380370)
- **InnoDB:** In debug builds, importing a tablespace without a `.cfg` file for table with an instantly added or dropped column raised an assertion failure. (Bug #107517, Bug #34307874)
- **InnoDB:** A potential memory leak in the `trx_undo_prev_version_build()` function was corrected.

  Thanks to Alex Xing for the contribution. (Bug #106952, Bug #34051207)
- **InnoDB:** A debug assertion failure occurred while redoing a space deletion during recovery. (Bug #103482, Bug #32819101)
- **InnoDB:** An `InnoDB` startup message that specified the number of object pools was removed to avoid confusion with the number of buffer pool instances. (Bug #80248, Bug #22653060)
- **Partitioning:** A partitioned table was not repartitioned as expected when an `ALTER TABLE` statement acted to change the character set of any of its columns. The root cause of this was that, when determining whether a change in character set could be done as an in-place alter operation, we did not consider whether any columns affected by the character set change were part of the table's partitioning expression. Since such a change might lead to a repartitioning of the table, it could not be done in place.

Now when a change in character set is planned, we also check whether this affects any columns that are part of a partitioning expression; if not, then we can perform the operation in place. (Bug #106884, Bug #34031693)

- **Replication:** Corrected an internal error in replication information handling. (Bug #34214416)
- **Replication:** When a single event caused multiple updates to the same row of a table on a replica, the change could not be applied if the replica had a primary key on the table that did not exist in the source. The hash-based row lookup algorithm now operates in this situation even if a primary key is present in the replica's table. (Bug #34114296)
- **Replication:** When a single statement generated multiple write events for different tables on a replica, extra auto-increment columns in the replica's tables were only cleaned up for the last event generated. They were not cleaned up at all if the last event did not involve a table with an extra auto-increment column. The cleanup process now always takes place for multiple write events. (Bug #34095290)
- **Replication:** A misleading error was returned to the client if the `group_replication_force_members` system variable was set, and the wait for the view change timed out. The error, and associated errors, have been updated so they do not suggest that the values given for the system variable were incorrect. (Bug #34091087)
- **Replication:** The FILTER_RULE column in the Performance Schema table `replication_applier_filters` showed incorrect data for filters whose rule did not exist. The table now displays the empty string in this situation. (Bug #33885484)
- **Replication:** It was possible in some rare instances of flushing or synchronization errors for the server to attempt an update of the endpoint of the binary log using invalid values for both the file and position within the file, leading to undefined behavior.

  To solve this issue, we now call the internal function `update_binlog_end_pos(`*`binlog_file`*`, `*`pos`*`)` only when both the file and position parameters are known to be valid. (Bug #33818238)
- **Replication:** Operations using the SID of a invalid GTID log event sometimes led to undefined behavior.

  We solve this issue by ensuring that the SID is always initialized, even when it belongs to an invalid GTID log event. (Bug #33784022)
- **Replication:** Calculation of the maximum length of status variables (`MAX_SIZE_LOG_EVENT_STATUS`) was not performed correctly.

  Issues corrected in this fix are listed here:

  - `sql_require_primary_key` and `default_table_encryption` require 2 bytes (one for type ID, one for the variable)
  - The maximum length of `time_zone` was calculated as 255 instead of `MAX_TIME_ZONE_NAME_LENGTH`
  - An additional byte (each) is required to save the length of the user variable, the number of database IDs stored in `binlog_accessed_db_names`, and the type of microseconds

  In addition, `master_data_written`, which was not actually used, has been removed. (Bug #33713071)
- **Replication:** On 32-bit platforms, setting `binlog_expire_logs_seconds` to `2147483648` ($2^{31}$) or more had the effect of causing all binary logs to be purged before the specified time had elapsed. (Bug #33670457)

- **Replication:** If a group configuration change is in progress, new members cannot join. The error message issued in this scenario now states what operation is in progress, such as a primary election, a swap between single primary mode and multi-primary mode, or a change of group communication protocol. (Bug #32766981)

- **Replication:** When the binary log sender thread waited for updates with heartbeats enabled, it sometimes missed update signals, so that changes were not replicated until the next signal was issued and noticed by the thread.

  Our thanks to Facebook for the contribution. (Bug #104491, Bug #33176377)

- **Group Replication:** The `START REPLICA` and `STOP REPLICA` statements were permitted for the Group Replication applier channel (`group_replication_applier`), but stopping and starting the channel manually can leave online members of the replication group incapable of committing remote transactions. These statements can no longer be used for the channel when Group Replication is running. (Bug #34231291)

- **Group Replication:** A deadlock could occur when Group Replication was started on a server instance, but errors such as incompatible configuration caused the member to leave immediately. The service registry now manages the registration and deregistration of services by an alternative process that avoids the issue. (Bug #34194893)

- **Group Replication:** In an InnoDB ClusterSet, in the event of a commit order deadlock and subsequent retry, Group Replication could roll back a transaction during certification with an error stating that the GTID for the transaction had already been used. The error is now only returned if the GTID has been used on the server instance as well as in the group. (Bug #34157846)

- **Group Replication:** After checking a transaction commit has no conflicts and is in the correct order, Group Replication reports back to the committing session. When the event scheduler thread was started, Group Replication was not able to find the committing session, resulting in the member entering `ERROR` state and leaving the group. The procedure to locate the committing session was extended to find daemon threads, as used to start the event scheduler thread. Thanks to Lou Shuai for the contribution. (Bug #107635, Bug #34312482)

- Calling many stored procedures from the same connection led the server to consume an ever-increasing amount of virtual memory. (Bug #35110491)

- When the system variable `replication_optimize_for_static_plugin_config` was enabled to improve performance for semisynchronous replication, the `RAPID` plugin could not be used in the server due to issues with notification handling. The issue has now been fixed. (Bug #34549470)

- Strings containing many leading zeros were incorrectly marked as overflowing when converting them to decimal, causing the value returned to be the maximum decimal value for the given precision and scale.

  We fix this by replacing all leading zeros in such cases with a single `0`, prior to conversion. (Bug #34479194)

- Differences between the `WSAPoll` function on Windows and the `poll()` function on Unix-like systems caused thread pool code on Windows to consider invalid file descriptors or sockets as ready for I/O. (Bug #34478731)

- If a MySQL instance at MySQL 8.0.27 or above joined a replication group running at MySQL 8.0.26, the member action `mysql_start_failover_channels_if_primary` was removed to match the member actions configuration for to the rest of the group. However, if the instances were upgraded to run at MySQL 8.0.27 or above, the member action was not reinstated unless `group_replication_reset_member_actions()` was issued on a server that then bootstrapped the group. Joining members now check if the member action

`mysql_start_failover_channels_if_primary` is available to them, and add it to their member actions configuration if it is. (Bug #34464526)

- A constant expression used in the `SELECT` list of a query was not always handled correctly by the server. (Bug #34418412)
- Added a missing error check in `Json_table_column::fill_column()`. (Bug #34416304)
- Some string functions accepting multiple arguments did not process all of the arguments properly. (Bug #34393009)
- Columns of type `YEAR` created for temporary tables holding `UNION` results were not always guaranteed to be of the correct length. (Bug #34370933)
- If a view (`v1`) accessed another view (`v2`), and if `v2` was recreated, then `SHOW COLUMNS FROM v1` reported an invalid view error. This issue could occur only when the user was granted global privileges with roles, but not table-level privileges. It is fixed by checking for global privileges if table-level privileges are absent. (Bug #34341533)
- Queries on the Performance Schema table `replication_group_member_actions` could fail due to conflicts between threads. The issue has now been fixed. (Bug #34337673)
- Added a missing error check after an item evaluation when reading const tables. (Bug #34327798)
- Reparsing and cloning of derived tables was not always handled correctly in certain cases. (Bug #34317258, Bug #34408241)
- Pushing a condition down to a derived table with set operations when multiple query blocks were present was not always handled correctly. (Bug #34306361)
- A startup failure during RSA key creation resulted in empty or incomplete keys. Keys are now generated in a fail-safe manner. (Bug #34274914)
- When comparing two string values with explicit `COLLATE` clauses, these should have the same collation, else the comparison should be rejected, but this was not always enforced.

  For more information, see Collation Coercibility in Expressions. (Bug #34274525)
- A `CREATE USER` operation read from binary log failed to execute on a replica server due to an unexpected syntax error. (Bug #34178823)
- A precision mismatch could sometimes occur in a `CAST` to `FLOAT` operation on an integer value, when using `SQL_SMALL_RESULT` as opposed to `SQL_BIG_RESULT` (see SELECT Statement).

  For `SQL_BIG_RESULT`, the value was truncated according to the inherent precision of an IEEE 32 bit floating point value, which is the correct behavior. With `SQL_SMALL_RESULT`, the precision from the original integer value was preserved; this happened because the value was copied to a temporary table which supported only double precision values, and these values were later copied back to the client.

  We fix this by distinguishing the type of floating point value when copying such values to a temporary table, so that double precision values are stored as type double and single precision values are stored as type float. (Bug #34167787)
- Executed `codespell` on the majority of source code, and fixed the reported spelling errors in the code comments. (Bug #34163323)
- In certain cases, when checking an internal function pointer before it was called, the wrong pointer was validated. (Bug #34134738)

- Both the access path and the iterator for `eq_ref` access had a `use_order` member for determining whether to use an ordered index scan. Since `eq_ref` always returns at most one row, ordering it is not necessary; thus, `use_order` serves no purpose here, and has been removed from the access path and the corresponding iterator. (Bug #34122706)

- Improved the `ADD_GDB_INDEX` `CMake` option to also function when loading unit tests. (Bug #34117732)

  References: See also: Bug #29925009.

- Removed GNU 'gold' linker support, including the associated USE_LD_GOLD CMake option. (Bug #34094324)

- Manually created directories in a MySQL 5.7 data directory (treated as schema directories in MySQL 5.7) with special characters in the directory name or schemas and tables created with special characters in their name using a `` `#mysql50#` `` prefix caused DDL failures after upgrading to MySQL 8.0. In MYSQL 8.0, the special characters are replaced with question mark characters (`?`) when the object names are added to the data dictionary. The name mismatch caused DDL operations on the affected objects to fail. Upgrade now stops with an error if invalid schema and table names are encountered.

  In MySQL 5.7, you can update invalid schema and table names to a valid encoding using `mysqlcheck`. To identify invalid names, run:

  ```
  mysqlcheck --check-upgrade --all-databases
  ```

  To fix invalid names, run:

  ```
  mysqlcheck --fix-db-names --fix-table-names --all-databases
  ```

  In MySQL 5.7, directory names that have special characters can also be updated to use valid encoding using the following SQL syntax:

  ```
  ALTER DATABASE `directory_name` UPGRADE DATA DIRECTORY NAME;
  ```

  (Bug #34066605)

- Conditions using the null-safe equals operator (`<=>`) can now considered by the executor as equijoin conditions. Predicates using such conditions are not considered as equijoin predicates, but rather as extra predicates to evaluate following the join. (Bug #34063499)

- Support was added for compiling the MySQL Enterprise Encryption component with OpenSSL 3. (Bug #34043029)

- Executed `codespell` on the source code and fixed the reported spelling errors in the code comments. (Bug #34006439)

- When one or more arguments to `GREATEST()` were temporal values, the order in which they were passed to the function could affect the result. (Bug #33996054)

- A `SELECT` with a `LEFT JOIN` on a derived table containing a nested subquery as part of a `WHERE` clause with an `IN` predicate returned incorrect results. (Bug #33971286)

- Calls to `ERR_clear_error()` in the `vio_ssl_write()` and `ssl_handshake_loop()` functions caused errors when compiling with OpenSSL 3.0. (Bug #33967647)

- The server did not always process nested views as expected. (Bug #33876690)

- `mysqlpump` might not be given the correct permissions to use derived tables (tables that are generated by a query `FROM` clause), causing the dump process to stop if these were present. Derived tables are now handled separately and privileges are set for them. (Bug #33866103)

- Repeated execution of a stored routine having as a subquery a `SELECT` containing multiple `AND`, `OR`, or `XOR` conditions led to excessive consumption and possibly eventual exhaustion of virtual memory. (Bug #33852530)

- `EXPLAIN ANALYZE` provides information about the estimated cost and elapsed time for each iterator in the query plan. We expect these numbers to be cumulative, so that the cost (or time) for a given iterator includes the cost of any iterators on which this iterator depends. For example, the estimated cost of joining table `t1` with table `t2`, using a table scan of `t1` followed by a primary key lookup on `t2`, should be no less than the cost of scanning `t1` plus the cost of the cardinality lookups on `t2`.

  For some queries involving materialization or temporary tables, these numbers were not cumulative. In addition, cost estimates were incorrect when materializing the result of a `UNION` operation and subsequently performing a table scan on the temporary (materialized) table. (Bug #33834146, Bug #34302461)

- Coercion of collations for comparisons of two string values was dependent upon the order of the operands. Given one string using the `utf8mb4` character set and one using `utf8mb3`, the comparison was carried out using the `utf8mb4` collation when the left operand was `utf8mb4`, and `utf8mb3` when the left operand was `utf8mb3`.

  The root of the problem was with the special handling used for coercion of ASCII strings; since both strings were literals, they were scanned for ASCII and non-ASCII characters, and the repertoire property set according to this. This meant that it was possible for one string to be treated as ASCII while the other string was not; when the `utf8mb4` string was considered ASCII and thus compatible with the `utf8mb3` string it was (wrongly) converted prior to the comparison.

  We fix this by using the internal `MY_CS_PUREASCII` property rather than `MY_REPERTOIRE_ASCII`; `MY_CS_PUREASCII` is derived strictly from the character set, and not the actual string, which makes determination of collation deterministic. Now in such cases, neither string is identified as ASCII, which leads to conversion of the `utf8mb3` to `utf8mb4` before the comparison is performed, as expected. (Bug #33801312)

- The scope of the `connect_timeout` limit was extended to full-packet reads. (Bug #33723597)

  References: See also: Bug #34574013.

- New validation was added to ensure that the Event Scheduler restarts automatically on the server as expected. (Bug #33711304)

  References: See also: Bug #33539082.

- When a prepared statement was executed with no cursor, it could return the wrong result if the previous execution used a cursor. This issue is fixed by creating and saving two query result objects, one for use with a cursor and one for use without a cursor. The objects are created on demand, so if a prepared statement is never used with cursors, the cursor-specific query result object is never created. (Bug #33535746)

- A prepared statement with parameters could fail to update a row, but the same statement with the same data did update the row when issued as a query. The fix for the problem is to assign a default data type to the parameters, although this can be inefficient because there is no context available for data type propagation and a character string type is given implicitly. In this case, the best practice is to wrap such parameter declarations in `CAST` clauses that supply the desired data types. (Bug #33521861)

- For a table using the FEDERATED storage engine, if consecutive queries on the local server had a delay between them that exceeded the remote server's wait timeout (`wait_timeout` setting), a packet error was returned on the local server. The timeout error from the remote server is now handled correctly by the local server, and it reconnects and executes the statement again. (Bug #33500956)

- Handled an unexpected condition which could arise during join optimization. (Bug #31401468)

  References: See also: Bug #34350945.

- Following execution of `SET @@character_set_client = @@character_set_system`, optimizer hints were not parsed correctly. (Bug #107820, Bug #34370403)

- A query performing a full-text search in boolean mode on a column of a `MyISAM` table led to an assertion in debug builds when the column was not of a string type. This was due to the fact that `MyISAM` allows `MATCH` columns without indexes in boolean mode, and therefore also allows integer columns, which are subject to constant propagation. Such queries on tables using any storage engine other than `MyISAM` were not affected by this issue.

  This is fixed by disabling constant propagation on the `MATCH` clause when the column is not of a string type, so that we can safely assume that it contains only column references. (Bug #107638, Bug #34311472)

- Corrected errors in comments in `my_time.cc`. Our thanks to Dimitry Kudryavtsev for the contribution. (Bug #107633, Bug #34308417)

- Following changes to connection error handling in MySQL 8.0.24, if an error such as exceeding `max_allowed_packet` caused the `mysql` client to disconnect from the server, the error was not reset afterwards to indicate the disconnect. On subsequent queries, the original error was returned, and automatic reconnection did not take place because the `mysql` client did not have a disconnection error. The error is now reset to indicate disconnection and allow clients to reconnect. (Bug #107605, Bug #34303841)

- Following a bug fix in MySQL 8.0.29, if the password was provided in an option file, the `mysql` client did not prompt for a password if the `-p` option was specified at login. The client now always prompts for a password when the `-p` option is specified and uses the specified password, even if an alternative was populated from an option file. (Bug #107205, Bug #34134984)

- The alternative for the deprecated `--ssl=off` server option to disable the use of encrypted connections was not working as documented, or as presented in the deprecation warning. Setting the `tls_version` system variable to the empty value (`tls_version=''`) now works correctly for this purpose. (Bug #106459, Bug #33858646)

- An aggregate query on a `BIT` column returned a value that was formatted as a bit string, but also had the `BINARY` flag added automatically. New validation now checks for and skips setting the `BINARY` flag for `BIT` results. (Bug #106241, Bug #33781442)

- `SHOW SLAVE STATUS` and `SHOW SLAVE HOSTS` used more CPU following the introduction of `SHOW REPLICA STATUS` and `SHOW REPLICA HOSTS` than previously. (Bug #105924, Bug #33684069)

- When a `NO_SKIP_SCAN` hint referred to a specific index not to be used for a table skip scan, all other possible indexes were also ignored, and, thus, a skip scan was not used for any indexes on the table.

  This occurred because processing was not performed for all possible keys for a skip scan if the `NO_SKIP_SCAN` hint was not applicable to all indexes. (Bug #104670, Bug #33251616)

- Queries using `WHERE column IN (list)` took an increasingly excess amount of CPU time as the list of values grew in length. (Bug #102037, Bug #32311183)

# Changes in MySQL 8.0.30 (2022-07-26)

- Character Set Support
- Compilation Notes

- Deprecation and Removal Notes
- Generated Invisible Primary Keys (GIPKs)
- Keyring Notes
- Performance Schema Notes
- Pluggable Authentication
- Security Notes
- Spatial Data Support
- SQL Syntax Notes
- XA Transaction Notes
- Functionality Added or Changed
- Bugs Fixed

## Character Set Support

- **Important Change:** A previous change renamed character sets having deprecated names prefixed with `utf8_` to use `utf8mb3_` instead. In this release, we rename the `utf8_` collations as well, using the `utf8mb3_` prefix; this is to make the collation names consistent with those of the character sets, not to rely any longer on the deprecated collation names, and to clarify the distinction between `utf8mb3` and `utf8mb4`. The names using the `utf8mb3_` prefix are now used exclusively for these collations in the output of `SHOW` statements such as `SHOW CREATE TABLE`, as well as in the values displayed in the columns of Information Schema tables including the `COLLATIONS` and `COLUMNS` tables. (Bug #33787300)

  References: See also: Bug #30624990.

- **Important Change:** When more than one language had the same collation definition, MySQL implemented collations for only one of the languages. This meant that some languages were covered only by `utf8mb4` Unicode 9.0 collations that are specific to other languages. This release fixes such issues by adding language-specific collations for those languages that were previously covered only by language-specific collations for other languages. The new collations are listed here:

  - *Norwegian*
    - (Bokmål) `utf8mb4_nb_0900_ai_ci`
    - (Bokmål) `utf8mb4_nb_0900_as_cs`
    - (Nynorsk) `utf8mb4_nn_0900_ai_ci`
    - (Nynorsk) `utf8mb4_nn_0900_as_cs`
  - *Serbian with Latin characters*:
    - `utf8mb4_sr_latn_0900_ai_ci`
    - `utf8mb4_sr_latn_0900_as_cs`
  - *Bosnian with Latin characters*:

    - `utf8mb4_bs_0900_ai_ci`
    - `utf8mb4_bs_0900_as_cs`
  - *Bulgarian*:
    - `utf8mb4_bg_0900_ai_ci`
    - `utf8mb4_bg_0900_as_cs`
  - *Galician*:
    - `utf8mb4_gl_0900_ai_ci`
    - `utf8mb4_gl_0900_as_cs`
  - *Mongolian with Cyrillic letters*:
    - `utf8mb4_mn_cyrl_0900_ai_ci`
    - `utf8mb4_mn_cyrl_0900_as_cs`

  For more information, see Language-Specific Collations. (Bug #31885256, WL #14307)

## Compilation Notes

- On Enterprise Linux, fixed ADD_LINUX_RPM_FLAGS so that the initial values of CMAKE_C_FLAGS and CMAKE_CXX_FLAGS are used before modifying them. (Bug #34131794)

  References: This issue is a regression of: Bug #33730302.

- Added a new SHOW_SUPPRESSED_COMPILER_WARNINGS CMake option. Enable it to show suppressed compiler warnings, and do so without failing with -Werror. It defaults to OFF. (Bug #34046748)
- Added macOS/ARM support. (Bug #34017614)
- On Windows, deprecation warnings (C4996) were globally disabled with the /wd4996 command-line option; now deprecation warnings are disabled on the localized level where appropriate. (Bug #33975638)
- On Windows, improved the generated INFO_BIN and INFO_SRC files. (Bug #33972317, Bug #34052301)
- Improved GCC 8 support to include -lstdc++fs in order to use std::filesystem. (Bug #33939798)

## Deprecation and Removal Notes

- **Replication:** Setting the `replica_parallel_workers` system variable (or the equivalent server option `--replica-parallel-workers`) to 0 is now deprecated, and doing so now raises a warning.

  To achieve the same result (that is, use single threading) without the warning, set `replica_parallel_workers=1` instead. (WL #13956)

- The `--skip-host-cache` server option is now deprecated, and subject to removal in a future release.

  Use a statement such as `SET GLOBAL host_cache_size = 0`, or set `host_cache_size` in the `my.cnf` file, instead. (WL #14359)

- The `--old-style-user-limits` option causes the server to enforce user limits as they were prior to MySQL 5.0.3, and is intended for backwards compatibility with very old releases. This option is now deprecated, and using it now raises a warning. You should expect this option to be removed in a future release of MySQL, and so you are advised to begin now to remove any dependency your MySQL applications might have on this option. (WL #13228)

## Generated Invisible Primary Keys (GIPKs)

- MySQL 8.0.30 now supports GIPK mode, which causes a generated invisible primary key (GIPK) to be added to any `InnoDB` table that is created without an explicit primary key. This enhancement applies to `InnoDB` tables only.

  The definition of the generated key column added to an `InnoDB` table by GIPK mode is shown here:

  ```
  my_row_id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT INVISIBLE PRIMARY KEY
  ```

  The name of the generated primary key is always `my_row_id`; you cannot, while GIPK mode is in effect, use this as a column name in a `CREATE TABLE` statement that creates a new `InnoDB` table unless it includes an explicit primary key.

  GIPKs are not enabled by default. To enable them, set the `sql_generate_invisible_primary_key` server system variable (also introduced in this release) to `ON`. This setting has no effect on replication applier threads; this means that a replica never generates a primary key for a replicated table that was not created on the source with a primary key.

  You cannot alter a generated invisible primary key while GIPKs are in effect, with one exception: You can toggle the visibility of the GIPK using `ALTER TABLE tbl CHANGE COLUMN my_row_id SET VISIBLE` and `ALTER TABLE tbl CHANGE COLUMN my_row_id SET INVISIBLE`.

  By default, generated invisible primary keys can be seen in the output of `SHOW CREATE TABLE` and `SHOW INDEX`; they are also visible in MySQL Information Schema tables such as the `COLUMNS` and `STATISTICS` tables. You can make them hidden instead by setting `show_gipk_in_create_table_and_information_schema` to `OFF`.

  You can exclude generated invisible primary keys from the output of `mysqldump` using the `--skip-generated-invisible-primary-key` option added in this release. `mysqlpump` also now supports a `--skip-generated-invisible-primary-key` option which excludes GIPKs from its output.

  For more information and examples, see Generated Invisible Primary Keys. For general information on invisible column support in MySQL, see Invisible Columns. (Bug #34092605, WL #13784)

## Keyring Notes

- The `keyring_aws` plugin has been updated to use the latest AWS Encryption SDK for C (version 1.9.186).

  The `keyring_aws_region` variable supports the additional AWS regions supported by the new SDK. Refer to the variable description for a list of supported AWS regions. (WL #14547)

## Performance Schema Notes

- **InnoDB:** In the Performance Schema instrumentation for InnoDB read-write locks, lock acquisition failures and successes for TRY (no-wait) operations were instrumented incorrectly. (Bug #34131395)
- **Group Replication:** Performance Schema now provides instrumentation for performance monitoring of Group Replication memory usage.

See Monitoring Group Replication Memory Usage with Performance Schema Memory Instrumentation. (WL #14726)

## Pluggable Authentication

- The SASL LDAP plugin failed to properly parse Kerberos Key Distribution Center (KDC) host information read from the Kerberos configuration file, resulting in SASL authentication error. (Bug #31862170)

## Security Notes

- It is now possible to compile the MySQL server package (`mysqld` + `libmysql` + client tools) using OpenSSL 3.0 on supported platforms, which should not change the behavior of the server or client programs. For additional information, see https://wiki.openssl.org/index.php/OpenSSL_3.0. (WL #14683)

## Spatial Data Support

- Previously, the `ST_TRANSFORM()` function added in MySQL 8.0.13 did not support Cartesian Spatial Reference Systems. Beginning with this release, support is provided by this function for the Popular Visualisation Pseudo Mercator (EPSG 1024) projection method, used for WGS 84 Pseudo-Mercator (SRID 3857). (WL #11961)

## SQL Syntax Notes

- It is now possible to determine whether a `REVOKE` statement which cannot be executed raises an error or a warning. This is implemented with the addition of two new statement options, listed here with brief descriptions:

  - `IF EXISTS` causes `REVOKE` to raise a warning rather than an error as long as the target user or role does not exist.

  - `IGNORE UNKNOWN USER` causes `REVOKE` to raise a warning instead of an error if the target user or role is not known, but the statement would otherwise succeed.

  For a single target user or role and a given privilege or role to be removed, using the `IF EXISTS` and `IGNORE UNKNOWN USER` options together in the same `REVOKE` statement means that the statement succeeds (albeit doing nothing, and with a warning), even if both the target user or role and the privilege or role to be removed are unknown, as long as the statement is otherwise valid. In the case of multiple targets, multiple privileges or roles to be removed, or both, the statement succeeds, performing those removals which are valid, and issuing warnings for those which are not.

  For more information, see REVOKE Statement. (Bug #102232, Bug #32495441, WL #14690)

## XA Transaction Notes

- **Replication; Group Replication:** Previously, recovery was not guaranteed when a server node in a replication topology unexpectedly halted while executing `XA PREPARE`, `XA COMMIT`, or `XA ROLLBACK`. To address this problem, MySQL now maintains consistent XA transaction state across a topology using either MySQL “classic” Replication or MySQL Group Replication when a server node is lost from the topology and then regained. This also means that XA transaction state is now propagated so that nodes do not diverge while doing work within a given transaction in the event that a server node halts, recovers, and rejoins the topology.

  For any multi-server replication topology (including one using Group Replication), the XA transaction state propagates consistently, so that all servers remain in the same state at all times. For any such

topology of any size (including a single server, as long as binary logging is enabled), it is now possible to recover any server to a consistent state after it has halted unexpectedly and been made to rejoin the topology after dropping out.

This enhancement is implemented for the case of a single server by adding support for a two-phase XA prepare between the storage engine and the server's internal transaction coordinator (ITC), with the state of the prepare retained by both. This means that the ITC can purge its internal logs safely, without the risk of losing state, should the server halt following the purge. In the single-node case, imposing order of execution between the storage engine and the binary log prevents externalization of GTIDs before the corresponding changes become visible to the storage engine; in a topology comprising multiple servers, this keeps the transaction state from being broadcast to the topology before it is guaranteed to be locally consistent and persistent. In all cases, the state of the XA transaction is extracted from the last binary log file to be written and synchronized with the transaction state obtained from the storage engine.

A known issue in this release can be encountered when the same transaction XID has been used to execute XA transactions sequentially. If there a disruption in operation occurs while the server is processing `XA COMMIT ... ONE PHASE` using this same XID, after the transaction has been prepared in the storage engine, the state between the binary log and the storage engine can no longer be reliably synchronized.

For more information, see XA Transactions. (WL #11300)

## Functionality Added or Changed

- **Important Change:** Binary packages that include curl rather than linking to the system curl library have been upgraded to use curl 7.83.1. (Bug #34138733)

- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.1.1o. Issues fixed in OpenSSL version 1.1.1o are described at https://www.openssl.org/news/cl111.txt and https://www.openssl.org/news/vulnerabilities.html. (Bug #34133985)

- **Important Change:** The `fido2` library included with MySQL, used with the `authentication_fido` plugin, has been upgraded to version 1.8.0. (Previously, version 1.5.0 was included with MySQL.) (WL #15111)

- **InnoDB:** The `innodb_doublewrite` system variable, which enables or disables the doublewrite buffer, has two new settings, `DETECT_ONLY` and `DETECT_AND_RECOVER`. With the `DETECT_ONLY` setting, database page content is not written to the doublewrite buffer, and recovery does not use the doublewrite buffer to fix incomplete page writes. This lightweight setting is intended for detecting incomplete page writes only. The `DETECT_AND_RECOVER` setting is equivalent to the existing `ON` setting. For more information, see Doublewrite Buffer.

  Thanks to Facebook for the contribution. (Bug #32727919, Bug #103211, WL #14719)

- **InnoDB:** `InnoDB` now supports dynamic configuration of redo log capacity. The `innodb_redo_log_capacity` system variable can be set at runtime to increase or decrease the total amount of disk space occupied by redo log files.

  With this change, the number of redo log files and their default location has also changed. From MySQL 8.0.30, `InnoDB` maintains 32 redo log files in the `#innodb_redo` directory in the data directory. Previously, `InnoDB` created two redo log files in the data directory by default, and the number and size of redo log files were controlled by the `innodb_log_files_in_group` and `innodb_log_file_size` variables. These two variables are now deprecated.

When the `innodb_redo_log_capacity` setting is defined, `innodb_log_files_in_group` and `innodb_log_file_size` settings are ignored; otherwise, those settings are used to compute the `innodb_redo_log_capacity` setting (`innodb_log_files_in_group` * `innodb_log_file_size` = `innodb_redo_log_capacity`). If none of those variables are set, redo log capacity is set to the `innodb_redo_log_capacity` default value, which is 104857600 bytes (100MB).

Several status variables are provided for monitoring the redo log and redo log capacity resize operations.

As is generally required for any upgrade, this change requires a clean shutdown before upgrading.

For more information about this feature, see Redo Log. (WL #12527)

- Added Ubuntu 22.04 support. (Bug #34123545)
- The order of the columns in the primary key definition for a few tables in the `mysql` schema has been changed, so that the columns containing the host name and user name are together in sequence at the start of the primary key. ACL queries on these tables are performed using only the host name and user name, and if those columns are not together in sequence, a full table scan must be performed to identify the relevant record. Placing the host name and user name together means that index lookup can be used, which improves performance for `CREATE USER`, `DROP USER`, and `RENAME USER` statements, and for ACL checks for multiple users with multiple privileges.

  The changed tables are `mysql.db`, `mysql.tables_priv`, `mysql.columns_priv` and `mysql.procs_priv`. When you upgrade to MySQL 8.0.30 or later, these tables are modified in the second step of the MySQL upgrade process. Use the `--upgrade=FORCE` option when performing logical upgrades using a backup or export utility such as `mysqldump` or `mysqlpump`, which ensures that the table structures are checked and rebuilt with the new column order. (Bug #33644645, Bug #33637244, WL #14965)
- The `myisam_repair_threads` system variable and `myisamchk --parallel-recover` option were removed. (Bug #31052408, WL #14938)
- A new `mysqldump` option `--mysqld-long-query-time` lets you set a custom value of the `long_query_time` system variable for `mysqldump`'s session. Use the new option to increase the elapsed time allowed for `mysqldump`'s queries before they are written to the slow query log file, in order to avoid unnecessary logging. Thanks to Facebook for the contribution. (Bug #96369, Bug #96369, Bug #30110717, WL #13447)

- Error log components can now be loaded implicitly at startup before the `InnoDB` storage engine is available. This new method of loading error log components loads and enables the components defined by the `log_error_services` variable.

  Previously, error log components had to be installed first using `INSTALL COMPONENT` and were only loaded after `InnoDB` was fully available, as the list of components to load was read from the `mysql.components` table, which is an `InnoDB` table.

  Implicit load of error log components has these advantages:

  - Log components are loaded early in the startup sequence, making logged information available sooner.
  - It helps avoid loss of buffered log information should a failure occur during startup.
  - Loading log components using `INSTALL COMPONENT` is not required, simplifying error log configuration.

  For more information about this feature, see Error Log Configuration.

  If you have previously installed loadable log components using `INSTALL COMPONENT` and you list those components in a `log_error_services` setting that is read at startup (from an option file, for example), your configuration should be updated to avoid startup warnings. For more information, see Error Log Configuration Methods. (WL #14793)

- MySQL Enterprise Audit's audit log file can now be extended with optional data fields to show the query time, the number of bytes sent and received, the number of rows returned to the client, and the number of rows examined. This data is available in the slow query log for qualifying queries, and in the context of the audit log it similarly helps to detect outliers for activity analysis. It is delivered to the audit log through new component services that you set up as an audit log filtering function. The extended data fields can only be added when the audit log is in JSON format (`audit_log_format=JSON`), which is not the default setting. (WL #14921)

- MySQL Server's `AES_ENCRYPT()` and `AES_DECRYPT()` functions now support the use of a key derivation function (KDF) to create a cryptographically strong secret key from information such as a password or a passphrase that you pass to the function. The derived key is used to encrypt and decrypt the data, and it remains in the MySQL Server instance and is not accessible to users. Using a KDF is highly recommended, as it provides better security than specifying your own premade key or deriving it by a simpler method when you use the function. The functions support HKDF (available from OpenSSL 1.1.0), for which you can specify an optional salt and context-specific information to include in the keying material, and PBKDF2 (available from OpenSSL 1.0.2), for which you can specify an optional salt and set the number of iterations used to produce the key. (WL #12669, WL #15188)

- A new system status variable `Tls_library_version` shows the runtime version of the OpenSSL library that is in use for the MySQL instance. The version of OpenSSL affects features such as support for TLSv1.3. (WL #13407)

- From MySQL 8.0.30, MySQL Enterprise Encryption's functions are provided by a component, rather than being installed from the `openssl_udf` shared library. The new functions provided by the component use only the generally preferred RSA algorithm, not the DSA algorithm or the Diffie-Hellman key exchange method, and they follow current best practice on minimum key size. The component functions also add support for SHA3 for digests (provided that OpenSSL 1.1.1 is in use), and do not require digests for signatures, although they support them.

  If you upgrade to MySQL 8.0.30 from an earlier release where the functions are installed manually from the `openssl_udf` shared library file, the functions you created remain available and are supported.

However, these legacy functions are deprecated from this release, and it is recommended that you install the component instead. The component functions are backward compatible, so RSA public and private keys, encrypted data, and signatures that were produced by the legacy functions can be used with the component functions. For the component functions to support decryption and verification for content produced by the legacy functions, you must set the new system variable `enterprise_encryption.rsa_support_legacy_padding` to `ON` (the default is `OFF`).

The component functions generate public and private RSA keys in PKCS #8 format. They allow a minimum key size of 2048 bits, which is a suitable minimum RSA key length for current best practice. You can set a maximum key size up to 16384 bits using the system variable `enterprise_encryption.maximum_rsa_key_size`, which defaults to a maximum key size of 4096 bits. (WL #15024)

- Connections whose users have the `CONNECTION_ADMIN` privilege are not terminated when MySQL Server is set to offline mode, which is done by changing the value of the `offline_mode` system variable to `ON`. Previously, checking for connections that had the `CONNECTION_ADMIN` privilege could cause a race condition because it involved accessing other threads. Now, a flag for each thread caches whether or not the user for the thread has the `CONNECTION_ADMIN` privilege. The flag is updated if the user privilege changes. When offline mode is activated for the server, this flag is checked for each thread, rather than the security context of another thread. This change makes the operation threadsafe.

  In addition, when offline mode is activated, connections whose users have the `SYSTEM_USER` privilege are now only terminated if the user that runs the operation also has the `SYSTEM_USER` privilege. Users that only have the `SYSTEM_VARIABLES_ADMIN` privilege, and do not have the `SYSTEM_USER` privilege, can set the `offline_mode` system variable to `ON` to activate offline mode. However, when they run the operation, any sessions whose users have the `SYSTEM_USER` privilege remain connected, in addition to any sessions whose users have the `CONNECTION_ADMIN` privilege. This only applies to existing connections at the time of the operation; users with the `SYSTEM_USER` privilege but without the `CONNECTION_ADMIN` privilege cannot make new connections to a system in offline mode. (WL #14317)

## Bugs Fixed

- **InnoDB:** A `TRUNCATE TABLE` operation failed to remove data dictionary entries for columns that were dropped using `ALGORITHM=INSTANT`.

  Thanks to Marcelo Altmann for the contribution. (Bug #34302445)

- **InnoDB:** An incorrect nullable column calculation on tables with instantly added columns caused data to be interpreted incorrectly. (Bug #34243694)

- **InnoDB:** After upgrading to MySQL 8.0.29, a failure occurred when attempting to access a table with an instantly added column. (Bug #34233264)

- **InnoDB:** Only the physical position of instantly added columns was logged, which was not sufficient for index recovery. The logical position of columns was also required. (Bug #34181432)

- **InnoDB:** The `field_phy_pos` debug variable in the `InnoDB` sources was not updated for child tables during a cascading update operation. (Bug #34181419)

- **InnoDB:** Some instances of the `rec_get_instant_row_version_old()` function in the InnoDB sources did not check for row versioning. (Bug #34173616)

- **InnoDB:** The `read_2_bytes()` function in the InnoDB sources, which reads bytes from the log buffer, returned a null pointer. (Bug #34173425)

- **InnoDB:** In a specific locking scenario, an implicit lock was not converted to an explicit lock as expected, triggering a `lock_rec_has_expl(LOCK_X | LOCK_REC_NOT_GAP, block, heap_no, trx)` debug assertion failure. (Bug #34123159)
- **InnoDB:** A check that determines if a table has instantly added columns was performed for each column, which affected the performance of `ADD` and `DROP COLUMN` operations on tables with numerous columns. The check is now performed once per table. (Bug #34112147)
- **InnoDB:** A workload that generated a large number of lock requests and numerous timeouts caused a long semaphore wait failure. To address this issue, optimizations were implemented to reduce the number of exclusive global lock system latches. (Bug #34097862)
- **InnoDB:** The `m_flush_bit` in the redo log block header, which was set for the first block of multiple blocks written in a single log write call, provided no benefit and has been removed. (Bug #34091444)
- **InnoDB:** Fixed clang-tidy and cppcheck warnings, which included the removal of unused code and unnecessary checks. (Bug #33957087)
- **InnoDB:** Recovery of a redo log file mini-transaction (`mtr`) caused a debug assertion failure on a MySQL Server instance with a small `innodb_log_buffer_size` setting.

  Thanks to Mengchu Shi for the contribution. (Bug #33945602)
- **InnoDB:** Compiling with the `WITH_VALGRIND` source configuration option produced `Wunused-variable` warnings. (Bug #33899862)
- **InnoDB:** Multiple issues with the lock-free hash table (`ut_lock_free_hash_t`) were addressed. (Bug #33830934)
- **InnoDB:** A query on a generated column with a secondary index caused a failure. The field number representing the position of the generated column was not valid. (Bug #33825077)
- **InnoDB:** Memory consumption was greater than expected when updating and inserting rows with multi-valued index columns. The memory allocated for multi-valued columns for each row update was held until the file handle was released. (Bug #33766482)
- **InnoDB:** The `UT_LOCATION_HERE` structure in the `InnoDB` sources was not used consistently. (Bug #33436161)
- **InnoDB:** A table object needed to retrieve an array of values from a multi-valued index column when computing the value of a generated column was unavailable. (Bug #32725063)
- **InnoDB:** A 4GB tablespace file size limit on Windows 32-bit systems has been removed. The limit was due to an incorrect calculation performed while extending the tablespace. (Bug #28934351)
- **InnoDB:** Hash and random generator functions in the `InnoDB` sources were improved. (Bug #16739204, Bug #23584861)
- **InnoDB:** A `DROP TABLE` operation on a table with a discarded tablespace caused an unnecessary assertion failure. (Bug #107207, Bug #34135187)
- **InnoDB:** A query on a table with a `JSON` column returned only a partial result set after adding a multi-valued index. (Bug #106621, Bug #33917625)
- **InnoDB:** Purging a record with multiple binary large object values raised an insertion failure due to a mini-transaction (mtr) conflict. (Bug #105592, Bug #33574272)
- **InnoDB:** Enabling the adaptive hash index (AHI) on a high-concurrency instance caused temporary AHI search latch contention while the hash index was being built.

Thanks to Zhou Xinjing from CDB Team at Tencent for the patch. (Bug #100512, Bug #31750840)

- **Packaging:** The SASL LDAP clientside plugin was missing from the MySQL Community packages for Windows.
- **Replication:** When a table definition diverged between the source and the replica because the replica had an extra primary key, updates and deletes on the replica would fail if that table had an index that was present both on the source and the replica. Primary keys for an InnoDB table are automatically included in all indexes, and the replication applier needs values for all parts of the key to be included in an event in order to search the index. Previously, the applier checked that all the user-defined key parts were present, but the check did not cover hidden primary keys that were automatically included. The applier now validates that both user-defined and automatically included key parts are present in an event before using the index to search the data. (Bug #34122738)
- **Replication:** The write sets extracted by MySQL Replication from transactions when the `transaction_write_set_extraction` system variable is enabled (which is the default) are extracted from primary keys, unique keys, and foreign keys. They are used to detect dependencies and conflicts between transactions. Previously, write sets involving multi-column foreign keys were incorrectly identifying each column as a separate foreign key. The issue has now been fixed and foreign key write sets include all referenced key columns. (Bug #34095747, Bug #34144531)
- **Replication:** When row-based replication was in use, a replica could sometimes override the SQL mode value that was sent by the source, in an attempt to avoid issues with additional columns on the slave. In extreme cases this could lead to data divergence. The problem has been corrected so the replica now preserves the source's SQL mode wherever possible. (Bug #33945038)
- **Replication:** MySQL's semisynchronous replication did not respect the value of the `net_read_timeout` system variable and forced a read timeout of one millisecond. This could result in the function experiencing partial reads of acknowledgment messages and packets arriving out of order, while other connections in the MySQL system were functioning correctly. The value of the `net_read_timeout` system variable is now applied to connections for semisynchronous replication. (Bug #101056, Bug #31976209)
- **Replication:** When the `--replicate-same-server-id` option was used to make the replica not skip events that have its own server ID, if the log file was rotated, replication stopped with an error. The log rotation event now checks and applies the current value of the option. (Bug #89375, Bug #27492990)
- **Group Replication:** The `COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE` column in the Performance Schema table `replication_group_member_stats` could persistently show transactions related to view change events (`View_change_log_event`) that had already been applied. These events are queued in the Group Replication applier channel but applied in the Group Replication recovery channel, causing a race condition that could result in the counter decrement being lost. The increment of the count now takes place at a more suitable point, and the counter for `COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE` is also now set to zero when the applier is not busy. (Bug #33602354, Bug #33674059)
- **Group Replication:** A deadlock could occur in Group Replication when a member was interacting with the service infrastructure, such as a joining member checking for incompatible configuration and then leaving the group due to it. The issue has now been fixed. (Bug #32688091)
- **Group Replication:** The message logged when a member tries to rejoin a Group Replication topology when there is an old incarnation of the same server still present has been upgraded from an informational note to a warning message. (Bug #32651024)
- **API:** Applications that previously used the MySQL client library to perform an automatic reconnection to the server received the following mysql_query error after the server was upgraded:

[4031] The client was disconnected by the server because of inactivity. See wait_timeout and interactive_timeout for configuring this behavior. (Bug #105229, Bug #34007830)

- Pushing a condition down to a derived table was not handled correctly in all cases. (Bug #34311090)
- After pushing down a condition to a derived table having a set operation, while folding an always true boolean condition, the rewrite was not correct due to not setting `abort_on_null` to true for the cloned condition when making a copy during condition pushdown to a derived table with a set operation. (Bug #34298238)
- A missing error return when processing an invalid `ORDER BY` expression in a view definition led to an assert in debug builds. (Bug #34239456)
- MySQL Server would not compile with the latest version of Visual Studio 2022. (Bug #34231639)
- While attempting to clone a system variable during condition pushdown, the server sometimes could not determine the correct context of the cloned expression.

  To prevent this, we disallow condition pushdown to derived tables when they use system variables, or if the underlying expressions in the derived table contain system variables. (Bug #34205559)
- Added Enterprise Linux 9 (EL9) support. (Bug #34190004)
- On macOS 11, MySQL Server did not have the correct entitlement to generate a core dump in the event of an unexpected server halt. A build option WITH_DEVELOPER_ENTITLEMENTS has been added to allow a build to generate core dumps. (Bug #34163987)
- Improved error handling for '-DWITH_LIBEVENT=system' and '-DWITH_EDITLINE=system' on systems missing libevent-devel or libedit-devel. (Bug #34131334, Bug #34123545)
- The fix for Bug #33830493 in MySQL 8.0.29 addressed the situation where if a MySQL instance stopped unexpectedly or was restarted shortly after a `SET PERSIST` statement was used to record system variable settings, the configuration file `mysqld-auto.cnf` could be left empty, in which case the server restart could not proceed. The persisted system variables are now written to a backup file, which is only renamed to `mysqld-auto.cnf` after the success of the write has been verified, leaving the original `mysqld-auto.cnf` file still available. On a restart, if a backup file with valid contents is found, the server reads from that file. Otherwise the `mysqld-auto.cnf` file is used and the backup file is deleted. The file was not flushed to disk by this fix, so it was still possible for the issue to occur. This patch adds those operations. (Bug #34122866)
- Fixed the -DENABLE_GCOV CMake option. (Bug #34113243)
- The `SENSITIVE_VARIABLES_OBSERVER` privilege, introduced in MySQL 8.0.29, is now granted to users with the `SYSTEM_VARIABLES_ADMIN` privilege during upgrade. Previously, the privilege was not granted to any database user during upgrade. (Bug #34068378)
- A select from a view that used left joins did not return any results. (Bug #34060289)
- Under certain circumstances `TRUNCATE performance_schema.accounts` caused duplicated counts in `global_status`.

  This occurred if some hosts were not instrumented. For example, if `performance_schema_hosts_size` was set to a low value.

  Our thanks to Yuxiang Jiang and the Tencent team for the contribution. (Bug #34057013, Bug #106939)
- It was possible under certain conditions for `EXPLAIN ANALYZE` to attempt access of an iterator that did not exist. (Bug #34051681)

References: This issue is a regression of: Bug #33905399.

- Support was added for compiling the `keyring_oci` plugin with OpenSSL 3. (Bug #34043013)
- Corrected two issues found with `EXPLAIN ANALYZE`:
  - Timings displayed as actual time were not cumulative.
  - For a materialized table, the output now show the number of rows materialized, rather than the number of rows read from the materialized table.

  (Bug #34025798, Bug #34135465)

  References: See also: Bug #33834146, Bug #34678179.
- Events recorded in the Performance Schema tables for thread creation and deletion were retained until server shutdown, instead of being removed when the client connection ended. Thread creation and deletion now takes place after the Performance Schema instrumentation is created for the user session, so it is cleaned up when the session ends. (Bug #34019988)
- Upgraded the bundled zlib library to zlib 1.2.12. Also made zlib 1.2.12 the minimum zlib version supported, and removed WITH_ZLIB from the WITH_SYSTEM_LIBS CMake option. (Bug #34015600)
- The `CONNECTION_ID()` function, since it returns a session ID which remains constant for the lifetime of the session, was treated as a constant function. This caused issues when `CONNECTION_ID()` was used inside a trigger attached to a table which might be reused by other sessions. We fix this by making the function `const` for execution, and returning the actual session ID when the function is evaluated. (Bug #34009876)
- Executed `codespell` on the source code and fixed the reported spelling errors in the code comments. (Bug #34006439)
- The MySQL Enterprise Encryption `openssl_udf` function library plugin was reimplemented to use OpenSSL 3 APIs. (Bug #33992115)
- `FEDERATED` storage engine code was revised to address NULL pointer and variable access issues. (Bug #33962357)
- Histograms in MySQL returned a selectivity estimate of `0` for values that outside buckets. This meant that values might be missing from the histogram because they were missed during sampling, or because the histogram had grown stale. To prevent this, we introduce a constant lower bound of `0.001` on the selectivity estimates produced by histograms. This choice of lower bound corresponds to the selectivity of a value or range that we are likely to miss during sampling.

  Using a constant lower bound rather than a statistical estimate for the selectivity of a missing value has the advantage of simplicity and predictability, and provides some protection against underestimating the selectivity due to stale histograms and within-bucket heuristics.

  For more information about histograms in MySQL, see Optimizer Statistics. (Bug #33935417)

- For certain queries using a common table expression (CTE), `EXPLAIN ANALYZE` did not provide any profiling data for the CTE even when the CTE was known to be executed. This happened when the following conditions were met:

  - The CTE was referenced more than once in the query plan.
  - The first reference to the CTE (in the order of the output of `EXPLAIN FORMAT=TREE`) was never executed.
  - At least one of the subsequent references was executed at least once.

  The problem was that the CTE plan was always printed when encountering the first reference to the CTE; if that reference was never executed, the CTE was not materialized there; and thus there was no profiling data to print.

  The fix for this issue ensures that we print the CTE plan when it is first executed, that is, the point at which it is materialized. The output then includes profiling data. If the CTE is never executed, we print the plan at the last reference, when there is no profiling data. (Bug #33905399)

- The output from the command `mysqld --verbose --help` previously showed plugin load options as `ON` even when they were off by default, or turned off using an option. The output now shows the current value for the plugin. (Bug #33870892)

- The Server now bundles curl (7.83.1) and only uses it when alternative SSL systems are used, such as openssl11 on EL7. (Bug #33859507, Bug #34154806)

- Debug MySQL binaries can now be built using `-0g` and `-fno-inline`. (Bug #33855533)

- The `FIREWALL_EXEMPT` privilege, introduced in MySQL 8.0.27, is now granted to users with the `SYSTEM_USER` during upgrade. Previously, the privilege was not granted to any database user during upgrade. (Bug #33854409)

- A correlated subquery did not use a functional index as expected. This occurred when an outer column reference used inside the subquery was not considered as constant for subquery execution, which allowed consideration of the functional index to be skipped.

  We fix this problem by making sure to consider the outer column reference as constant while executing the subquery. (Bug #33851055)

- Added alternate OpenSSL system package support by passing in openssl11 on EL7 or openssl3 on EL8 to the WITH_SSL Cmake option. Authentication plugins, such as LDAP and Kerberos, are disabled as they do not support these alternative versions of OpenSSL. (Bug #33835934)

- Prepared statements with subqueries that accessed no tables, but the subquery evaluation raised an error, triggered an assert failure in debug builds. (Bug #33773799)

- Some stored functions were not executed correctly following the first invocation. (Bug #33754993)

- When performing a query using a recursive common table expression (CTE) with a removal of a query expression after constant predicate elimination, it is expected that when the reference count of table objects for the CTE temporary table reaches zero, it should be possible once again to recreate the table, but in certain cases one of the table references was not properly recorded as attached to the CTE. (Bug #33725503)

  References: See also: Bug #32962511.

- Added a missing error return to the parser. (Bug #33725502)

- A number of issues with pushdown of conditions making use of outer references, relating to work done in MySQL 8.0.22 to implement condition pushdown for materialized derived tables, have been identified and resolved. (Bug #33725403, Bug #33725500, Bug #33725508, Bug #33725534, Bug #33725545)

- The plan generated for a `SELECT` using a common table expression involves table materialization and an index scan on the materialized table. Because the `temptable` engine does not yet support all index scan methods, such queries might not always execute correctly.

  With other MySQL engines, the materialization access path has special handling when the access path is not considered basic; for `temptable`, an index scan was not considered basic, which led to undefined behavior.

  We fix this issue by considering the index scan access path basic, and thus avoiding use of any index scan access methods on `temptable` tables. (Bug #33700735)

- The `Data_free` column in the `INFORMATION_SCHEMA.FILES` table was not updated after adding a new data file to the `InnoDB` system tablespace. (Bug #33508534)

- If a plugin attempted to register a system variable with a name that duplicated that of an existing system variable, the existing static system variable might be overwritten, and uninstalling the plugin might leave pointers to the freed memory. The issues have now been fixed. (Bug #33451101)

- `SHOW TABLES` and `SELECT * FROM INFORMATION_SCHEMA.TABLES` did not return any results from the Performance Schema if the user had access privileges on individual Performance Schema tables, only. (Bug #33283709)

- Calling a function relating to the `data_masking` plugin without first installing the plugin led to an unplanned server shutdown. Functions relating to this plugin are initialized by calling init functions which in turn access the UDF metadata service, but this is valid only when the data masking plugin is installed. We fix this problem by adding a check to verify that the plugin is installed before initializing such functions, and to return an appropriate error message if the plugin providing them is not installed. (Bug #33234046)

- Under certain conditions, the server did not handle the expiration of `max_execution_time` or the execution of a `KILL` statement correctly. (Bug #33218625)

- `mysqlslap`, which uses multiple threads to connect to the server, could not run with a user account that used FIDO authentication. The issue has been fixed by an update to the FIDO library allowing the authentication to be performed on multiple threads. (Bug #33067183)

- If an incorrect value was set for the `binlog_checksum` system variable during a session, a `COM_BINLOG_DUMP` command made in the same session to request a binary log stream from a source failed. The server now validates the specified checksum value before starting the checksum algorithm setup process. (Bug #32442749)

- For slow query logging, the `Slow_queries` status variable was not implemented unless the slow query log was enabled, contrary to the documentation. (Bug #28268680, Bug #91496)

- New parameters (`examined_row_count`, `affected_row_count`, and `return_row_count`) were added to the audit plugin. Our thanks to J D for the contribution. (Bug #110628, Bug #35267239)

- A prepared statement could accept an empty string as a valid float value, which represents a regression from 8.0.27 behavior. This fix explicitly checks that the length of an interpreted string is non-empty and fully interpreted as a (float) number. In addition, new verification now ensures that:

  - All numeric values are supported with empty strings and strings that are all spaces.

  - Regular numeric values are supported, as well as numeric values with leading and trailing spaces.

(Bug #107399, Bug #34213338)

References: This issue is a regression of: Bug #32213576.

- Upgrading to MySQL 8.0.29 led to issues with existing spatial indexes (see Creating Spatial Indexes). The root cause of the problem was a change in how geographic area computations were performed by the included Boost library, which was upgraded to version 1.77.0 in MySQL 8.0.29. We fix this by ensuring that we accommodate the new method whenever such computations are performed. (Bug #107320, Bug #34184111)

  References: This issue is a regression of: Bug #33353637.

- When pushing a condition down to derived table for prepared statements, we clone a condition which also includes parameters when a derived table contains unions. When a statement needed to be reprepared during execution—for example, when the signedness of the value specified did not match that of the actual datatype—the parameter was not cloned correctly resulting in errors. This occurred because the value specified for the parameter was used to print the string for reparsing, instead of a literal `?` placeholder character.

  Now in such cases we set a flag `QT_NO_DATA_EXPANSION` for printing parameters for reparsing which, when enabled, causes the `?` placeholder to be printed, rather than the actual value. (Bug #107230, Bug #34148712)

- On MacOS, improved Boost library detection logic for Homebrew as a potentially incompatible system's Boost version could get used even with `-DWITH_BOOST` set. (Bug #107151, Bug #34121866)

  References: This issue is a regression of: Bug #33769505.

- On RHEL 7.x, fetching the CPU cache line size returned 0 on s390x RHEL 7.x which caused rpl_commit_order_queue and integrals_lockfree_queue to fail.

  Our thanks to Namrata Bhave for the contribution. (Bug #107081, Bug #34095278)

- When the `mysql` client was unable to reconnect to the server following an unexpected server halt, the process of building the completion hash allocated memory that was not freed. The reconnection operation now does not build the completion hash if the client fails to reconnect, and the memory concerned is freed if the client is disconnected. (Bug #106864, Bug #34019571)

- Added a cycle timer for the s390x architecture.

  Our thanks to Namrata Bhave for the contribution. (Bug #106824, Bug #33997819)

- In certain cases, incorrect results could result from execution of a semijoin with materialization, when the `WHERE` clause of the subquery contained an equality. In some cases, such as when one side of such an equality was an `IN` or `NOT IN` subquery, the equality was neither pushed down to the materialized subquery, nor evaluated as part of the semijoin. This also caused issues with some inner hash joins. (Bug #106710, Bug #106718, Bug #33952115, Bug #33957233)

  References: See also: Bug #84705, Bug #25466100.

- Comparator functions for queries like `(<date column> <non-date column>) IN ((val1, val2), (val3, val4), …)` could return the wrong results. (Bug #106567, Bug #33897969)

- Fixed an assert definition in SetOsLimitMaxOpenFiles; our thanks to hongyuan li for the contribution. (Bug #106555, Bug #33893197)

- Previously, it was assumed that, when the same non-nullable expression was used as both the first and second arguments to `LIKE`, the result was always true, and so could be optimized away. This assumption turns out not to be valid, due to the fact that `LIKE` treats the backslash (`\`) as an escape character, even when `ESCAPE` is not specified. This led to different results when the condition was used in the `SELECT` list as opposed to the `WHERE` clause. To fix the problem, we no longer perform this optimization with `LIKE`, with or without an `ESCAPE` clause. (Bug #106444, Bug #33852756)

- In some cases, when arguments other than global transaction IDs (such as column values) were passed to `GTID_SUBSET()`, the function returned values other than the expected `NULL`. (Bug #106298, Bug #33793942)

- A problem with evaluation of general quantified comparison predicates occurred when the left-hand side of the predicate was `NULL`. In such cases, the value of the subquery evaluation from the last current row is saved, so that it does not need re-evaluation, but the cached value (`result_for_null_param`) was not cleared between executions, so that the next execution could re-use the result from the previous execution. One consequence of this was that, when a subquery execution first caused zero rows to match from the subquery—which for an `ALL` predicate should return `TRUE`—a subsequent execution causing at least one row to match also returned `TRUE`, even though `FALSE` was expected.

  To solve this issue, we now make sure to clear `result_for_null_param` while cleaning up the subquery predicate following execution. (Bug #106164, Bug #33755139)

- Test cases executed with the `--async-client` option and shutdown commands caused `mysqltest` to halt unexpectedly. (Bug #105797, Bug #33643149)

- MySQL supports the use of equiheight histograms to improve selectivity estimates. Each bucket in an equiheight histogram for a column should contain roughly the same number of values (rows); keeping the buckets small helps minimize any error.

  When constructing an equiheight histogram, too many values were sometime placed in the same bucket, which could result in substantial errors in selectivity estimation. We fix this by introducing a new equiheight construction algorithm that guarantees low error, and adapts to the distribution of the data to make efficient use of its buckets. In addition, a new estimator for the number of distinct values in histogram buckets provides improved worst-case error guarantees.

  See The INFORMATION_SCHEMA COLUMN_STATISTICS Table, and Optimizer Statistics, for more information. (Bug #104789, Bug #33302021)

- Deprecation warnings returned to client programs were sent to `stdout` rather than `stderr`, which in the case of `mysqldump` could mean that the dump file no longer worked because the warnings were included in it. The issue has now been fixed and the warnings are sent to `stderr`. (Bug #104769, Bug #33733529)

- Extended support for chained SSL certificates. (Bug #54158, Bug #27491518)

# Changes in MySQL 8.0.29 (2022-04-26)

**Important**

This release is no longer available for download. It was removed due to a critical issue that could cause data in `InnoDB` tables having added columns to be interpreted incorrectly. Please upgrade to MySQL 8.0.30 instead.

- Authentication Notes

- Character Set Support

- Compilation Notes
- Deprecation and Removal Notes
- SQL Function and Operator Notes
- Optimizer Notes
- Performance Schema Notes
- Security Notes
- SQL Syntax Notes
- Test Suite Notes
- XA Transaction Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Authentication Notes

- The maximum size of FIDO authenticator data was increased. (Bug #33655192)

## Character Set Support

- **Important Note:** The server now uses `utf8mb3` rather than `utf8` in the following cases:

  - In the output of `SHOW` SQL statements (`SHOW CREATE TABLE`, `SHOW CREATE VIEW`, `SHOW CREATE DATABASE`)
  - When reporting invalid strings.

  (Bug #33385252, Bug #33395007)

- The server now uses `utf8mb3` in place of the alias `utf8` for character set names when populating data dictionary tables from built-in character sets. This affects the display of character set and related information in the MySQL Information Schema tables listed here:

  - `CHARACTER_SETS`
  - `COLLATIONS`
  - `COLUMNS`
  - `COLLATION_CHARACTER_SET_APPLICABILITY`
  - `PARAMETERS`
  - `ROUTINES`
  - `SCHEMATA`

  This change also affects the output of the SQL `SHOW CHARACTER SET`, `SHOW COLLATION`, `SHOW CREATE DATABASE`, and `SHOW CREATE TABLE` statements. (Bug #30624990)

## Compilation Notes

- **InnoDB:** After addressing the associated issues, the C4100, C4127, C4245 and C4389 MSVC++ level 4 compiler warnings were re-enabled. (Bug #33437498, Bug #33571677)
- GCC 11 is now a supported compiler for building MySQL on EL7 or EL8. This compiler is available in the `devtoolset-11` (EL7) or `gcc-toolset-11` (EL8) package. It is also recommended to use GCC 11 when building third-party applications that are based on the `libmysqlclient` C API library. (Bug #33730302)
- The server could not be compiled with Bison 3.8.1 or newer. (Bug #33488047)

## Deprecation and Removal Notes

- **Important Change:** Previously, MySQL allowed arbitrary delimiters and an arbitrary number of them in `TIME`, `DATE`, `DATETIME`, and `TIMESTAMP` literals, as well as an arbitrary number of whitespaces before, after, and between the date and time values in `DATETIME` and `TIMESTAMP` literals. This behavior is now deprecated, and you should expect it to be removed in a future version of MySQL. With this release, the use of any nonstandard or excess delimiter or whitespace characters now triggers a warning of the form `Delimiter 'char' in position pos in datetime value 'value' at row rownum is superfluous and is deprecated`, followed by `Please remove`, or in cases in which a suitable replacement can be suggested, `Prefer the standard 'replacementchar'`.

  A deprecation warning is returned only for the first nonstandard delimiter or whitespace character encountered in the literal value. An example is shown here:

```
mysql> SELECT DATE"2020/02/20";
+------------------+
| DATE"2020/02/20" |
+------------------+
| 2020-02-20       |
+------------------+
1 row in set, 1 warning (0.00 sec)

mysql> SHOW WARNINGS\G
*************************** 1. row ***************************
  Level: Warning
   Code: 4095
Message: Delimiter '/' in position 4 in datetime value '2020/02/20' at row 1 is
deprecated. Prefer the standard '-'.
1 row in set (0.00 sec)
```

  Such warnings are not elevated to errors in strict mode.

  For more information and examples, see String and Numeric Literals in Date and Time Context. (WL #13601)

- **Replication:** The `replica_parallel_type` system variable is now deprecated; whenever this variable is read or set, or the equivalent option is set in `my.cnf`, the server now issues a deprecation warning, as shown here:

```
mysql> SELECT @@replica_parallel_type; SHOW WARNINGS\G
+--------------------------+
| @@replica_parallel_type |
+--------------------------+
| LOGICAL_CLOCK           |
+--------------------------+
1 row in set, 1 warning (0.00 sec)

*************************** 1. row ***************************
```

```
  Level: Warning
   Code: 1287
Message: '@@replica_parallel_type' is deprecated and will be removed in a future
release.
1 row in set (0.00 sec)
```

You should expect this variable and the equivalent server option `--replica-parallel-type` to be removed in a future MySQL release.

- The `myisam_repair_threads` system variable and `myisamchk --parallel-recover` option are deprecated; expect support for both to be removed in a future release of MySQL.

  Values other than 1 (the default) for `myisam_repair_threads` produce a warning. (WL #14937)

- The server system variables `query_prealloc_size` and `transaction_prealloc_size` are now deprecated, and setting either or both of these no longer has any effect in the MySQL server. Expect them to be removed in a future MySQL release.

  For more information, see the descriptions of these variables in the documentation (Server System Variables). (WL #13720)

  References: See also: Bug #26940369.

- The `--abort-slave-event-count` and `--disconnect-slave-event-count` options for `mysqld`, previously used in testing, and not normally required in a production setting, have been deprecated. Expect them to be removed in a future version of MySQL. (WL #14526)

## SQL Function and Operator Notes

- Aggregate functions based on expressions comparing values with a `NULL` were not ignoring the `NULL` correctly. (Bug #33624777, Bug #105762)

- When an aggregation function is evaluated during optimization, it is presumed that the `WHERE` condition has already been evaluated; thus, it can be removed. Before the removal, the optimizer verifies whether the `WHERE` condition has any table-independent conditions by making a call to `make_cond_for_table()`. When a condition was considered expensive (for example, it used a stored procedure), it was incorrectly assumed that there was a table-independent condition. We fix this by excluding expensive conditions in the call to `make_cond_for_table()`.

  In addition, a constant condition which was expensive to evaluate, such as `f() = 1` where function `f()` did not use any tables, often led to incorrect results. In most cases, if a condition is constant for execution, it is evaluated when optimizing the `WHERE` condition, skipping it if it is considered too expensive. The present issue arose due to skipping the condition while optimizing an implicitly grouped query. To avoid this, we now evaluate such conditions prior to aggregation. (Bug #33305617)

- Columns used by the `DEFAULT()` function were not marked internally as `READ` when fixing its arguments. (Bug #33142135)

## Optimizer Notes

- If the column of a table being loaded using `LOAD DATA` was used in the `WHERE` clause of a subquery used inside a `SET` statement, the column was reported as `unknown`. (Bug #33714885)

- Aggregated queries that performed an index lookup (`eq_ref`) could in some cases return wrong results. This was seen when the execution plan used streaming aggregation instead of aggregation in temporary tables.

  We fix this by disabling `eq_ref` caching below aggregate nodes. (Bug #33491183)

References: This issue is a regression of: Bug #100614, Bug #31790217.

- The derived materialized table condition pushdown optimization can now be used with most unions. This means that an outer `WHERE` condition can now be pushed down to every query block of the query expression of the materialized derived table or view.

  Suppose we create tables `t1` and `t2`, and then the view `v` based on these two tables, using the SQL statements shown here:

```
CREATE TABLE t1 (
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  c1 INT,
  KEY i1 (c1)
);

CREATE TABLE t2 (
  id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  c1 INT,
  KEY i1 (c1)
);

CREATE OR REPLACE VIEW v AS
     SELECT id, c1 FROM t1
     UNION ALL
     SELECT id, c1 FROM t2;
```

  Now, when the query `SELECT * FROM v WHERE c1 = 12` is executed, the condition `c1 = 12` is pushed down to both query blocks of view `v`, as shown here in the output of `EXPLAIN`:

```
mysql> EXPLAIN FORMAT=TREE SELECT * FROM v WHERE c1 = 12\G
*************************** 1. row ***************************
EXPLAIN: -> Table scan on v  (cost=1.26..2.52 rows=2)
    -> Union materialize  (cost=2.16..3.42 rows=2)
        -> Covering index lookup on t1 using i1 (c1=12)  (cost=0.35 rows=1)
        -> Covering index lookup on t2 using i1 (c1=12)  (cost=0.35 rows=1)

1 row in set (0.00 sec)
```

  This can now be done for most `UNION` queries. For exceptions, and additional information, see Derived Condition Pushdown Optimization. (Bug #24012, Bug #36802, Bug #106006, Bug #11746156, Bug #11748590, Bug #13650627, Bug #30587347, Bug #33318096, Bug #33738597, WL #13730)

## Performance Schema Notes

- New Performance Schema instrumentation collects information about whether a query was processed on the `PRIMARY` or `SECONDARY` engine, where the `PRIMARY` engine is `InnoDB` and the `SECONDARY` engine is MySQL HeatWave. This instrumentation is intended for use with MySQL Database Service with MySQL HeatWave.

  Two new Performance Schema table columns were added:

  - An `EXECUTION_ENGINE` column that indicates whether a query was processed on the `PRIMARY` or `SECONDARY` engine was added to the Performance Schema statement event tables (see Performance Schema Statement Event Tables), and to the `performance_scema.threads` and `performance_scema.processlist` tables.

  - A `COUNT_SECONDARY` column that indicates the number of times a query was processed on the `SECONDARY` engine was added to the Performance Schema statement summary tables (see Statement Summary Tables).

An `execution_engine` column was also added to the `sys.processlist` and `sys.x$processlist` views in the MySQL `sys` Schema. (WL #14346)

## Security Notes

- For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.1.1n from 1.1.1l. Issues fixed in OpenSSL are described at https://www.openssl.org/news/cl111.txt and at http://www.openssl.org/news/vulnerabilities.html. (Bug #33840722, Bug #33970835)

## SQL Syntax Notes

- An `IF NOT EXISTS` option is now supported for the statements `CREATE FUNCTION`, `CREATE PROCEDURE`, and `CREATE TRIGGER`.

  For `CREATE FUNCTION`, when used to create a stored function, and for `CREATE PROCEDURE`, this option prevents an error from occurring if there is already a routine having the same name. For `CREATE FUNCTION`, when used to create a loadable function, the option prevents an error in the event there already exists a loadable function having that name. For `CREATE TRIGGER`, the option prevents an error from occurring if there is already a trigger with the same name, on the same table, and in the same schema. This is intended to improve ease of use for these statements in scripting, rapid (re)deployment, replication, and other instances of potential repeated use.

  This enhancement makes the syntax of these statements more consistent with that of `CREATE DATABASE`, `CREATE TABLE`, `CREATE USER`, and `CREATE EVENT`, all of which already support `IF NOT EXISTS`. This also provides a complement to the `IF EXISTS` option already supported by `DROP PROCEDURE`, `DROP FUNCTION`, and `DROP TRIGGER`.

  For more information, see Stored Objects, as well as Function Name Resolution. See also Replication of CREATE TABLE ... SELECT Statements. (Bug #15287, Bug #11745440, WL #14722)

  References: See also: Bug #33301931.

## Test Suite Notes

- Null pointers now are checked to reduce `heap-use-after-free` errors under Valgrind with the `--async-client` option. (Bug #33702755)

## XA Transaction Notes

- **Group Replication:** Group replication in some scenarios faced problems because it was not possible to commit an XA transaction prepared on another connection. To address such issues, MySQL now supports detached XA transactions; once prepared, an XA transaction is no longer connected to the current session. This happens as part of executing `XA PREPARE`. The prepared XA transaction can be committed or rolled back by another connection, and the current session can then initiate another XA or local transaction without waiting for the prepared XA transaction to complete.

  Support for this feature can be disabled, and the classic behavior reinstated, by setting the `xa_detach_on_prepare` system variable introduced in this release to `OFF`. The default is `ON`, which is recommended, especially in a replication setting.

  You should be aware that, when detached XA transaction support is enabled (`xa_detach_on_prepare = ON`), it is not possible to use temporary tables within XA transactions.

For more information, see XA Transaction States, as well as Server Instance Configuration. (Bug #100163, Bug #31599926, WL #14700)

## X Plugin Notes

- A difference in format between Unix socket lock files for classic MySQL protocol and X Protocol meant that the server could not start if the file for the other protocol was present. For the benefit of instances where it is not possible to remove the file manually, such as MySQL Database Service instances, the protocols now use the same format for the file. (Bug #31468581)

## Functionality Added or Changed

- **Incompatible Change:** MySQL Server now has the capability to store securely persisted system variable values containing sensitive data such as private keys or passwords, and to restrict viewing of the values. No MySQL Server system variables are currently marked as sensitive, but the new capability allows system variables containing sensitive data to be persisted securely in the future. A keyring component must be enabled on the MySQL Server instance to support secure storage for persisted system variable values, rather than a keyring plugin, which do not support the function.

  **Important**

  As part of this this change, component keyring files are now loaded prior to initializing the MySQL Server data directory. This means that it is no longer possible to keep such files in the MySQL data directory.

  In the operating system file where persisted system variables are stored, the names and values of sensitive system variables are stored in an encrypted format, along with a generated file key to decrypt them. The generated file key is in turn encrypted using a master key that is stored in a keyring. After upgrading to MySQL 8.0.29, the format of the `mysqld-auto.cnf` option file remains the same until the first time a `SET PERSIST` or `SET PERSIST ONLY` statement is issued, and at that point it is changed to a new format, even if the system variable involved is not sensitive. An option file using the new format cannot be read by older releases of MySQL Server.

  The new system variable `persist_sensitive_variables_in_plaintext` controls whether the server is permitted to store the values of sensitive system variables in an unencrypted format, if keyring component support is not available at the time when `SET PERSIST` is used to set the value. The default setting, `ON`, encrypts the values if keyring component support is available, and persists them unencrypted (with a warning) if it is not. This setting also allows the server to start if encrypted system variable values cannot be decrypted, in which case their defaults are used. When `persist_sensitive_variables_in_plaintext` is set to `OFF`, which is the most secure setting, sensitive system variable values cannot be persisted if keyring component support is unavailable, and the server does not start if encrypted system variable values cannot be decrypted.

  The values of sensitive system variables are also protected when they are handled. If a `SET` statement is issued for a sensitive system variable, the query is rewritten to replace its value with `redacted` before it is logged to the general log and audit log. The new privilege `SENSITIVE_VARIABLES_OBSERVER` allows a holder to view the values of sensitive system variables in the Performance Schema tables `global_variables`, `session_variables`, `variables_by_thread`, and `persisted_variables`, to issue `SELECT` statements to return their values, and to track changes to them in session trackers for connections. Users without this privilege cannot view those system variable values. (WL #13469)

- **InnoDB:** To improve code quality and facilitate debugging, instances of `#define` in the `InnoDB` sources were replaced by `constexpr` specifiers or inline functions. (WL #14680)

- **InnoDB:** `InnoDB` now supports `ALTER TABLE ... DROP COLUMN` operations using `ALGORITHM=INSTANT`.

  Operations that support `ALGORITHM=INSTANT` only modify metadata in the data dictionary. Table data is unaffected, making the operations instantaneous. If not specified explicitly, `ALGORITHM=INSTANT` is used by default by DDL operations that support it.

  Prior to MySQL 8.0.29, an instantly added column could only be added as the last column of the table. From MySQL 8.0.29, an instantly added column can be added to any position in the table.

  Instantly added or dropped columns create a new row version. Up to 64 row versions are permitted. A new `TOTAL_ROW_VERSIONS` column was added to the `INFORMATION_SCHEMA.INNODB_TABLES` table to track the number of row versions.

  For more information about DDL operations that support `ALGORITHM=INSTANT`, see Online DDL Operations. (WL #13899)

- **Replication:** Automatic purging of binary log files by the server is now controlled using the `binlog_expire_logs_auto_purge` system variable introduced in this release. By default, automatic purging is enabled (`binlog_expire_logs_auto_purge` set to `ON`); to disable it, set the value of this variable to `OFF`.

  The interval to wait before purging is controlled by `binlog_expire_logs_seconds` and `expire_logs_days`. Setting both of these system variables to 0 stops automatic purging from taking place, even when `binlog_expire_logs_auto_purge` is set to `ON`. (WL #14930)

- **Microsoft Windows:** Added jemalloc support on Windows, and enabled it for official MySQL binaries. The new `WITH_WIN_JEMALLOC` CMake option accepts a directory containing `jemalloc.dll` that's copied to the same directory as `mysqld.exe` and/or `mysqld-debug.exe` and utilizes it for memory management operations. Standard memory functions are used if `jemalloc.dll` is not found or does not export the required functions. An INFORMATION level log message records whether or not jemalloc is found and used. (WL #14633)

- The new `clone_delay_after_data_drop` variable permits specifying a delay period immediately after removing existing data on the recipient MySQL Server instance at the start of a remote cloning operation. The delay is intended to provide enough time for the file system on the recipient host to free space before data is cloned from the donor MySQL Server instance. Certain file systems free space asynchronously in a background process. On these file systems, cloning data too soon after dropping existing data can result in clone operation failures due to insufficient space. The maximum delay period is 3600 seconds (1 hour). The default setting is 0 (no delay). (Bug #32826134, WL #14857)

- The `group_replication_set_as_primary` function appoints a specified group member as the new primary and overrides the election process. Previously, the function waited for all active transactions on the existing primary to end, including incoming transactions after the function was used, before making the current primary read only and changing to the new primary. There was no upper limit to the wait time.

  An optional `timeout` parameter is now available to let you set a timeout for running transactions. You can set a timeout from 0 seconds (immediately) up to 3600 seconds (60 minutes) for transactions that are running when you use the function. There is no default setting for the timeout, so if you do not set it the former behavior of the function still applies, with no upper limit to the wait time. When the timeout expires, for any transactions that did not yet reach their commit phase, the client session is disconnected so that the transaction does not proceed. Transactions that reached their commit phase are allowed to complete. When you set a timeout, it also prevents new transactions starting on the primary from that point on. Explicitly defined transactions (with a `START TRANSACTION` or `BEGIN` statement) are subject to the timeout, disconnection, and incoming transaction blocking even if they do not modify any data. To

allow inspection of the primary while the function is operating, single statements that do not modify data, as listed in Permitted Queries Under Consistency Rules, are permitted to proceed. (WL #14585)

- The `keyring_okv` keyring plugin, which uses the Key Management Interoperability Protocol (KMIP) to communicate securely, is used with MySQL's keyring service to store keyring data in a KMIP-compatible back end keyring storage product such as Oracle Key Vault, Gemalto SafeNet KeySecure Appliance, Townsend Alliance Key Manager, and Entrust KeyControl. The plugin now allows you to specify more than one standby server to provide backup connections to the back end keyring storage product if the primary server is unavailable. You can add up to 64 standby servers by editing the `okvclient.ora` file to specify the IP addresses and port numbers of the servers as a comma-separated list in the value of the `STANDBY_SERVER` variable. The plugin iterates over the standby servers until it can establish a connection, with a 20-second wait for each connection attempt. You should therefore ensure that the list of standby servers is kept short, accurate, and up to date, and servers that are no longer valid are removed, as they can significantly affect the `keyring_okv` plugin's connection time and therefore the server startup time. (WL #14363)

- Isolation of access to system variables is now better enforced in the server code that parses, resolves, and executes SQL statements accessing or updating such variables. (WL #14529)

- MySQL Server now supports SSL session reuse by default with a timeout setting to control how long the server maintains a session cache that establishes the period during which a client is permitted to request session reuse for new connections. All MySQL client programs support session reuse. For server-side and client-side configuration information, see Reusing SSL Sessions.

  In addition, C applications now can use the C API capabilities to enable session reuse for encrypted connections (see SSL Session Reuse). (WL #13075)

## Bugs Fixed

- **InnoDB:** A failure occurred when attempting to purge undo records for a table with an instantly added column. (Bug #33924532)

- **InnoDB:** High-priority transactions were not permitted to stop waiting when interrupted or to timeout while waiting for a lock, preventing deadlocks from being resolved. In cases where the blocking transaction is also high-priority, high-priority transactions are now permitted to stop waiting when interrupted or timeout when exceeding the lock wait timeout period. If a blocking transaction is not high-priority, high-priority transactions wait for the blocking transaction to release its locks. (Bug #33856332)

- **InnoDB:** The AIO synchronization queue used on Windows was removed. The synchronous file I/O read-write function (`SyncFileIO::execute`) was revised to handle files opened for both normal and overlapped I/O, as it does on Linux. (Bug #33840645)

- **InnoDB:** Table version metadata was not reset after truncating all partitions of a table with an instantly added column. (Bug #33822729)

- **InnoDB:** The `srv_error_monitor_thread()` function, which prints warnings about semaphore waits, failed to handle a long semaphore wait as expected. To address this issue, a blocking call was moved to a more appropriate location. Related monitor thread code was simplified and improved, and missing shutdown signals were added for several server threads.

  Enabling and disabling of the standard monitor by InnoDB is now performed independently of the user-settable `innodb_status_output` variable. This change addresses an issue in which the monitor was enabled by `InnoDB` in a particular scenario but not set back to its previous value. Thanks to Yuhui Wang for the contribution. (Bug #33789526, Bug #93878)

- **InnoDB:** Valgrind testing identified an off-by-one error in `rec_convert_dtuple_to_rec_old()` in the InnoDB sources. (Bug #33784672)

- **InnoDB:** The `UNIV_DEBUG` variant of the `mem_heap_alloc()` function in the `InnoDB` sources was modified to improve Valgrind error detection. (Bug #33783709)
- **InnoDB:** A fast shutdown did not wait for all active I/O operations to finish before closing all files. (Bug #33768584)
- **InnoDB:** A Clang warning reported an incorrectly placed `@return` command. (Bug #33734011)
- **InnoDB:** Values of the new record locks array (`m_prebuilt->new_rec_locks[]`) were not properly synchronized when switching between partitions, causing an assertion failure due to locks being freed or not freed as expected. (Bug #33724166)
- **InnoDB:** A race condition in the function that updates the double write buffer when a write request is completed caused a long semaphore wait error. (Bug #33712370)
- **InnoDB:** A function wrongly assumed that changing a record in an indexed column always requires creating a new record in the secondary index, resulting in an lock-related assertion failure. To address this and other similar cases, the `lock_rec_convert_impl_to_expl()` function that converts an implicit record lock to an explicit record lock is now used only when an implicit record lock is actually held. (Bug #33657235)
- **InnoDB:** A number of Doxygen issues in the `InnoDB` sources were addressed. (Bug #33603036)
- **InnoDB:** A missing null pointer check for an index instance caused a failure. (Bug #33600109)
- **InnoDB:** Starting the server after an incremental backup failed with an error indicating that page tracking was unusable. (Bug #33521528)
- **InnoDB:** An assertion failure was raised during `InnoDB` recovery. The failure was due to a server exit that occurred after pages were freed in a local mini-transaction before the free list and index list could be reset. The free list and index list, which spanned the freed pages, were traversed during recovery. (Bug #33454557)
- **InnoDB:** The `btr_insert_into_right_sibling()` B-tree function, which is inserts a tuple into the right sibling page when the cursor is at the end of a page, caused an ACID violation due to an incorrect lock function call. (Bug #33405696)
- **InnoDB:** When using `COMPACT` or `REDUNDANT` row format, it was possible to create a table that exceeded the maximum row size, which could eventually result in 'Row size too large' errors when inserting data. BLOB prefixes were not included in the record size check performed during the `CREATE TABLE` operation. (Bug #33399379)
- **InnoDB:** A function that retrieves the master encryption key (`get_master_key()`) tried to acquire the master key ID mutex (`master_key_id_mutex`), which was not yet initialized. (Bug #33067891)
- **InnoDB:** A redundant and costly check was removed from the `lock_sec_rec_read_check_and_lock` lock system function, which is used to place a lock on a secondary index record. (Bug #33059387)
- **InnoDB:** The `TempTable` storage engine did not properly handle a file-full error. (Bug #32929392)
- **InnoDB:** A DDL log error encountered during an `ALTER TABLE ... ALGORITHM=COPY` operation was not handled, causing a failure during a subsequent `ALTER TABLE ... ALGORITHM=INPLACE` operation. (Bug #32716838)
- **InnoDB:** Purge threads processed undo records of an encrypted table for which the tablespace was not loaded, causing a failure. (Bug #32586721)

- **InnoDB:** A thread posted an asynchronous I/O operation and closed itself, resulting in an operating system file operation error. (Bug #30567295)

- **InnoDB:** Incorrect `AUTO_INCREMENT` values were generated when the maximum integer column value was exceeded. The error was due to the maximum column value not being considered. The previous valid `AUTO_INCREMENT` value should have been returned in this case, causing a duplicate key error. (Bug #87926, Bug #26906787)

- **InnoDB:** Transaction lock priority was modified so that a transaction holding a shared lock and waiting to upgrade to an exclusive lock is prioritized ahead of a another transaction waiting for an exclusive lock on the same row. In other words, the transaction that already holds a shared lock is granted an exclusive lock first. Some other transaction holding a shared lock on the same row can still prevent an exclusive lock from being a granted in this scenario.

  The `lock_rec_find_set_bit` and `lock_rec_has_expl` functions in the transaction lock system sources were optimized. (Bug #21356, Bug #11745929)

- **Partitioning:** In some cases, establishing a connection to MySQL server could fail if the `.ibd` file for a partition was missing. (Bug #33459653)

- **Partitioning:** Following work done in MySQL 8.0.17 to cause all `NOT IN` and `NOT EXISTS` subqueries to be transformed into antijoins, the inner part of a nested outer join produced by this transformation was not considered while pruning table partitions. (Bug #33060953)

- **Replication:** In a replication group with `group_replication_consistency=AFTER` set, if a new primary was elected and then the previous primary was restarted, the new primary left the group with an error message. The handling of delayed view changes and transaction prepares has been adjusted to prevent this situation. (Bug #33755920)

- **Replication:** A deadlock caused when the network provider was being stopped could cause `STOP GROUP_REPLICATION` to wait indefinitely on some group members. (Bug #33044886)

- **Group Replication:** Group Replication checked too late in the auto-rejoin process that a rejoining member was the only type of instance allowed to enter a group while in an error state. The issue has now been fixed. (Bug #33615493)

- **Group Replication:** Group Replication could log a warning message about a member joining while a group configuration operation was running, in situations where there was no joining member, just a view change. The message is now only logged when appropriate. (Bug #33378364)

- **Group Replication:** Group Replication now logs operating system errors returned when there is a problem connecting to the local XCom instance, so it is easier to resolve issues such as a firewall blocking the connection. (Bug #33189767, Bug #104523)

- **Group Replication:** To help prevent a primary election from hanging, Group Replication stops waiting for a confirmation from any members that leave the group during the election. If all the members for which confirmation was pending leave, Group Replication declares that the primary election completed successfully. Previously, if the primary member was one of the members that left, a primary change was incorrectly stated as having happened when that had not been able to take place. Now, if the primary member leaves during an election, Group Replication declares the primary election as an unsuccessful election with no primary change. (Bug #33059773)

- **Microsoft Windows:** On Windows, added WinFlexBison support. (Bug #33788290)

- On EL6/EL7, a mysql-community-server-debug installation was not obsoleted by installing commercial packages. (Bug #33956760)

- MySQL Server would not build EL7 on ARM when building against shared libraries that are copied by the cmake build system, such as a non-system OpenSSL. Now *--page-size* is set as reported by getconf. (Bug #33904267)

- An anonymous user with the `PROCESS` privilege could not select rows from the `processlist` table. (Bug #33869388)

- The range optimizer did not take into account the session memory limit. (Bug #33869004)

- When sorting subqueries, the same filesort operation may be executed many times, with different input, and so the longest add-on column may change between executions, but the buffer for this column was not reallocated when necessary. Now, when the previously allocated buffer is sufficiently large, we reuse it, otherwise we allocate a new one. (Bug #33865094)

- An invalid result was obtained from a common table expression that was materialized and used in multiple query blocks, where, in the first query block that used the CTE, there were multiple possible key definitions, which caused the internal function `JOIN::finalize_derived_keys()` to move the definition of the key used into position zero, and the second query block using the CTE manipulated an index that occupied the same position as the original position for the key chosen for the first query block. This happened as an unintended side effect of work done in MySQL 8.0.22 to prepare DML statements once only.

  For more information, see WITH (Common Table Expressions). (Bug #33856374)

- Improved error handling when executing stored procedures and stored functions. (Bug #33851256)

- A trigger that selected from a view did not always yield the expected result. (Bug #33847722)

- Computations performed while loading time zone information require negative values of `my_time_t`. Normally, that type's minimum value is zero, corresponding to the start of the UNIX Epoch (`1970-01-01 00:00:00`, with no time zone offset), which is reflected in the constant `MYTIME_MIN_VALUE`. This minimum value did not work correctly for setting up the `TIME_ZONE_INFO` data structures in all cases, since some time zone offsets are negative. This fix permits negative values for `my_time_t` while populating the time zone information database cache. (Bug #33837691)

- If a MySQL instance stopped unexpectedly or was restarted shortly after a `SET PERSIST` statement was used to record system variable settings, the configuration file `mysqld-auto.cnf` could be left empty, in which case the server restart could not proceed. The persisted system variables are now written to a backup file, which is only renamed to `mysqld-auto.cnf` after the success of the write has been verified, leaving the original `mysqld-auto.cnf` file still available. On a restart, if a backup file with valid contents is found, the server reads from that file. Otherwise the `mysqld-auto.cnf` file is used and the backup file is deleted. (Bug #33830493)

- `filesort` did not always check the lengths of string values properly when calculating sort keys. (Bug #33830073)

  References: This issue is a regression of: Bug #29739778.

- A primary key comprising the `GROUP_ID` and `MEMBER_ID` columns was added to the `mysql.firewall_membership` table. (Bug #33824544)

- A fix made in MySQL 8.0.20 for an inconsistency in the `rows matched` value when updating an updatable view having `WITH CHECK OPTION` handled this issue for an update of a single table only. In this release, we fix the issue for multi-table `UPDATE` statements as well. (Bug #33815426)

  References: See also: Bug #30158954.

- The linked ProtoBuf library for MySQL Server has been updated to version 3.19.4. Issues fixed in the new Protobuf version are described at https://github.com/protocolbuffers/protobuf/releases. (Bug #33813846)

- It was possible to insert invalid characters into a `utf32` column. (Bug #33810558)

- Reinstalling the MySQL Enterprise Firewall plugin caused a failure. The failure was due to a call to update a previously persisted firewall system variable before the plugin was fully initialized. (Bug #33809478)

- Hash joins using `VARCHAR` or `DECIMAL` join columns returned an excessive number of rows. An example using `VARCHAR` is shown here:

  ```
  # Create and populate table
  CREATE TABLE t(a VARCHAR(10), b VARCHAR(10));
  INSERT INTO t VALUES ('x', 'xx'), ('xx', 'x');

  # No indexes, so uses a hash join
  SELECT * FROM t AS t1, t AS t2 WHERE t1.a = t2.a AND t1.b = t2.b;
  ```

  Prior to the fix, the query just shown returned four rows; now it returns two rows as expected. (Bug #33794977)

- The `INFORMATION_SCHEMA.KEY_COLUMN_USAGE` table did not list invisible key columns. (Bug #33781534)

- The server did not perform proper cleanup after accessing a view created from a view. (Bug #33777821)

- In a grouped query using `WITH ROLLUP` and a window function which ordered on an aggregate already present in the `SELECT` list, an extra (and unnecessary) aggregate was added as a hidden column when resolving the window's ordering clause. (Bug #33769911)

- The data masking `mask_ssn()` function returned a `'String argument width too large'` error for a valid non-constant value (data read from a database row as opposed to a hard-coded value). Additionally, validation was not performed for zero-length constant and non-constant values (empty strings) or for constant values that are too small. These issues were addressed for all data masking functions that string inputs with length boundaries. (Bug #33759276)

- `ROLLUP` was not handled correctly in some cases with `ORDER BY` and a window function. (Bug #33753245)

- Corrected a number of issues in Doxygen comments. (Bug #33734001, Bug #33734035, Bug #33734062, Bug #33734075, Bug #33734104, Bug #33734117, Bug #33734129, Bug #33734143, Bug #33734155, Bug #33734181, Bug #33734188, Bug #33734206)

- Statements that cannot be parsed (due, for example, to syntax errors) are no longer written to the slow query log. (Bug #33732907)

- The message describing MySQL Enterprise Thread Pool's initial configuration is now issued as a SYSTEM message rather than informational, so that it is visible at the default logging level. (Bug #33729821)

- A pre-locking optimization related to session tracking caused recursive locking in some scenarios. (Bug #33728209)

- A union involving an `IN` subquery was not always handled properly when it also used an `ORDER BY` clause. (Bug #33725507)

- The `mysqlpump` client utility could stop unexpectedly when a password prompt was requested using command line options. If an error is encountered during this process an appropriate error message is now returned before the client exits. (Bug #33688141)

- Implemented standard package policy to ship separate debuginfo RPMs for the following platforms: SLES/openSUSE, and community EL6 and EL7. Commercial versions of EL6 and EL7 are now built with the RelWithDebInfo build type instead of Debug, which greatly decreases their size. (Bug #33663811, Bug #33684418, Bug #33664929)

- It was not possible to run `mysqld` with the trace log enabled. (Bug #33653824)

- When the `--fido-register-factor` option was used at startup for FIDO device registration, the `mysql` client could use incorrect syntax in the `ALTER USER` statement, resulting in a failure. The issue has now been fixed. (Bug #33650498)

- When searching a `SET` column value, a user variable passed as the first argument to `FIND_IN_SET()` did not produce the same result as a constant value used in the same manner.

  *Example*: Consider the table `t1` created and populated as shown here:

  ```
  CREATE TABLE t1 (c1 SET('a', 'b', 'c', 'd'));

  INSERT INTO t1 (c1) VALUES ('a, c, d'), ('c');
  ```

  The values shown for `var` and `str` should be the same in both rows of the output from the following query, but were not:

  ```
  SET @a = 'c';

  SELECT FIND_IN_SET(@a, c1) AS var, FIND_IN_SET('c', c1) AS str FROM t1;
  ```

  Now we make sure in such cases to return the position of the match within the set used for the column definition whether the value sought is a constant or a column value. (Bug #33635637)

- A `SET PASSWORD` operation using an authentication method that does support it produced a warning instead of an expected error. (Bug #33635445)

- If the plugin directory was not specified at MySQL Server startup, and a user attempted FIDO device registration, the `mysql` client stopped unexpectedly. The situation is now handled and an appropriate error is reported. (Bug #33631144)

- A change made in MySQL 8.0.27 moved the `EXPLAIN` output for composite access paths including filter and sort from inside the table path to the `MATERIALIZE` access path, but did not include `INDEX_RANGE_SCAN` among the composite access paths. This could in some cases result in undefined behavior. (Bug #33611545)

  References: This issue is a regression of: Bug #32788576.

- It was not possible to revoke the `DROP` privilege on the Performance Schema. (Bug #33578113)

- The internal `CreateIteratorFromAccessPath()` function has been rewritten so that it is no longer employs recursion; this is expected to reduce stack usage significantly when executing queries. (Bug #33569770)

- The `mysql` client could stop unexpectedly when attempting FIDO device registration with a challenge-response error. The issue has now been fixed. (Bug #33568944)

- An `INFORMATION_SCHEMA.FILES` or `INFORMATION_SCHEMA.TABLES` query executed by prepared statement or stored function returned stale data. Cached data was reported after the data had changed. (Bug #33538106, Bug #105450)

- A query returned an incorrect result when it met all of the following conditions:

  - The result set contained a column with a nondeterministic element.

  - The column did not refer to a table.

  - There was implicit grouping.

  - The query plan used aggregation from a temporary table. This temporary table was empty.

  Given an empty table `t` with column `c1`, the statement `INSERT INTO t SELECT MAX(t.c1), RAND(0) AS x FROM t` provides an example of such a query; by inserting into the same table, we ensure that the query plan aggregates from a temporary table. The temporary table has a column for `c1`, and the (single) result row picks `c1` from the first row of the temporary table. If the temporary table was empty, there was no first row, and `c1` became 0; this led to the insertion of `(NULL, 0)` rather than, for example, `(NULL, 0.155)` into `t`.

  We fix this by not storing such columns in the temporary table. (Bug #33535379)

  References: See also: Bug #32384355.

- Following changes in password handling in MySQL 8.0.27, if the `mysql` client was interrupted with **Ctrl + C**, a password prompt could be issued requiring the user's password in order to action the interrupt. The password is now recorded when it is provided in response to a prompt (as well as when it is provided on the command line at startup), so it can be used automatically by the new connection that is established for the interrupt. (Bug #33514253)

- The internal `keyread` flag indicates reading only index entries and not full rows. When this flag was set for an index merge scan during initialization, it was subsequently reset when initializing the individual range scans picked as part of the index merge scan, which indicated reading rows instead of index entries. We fix this by setting the indexes chosen as covering indexes while reading the index entries. (Bug #33499071)

- Data copying stalled when cloning page-compressed tables where the compressed data size on the donor host was equivalent to the available disk space on the recipient host. The holes punched on the recipient host were smaller than the page size, resulting in disk fragmentation that limited the amount of space available for cloned data. The process for handling page-compressed tables on the recipient host was modified to reduce fragmentation and improve data copying speed.

  To improve the rate of flushing, the clone plugin now uses the `fsync` flush method when flushing page-compressed tables on file systems that support hole punching. (Bug #33482747)

- It was possible, when creating a view, for a subquery to be evaluated during resolution when it was deemed constant. (Bug #33438883)

- Nominally, in `EXPLAIN FORMAT=TREE` output, we write a number of rows only as an integer. For a result of zero, it was difficult to determine whether this was 0.49 or 0.00001, so we add enough precision to get one leading digit in such cases. (Bug #33426283)

- `VALUES` in some instances did not function as specified. (Bug #33414289)

- In debug builds, certain combinations of user variables of type `MYSQL_TYPE_DOUBLE` hit an assert when used in the context of `CAST()` or `CONVERT()`. (Bug #33406728)

- When setting up multiple equalities, the optimizer added an outer reference when it was not required. (Bug #33394701)

- The minimum supported version of the Boost library for MySQL builds has now been raised to 1.77.0. (Bug #33353637)

  References: See also: Bug #33052171.

- Redundant `init` and `checkout_access_maps` function code was removed from the `Security_context` class. (Bug #33339129, Bug #104915)

- The sys schema view `schema_unused_indexes` did not update correctly.

  `schema_unused_indexes` retrieves unused index information from `table_io_waits_summary_by_index_usage`, returning no data if the index was used to query a row which is not present in the table.

  As of this release, `schema_unused_indexes` is updated even if the row is not found. (Bug #33302400)

- Some queries using common table expressions were not handled correctly by `EXPLAIN`. (Bug #33300271)

- The server did not obtain index statistics correctly for certain complex queries. (Bug #33270516)

- Under certain circumstances, when executing `TRUNCATE TABLE performance_schema.status_by_thread;` a race condition could occur.

  Our thanks to Facebook for their contribution to this fix. (Bug #33165726)

- A page cleaner thread timed out as it waited for an exclusive lock on an index page held by a full-text index creation operation on a large table. (Bug #33101844)

- For an `InnoDB` table, a secondary index is extended to include the table's primary key; that is, all columns that are part of the primary key are marked as being part of all secondary indexes. An exception to this is when a secondary index is defined as unique, server does not use the primary key extension. This could lead to problems while setting up the read sets for dynamic range scan. We fix this issue by making sure in the case of a unique seconary index to include all primary key parts in the read set. (Bug #33101025)

  References: This issue is a regression of: Bug #30417361.

- An existing foreign key constraint that referenced a non-existing table could result in a connection error after the table was created or renamed. (Bug #33054071)

- While creating a value list for an empty `VALUES` during execution of an `INSERT` statement, a hidden element was added, which subsequently led to an assertion in debug builds. To fix the problem, we now skip any hidden elements when the `VALUES` list is created.

  See INSERT ... ON DUPLICATE KEY UPDATE Statement, for more information. (Bug #32774799)

- Aliases to `SELECT` list items in `ORDER BY` which were part of expressions were not resolved in the same way as standalone aliases, so that `ORDER BY alias` was resolved correctly, but `ORDER BY function(alias)` was not. (Bug #32695670)

- An offline keyring migration from an empty or non-existing keyring source (specified by `keyring_file_data`) appeared to complete successfully, without warning. The keyring migration process now checks for errors when fetching keys. If no keys are found at the specified keyring source, a warning message is written to the error log indicating that the keyring has no keys. (Bug #32327411)

- When removing a condition which was always true, a subquery referring to a view was removed even though the subquery was itself referred to elsewhere in the query. (Bug #31946448)

  References: See also: Bug #31216115.

- An `EXPLAIN` statement, if used with multiple-table `UPDATE` statements that contained derived tables, could cause the server to exit. (Bug #31884434)

- A memory leak occurred if `mysqldump` was used on more than one table with the `--order-by-primary` option. The memory allocated for sorting each table's rows is now freed after every table, rather than only once. (Bug #30042589, Bug #96178)

- `mysqld_safe` log message textual errors were corrected. Thanks to Bin Wang at China Mobile for the contribution. (Bug #106590, Bug #33903639)

- Equivalent queries could return different results because the execution paths differed slightly. For example, `Item::val_int_from_string()` and `Item_string::val_int()` should use the same algorithm for string to integer conversion. (Bug #106517, Bug #33881339)

- Updated CMake rules to handle the deprecated `PermissionsStartOnly` systemd option. The alternative executable prefix was added in systemd 231 (July 2016) while PermissionsStartOnly was deprecated in systemd 240 (Dec 2018). The preferred executable prefix is now used where available. (Bug #106468, Bug #33862323)

- Following work done in MySQL 8.0.22 to implement condition pushdown for materialized derived tables, an implicitly grouped query with an aggregation function in some cases returned an empty set instead of `NULL`. (Bug #106414, Bug #33838439)

- A prepared and executed query like `'SELECT ?'` with a `DATE`, `TIME`, `DATETIME`, or `TIMESTAMP` as the bound argument could return the wrong column type. The issue is fixed by restoring the previous logic that reprepared such queries to give each dynamic parameter its literal value. (Bug #106352, Bug #33813951)

  References: This issue is a regression of: Bug #32915973.

- A null-safe comparison (`<=>`) did not evaluate correctly for a `TIMESTAMP` column in a trigger. (Bug #106286, Bug #33790919)

  References: This issue is a regression of: Bug #32942327.

- When writing `DECIMAL` values to the hash join buffer, the values were first normalized. This normalization could in some cases unintentionally alter the values, which made the values fall into the wrong hash bucket and caused incorrect results to be returned from the join.

  The problem was that the normalization removed leading zeros in a manner which caused some values to lose some of their least significant digits. We fix this by calculating the desired precision and passing the calculated value instead of changing the original value. (Bug #106272, Bug #33787914)

- A query having a `WHERE` condition of the form `c NOT BETWEEN NULL AND COALESCE(a, b)`, having `DATETIME` columns `a`, `b`, and `c`, with columns `a` and `c` created as `NOT NULL`, did not return correct results. Further investigation showed that, in such cases, when the second or third argument to `BETWEEN` was `NULL`, the comparison results were inverted, causing it to evaluate as `NULL` for `FALSE`, and `FALSE` for `NULL`. (Bug #106267, Bug #33788832)

- A NULL-safe comparison between `0` (`FALSE`) and `NULL` could return `TRUE`, even though the implementation responsible for evaluating an object of this type can never result in `NULL` and should always return `FALSE`. (Bug #105773, Bug #33630225)

- The `QUICK_RANGE` constructor used the main thread memory root for allocating keys, rather than the same memory root as used for allocating the `QUICK_RANGE` object itself. This resulted in unexpectedly high memory usage when executing multiple queries using a range check for the query plan. (Bug #105331, Bug #33516707)

  References: See also: Bug #28805105, Bug #28857990.

- `SELECT DISTINCT` on a `VARCHAR` column returned 2 distinct values when the column contained an empty string and a string consisting of a single whitespace character when the TempTable storage engine was used (`internal_tmp_mem_storage_engine = TempTable`). This behavior was incorrect when using a `utf8_bin` collation with `PAD SPACE`; trailing spaces should have been ignored when selecting distinct values.

  For temporary tables created using the `TempTable` storage engine, the two strings were hashed to different values, since the `TempTable` hashing function always hashed zero-length data to `0`. (In the case just described, the empty string was hashed to `0` while the space character from the other string was used to generate a nonzero hash value.)

  This is now fixed by removing any special handling for zero-length data in the `TempTable` storage engine's hashing function.

  Our thanks to Brian Yue for the contribution. (Bug #105316, Bug #33506241)

- `SELECT COUNT(*)` using a multi-valued index reported the wrong number of rows. (Bug #104898, Bug #33334928)

- The test program `main.ssl-big` failed with `Too many connections`.

  Our thanks to Facebook for the contribution. (Bug #104377, Bug #33139204, Bug #33165693)

- The two types of histograms supported by MySQL, singleton and equi-height, previously stored their collections of buckets in a map and a set, respectively. This release changes both histogram types such that they now store buckets in dynamic arrays instead, which reduces space overhead and speeds up selectivity estimation due to reduced indirection when performing binary search on the buckets.

  Because both types of histograms are constructed by inserting buckets in sorted order, and buckets are serialized to JSON in sorted order, there is no additional work involved in switching to a dynamic array. (Bug #104109, Bug #33045204)

- An invalid conversion between signed and unsigned variables led to incorrect result from a query of the form `SELECT ALL t1.c1,t2.c1 FROM t1,t2 WHERE t1.c1 > t2.c1`. (Bug #102025, Bug #32302724)

- As of this release, the Performance Schema is collation-aware (Bug #98545, Bug #30881109)

- An assert error could occur when inserting data larger than 32 bytes in the `ROLE` column of `setup_actors`. As of this release, the `ROLE` column is increased from CHAR(32) to CHAR(96). (Bug #74678, Bug #19947842)

# Changes in MySQL 8.0.28 (2022-01-18)

- Audit Log Notes
- C API Notes
- Character Set Support
- Compilation Notes

- Data Type Notes
- Deprecation and Removal Notes
- Event Scheduler Notes
- Full-Text Search Notes
- SQL Function and Operator Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Functionality Added or Changed
- Bugs Fixed


## Audit Log Notes

- The new `audit_log_disable` system variable permits disabling audit logging for all connecting and connected sessions. See Disabling Audit Logging. (WL #14699)

## C API Notes

- Building C API client programs on Windows using Clang as the compiler returned various compiler warnings, including an unused variable warning. (Bug #33520805, Bug #33520861)

## Character Set Support

- **Incompatible Change:** `CONVERT(`*`string`*` USING `*`charset`*`)` did not compute the correct maximum length for its return value, which should be the same as that calculated for `CAST(`*`string`*` AS `*`charset`*`)`. This meant that some conversions of strings from `BINARY` to nonbinary character sets which should have been rejected as invalid returned values instead.

  Prior to upgrading, applications that may rely on the previous `CONVERT()` behavior should be checked and updated as necessary. In particular, for indexes on generated columns using `CONVERT()` with invalid values, you should correct such values, drop the index, then re-create it before upgrading to this release. In some cases, it may be simpler to rebuild the table using `ALTER TABLE `*`table`*` FORCE`, rather than dropping and re-creating the index. See SQL Changes, for more information. (Bug #33199145)

- The output from `EXPLAIN FORMAT=TREE` hex-encoded ranges for multi-valued indexes, even when the data type was not a binary one. In addition, ranges using string types were also printed with hex-encoded values, when they had a binary collation. The latter issue also affected regular indexes, but was more visible with multi-valued indexes, since these always use `utf8mb4_0900_bin` collation. Now, hex-encoding is used only for string types having a binary character set. Strings with non-binary character sets are now printed in `EXPLAIN FORMAT=TREE` output as plain text, with escaping for any special characters. (Bug #33343948)

- For some functions, the resolved character set was not always the same as the character set of the first argument. (Bug #32668730, Bug #33238711)

## Compilation Notes

- **InnoDB:** Compilation issues associated with the following MSVC++ level 4 compiler warnings were addressed: C4201, C4701, C4702, C4703, C4706. The compiler warnings, which were previously disabled, are now enabled. (Bug #33464699)
- **InnoDB:** MSVC++ level 4 compiler warnings were enabled. (Bug #33437498)
- **InnoDB:** An access violation occurred when building a debug version of MySQL using Visual Studio 2019 version 16.10 or version 16.11. The violation was due to an STL iterator bug. (Bug #33223243)
- Binary packages that include curl rather than linking to the system curl library have been upgraded to use curl 7.80.0. (Bug #33576431)

## Data Type Notes

- This release fixes the following two issues relating to date and time values:

  - Inserting a `CHAR` value such as `'12:00:00'` into a `DATE`, `DATETIME`, or `TIMESTAMP` column raised the wrong error. In the case of a `DATE` column, this error was similar to `Data truncation: Incorrect date value: '2012-00-00' for column 'd' at row 1`. This occurred for both the binary and text protocols.
  - Inserting a value with an offset into a `DATE` or `TIME` column using the binary protocol gave a wrong result. For example, when the connection time zone was set to GMT-5, inserting `'2021-10-10 00:00:00.123+01:00'` into a `TIME` column yielded `'18:00:00'`; that is, the value was converted to the connection time zone (this should be done only with respect to `DATETIME` columns).

  We fix these by recognizing and adjusting for time zone offsets whenever a `TIMESTAMP` value with a time zone offset is inserted into a `TIME` or `DATE` column. (Bug #33616957, Bug #33649009)

  References: See also: Bug #33539844.

- A fix in MySQL 8.0.27 for a previous issue changed the resolved type of a boolean expression from signed `INT` to unsigned `BIGINT` in order to simplify type handling, but what appeared then as a harmless metadata change turned out to cause problems for some of the MySQL Connectors.

  We now revert the metadata change and provide a different fix for the original problem, by adjusting the `max_length` for the negation of an integer to at least two characters. (Bug #33516898)

  References: This issue is a regression of: Bug #33117410.

- Sorts of some column types, including `JSON` and `TEXT`, sometimes exhausted the sort buffer if its size was not at least 15 times that of the largest row in the sort. Now the sort buffer need only be only 15 times as large as the largest sort key. (Bug #103325, Bug #105532, Bug #32738705, Bug #33501541)

  References: This issue is a regression of: Bug #30400985, Bug #30804356.

## Deprecation and Removal Notes

- Support for the TLS v1.0 and TLS v1.1 connection protocols is removed as of MySQL 8.0.28. The protocols were deprecated from MySQL 8.0.26. For background, refer to the IETF memo Deprecating TLS v1.0 and TLSv 1.1. Make connections using the more-secure TLSv1.2 and TLSv1.3 protocols. TLSv1.3 requires that both the MySQL Server software and the client application were compiled with OpenSSL 1.1.1 or higher.

From MySQL 8.0.28, client programs, including MySQL Shell, that support a `--tls-version` option for specifying TLS protocols for connections to the MySQL server cannot make a TLS/SSL connection with the protocol set to TLSv1 or TLSv1.1. If a client attempts to connect using these protocols, for TCP connections, the connection fails, and an error is returned to the client. For socket connections, if `--ssl-mode` is set to REQUIRED, the connection fails, otherwise the connection is made but with TLS/SSL disabled.

On the server side, the following settings are changed from MySQL 8.0.28:

- Permitted values for `tls_version` and `admin_tls_version` no longer include `TLSv1` or `TLSv1.1`.
- Permitted values for `group_replication_recovery_tls_version` no longer include `TLSv1` or `TLSv1.1`.
- For asynchronous replication, replicas can no longer set the protocol for connections to the source server (`SOURCE_TLS_VERSION` option for `CHANGE REPLICATION SOURCE TO`) to `TLSv1` or `TLSv1.1`.

(WL #14775)

- The shortcuts `ASCII` for `CHARACTER SET latin1` and `UNICODE` for `CHARACTER SET ucs2` are now deprecated, and you should expect their removal in a future version of MySQL. Using either of these now raises a warning; use `CHARACTER SET` instead. (WL #13146)
- The character sets listed here, along with all of their collations, are now deprecated, and subject to removal in a subsequent release of MySQL:
  - `ucs2`
  - `macroman` and `macce`
  - `dec`
  - `hp8`

  The use of any of these character sets or their collations in SQL statements or elsewhere in the MySQL server now produces a deprecation warning.

  You should use `utf8mb4` instead of any of the character sets just listed. See also The ucs2 Character Set (UCS-2 Unicode Encoding), West European Character Sets, and Central European Character Sets. (WL #13594, WL #13595, WL #13596, WL #13597)

## Event Scheduler Notes

- As a previous convenience, the server automatically restarted the Event Scheduler as needed when the `super_read_only` system variable was disabled. Now this same convenience is applied independently when the `read_only` system variable is disabled. Prior to this update, disabling `read_only` could also disable `super_read_only` if needed, but because the code was separate, the Event Scheduler was not restarted. (Bug #33539082)

## Full-Text Search Notes

- Due to the fact that, as implemented, `MATCH()` does not act as a function of its arguments, but rather as a function of the row ID of the current row in the underlying scan of the base table, a query using a rollup column as the argument to this function tended to perform poorly, and with unreliable results. This

being the case, the use of a rollup column with `MATCH()` is no longer permitted whenever the following conditions are true:

- `MATCH()` appears in the `SELECT` list, `GROUP BY` clause, `HAVING` clause, or `ORDER BY` clause.
- The query uses `GROUP BY ... WITH ROLLUP`.
- A grouping column is used as the argument to `MATCH()`.

Any such queries are now rejected with `ER_FULLTEXT_WITH_ROLLUP`. (The use of `MATCH()` with a rollup column in the `WHERE` clause is not affected by this change, and is still permitted.)

For more information, see Full-Text Search Functions. (Bug #32565923, Bug #32996762, WL #14697)

## SQL Function and Operator Notes

- **Important Change:** When using prepared statements, the `DATE_ADD()` and `DATE_SUB()` functions returned `DATETIME` values, even when calculations involved combinations of `YEAR`, `MONTH`, or `DAY` parts only (that is, no time parts).

  Previous to implementing single preparation of prepared statements in MySQL 8.0.22, `TIME` values were returned in such cases; before this was done, values used as arguments and their types were used to determine the result type of certain temporal functions at resolution time, such as `DATE_ADD()`, `DATE_SUB()`, and `ADDTIME()`. Afterwards, user variable references and dynamic parameters are considered constant for the lifetime of one execution only, requiring that the return type be determined in another fashion, in this case from the function type. For example, the default resolved type for `DATE_ADD()` was deemed to be `DATETIME` if the first argument was a dynamic parameter, since `DATETIME` accomodates all temporal values and thus an implicit reprepare can be avoided.

  The change just described represents a regression; the problem is better solved by deriving a more precise resolved data type, and performing a reprepare only if that does not match the actual value of the parameter. (Such functionality was already in use in the MySQL server for numeric parameters.) This solution is implemented by this fix.

  We now parse string and numeric values when temporal values are expected. When a valid temporal value is found, the value is converted. This fix also improves determination of resolved data types for temporal functions.

  With this fix, the `DATE_ADD()` and `DATE_SUB()` functions (and their synonyms functions `ADDDATE()` and `SUBDATE()`) resolve types as follows:

  - If the first argument is a dynamic parameter, its resolved type is `DATE` if the second argument is an interval that contains some combination of `YEAR`, `MONTH`, or `DAY` values only; otherwise, its type is `DATETIME`.
  - If the first argument is resolved as `DATETIME`, the resolved type of the function is also `DATETIME`.
  - If the first argument is a `DATE`, the resolved type of the function is also `DATE`, unless the interval argument uses `HOUR`, `MINUTE`, or `SECOND`, in which case it is `DATETIME`.
  - If the first argument is a `TIME` value, the resolved type is also `TIME`, unless the interval argument uses any of `YEAR`, `MONTH`, or `DAY`, in which case the resolved type of the function is `DATETIME`.

  If none of the preceding conditions are met, the function is resolved as `VARCHAR` (as in MySQL 8.0.21 and earlier).

The `ADDTIME()` and `SUBTIME()` functions now resolve types as follows:

- If the first argument is a dynamic parameter, the resolved type is `TIME`, rather than `DATETIME`.
- Otherwise, the resolved type of the function is derived from the resolved type of the first argument.

In addition, for `DATE_ADD()` and `DATE_SUB()`, if the resolved type of the first argument is `DATE`, and a `DATETIME` value is provided, the statement is reprepared so that the function has the resolved type `DATETIME`. Behavior is unchanged when a `TIME` value is provided.

For `ADDTIME()` and `SUBTIME()`, there are no forced reprepares. (Bug #103781, Bug #32915973, Bug #33477883, Bug #33539844)

- Previously, loadable functions and stored functions shared the same namespace and could not have the same name. A subsequent implementation change eliminated the reason to share the same namespace and permitted a stored function to be created with the same name as an existing loadable function. To invoke the stored function, it is necessary to qualify it with a schema name. The server now generates a warning if the stored function name collides with an existing loadable function name. (Bug #33301931)
- Queries making use of the `MBRContains()` function did not employ all available spatial indexes. (Bug #32975221)

  References: This issue is a regression of: Bug #29770705.
- The `FORMAT()` function returned a formatted number without showing the thousands separator and grouping between separators when either the es_ES or es_MX locale was specified. (Bug #31374305)
- The result length of the `GROUP_CONCAT()` function was wrong when the value of `group_concat_max_len` was increased. With a small `group_concat_max_len` value, the result was correct. This issue was caused by arithmetic overflow.

  Our thanks to Hope Lee for the contribution. (Bug #105380, Bug #33521497)

## Optimizer Notes

- For a query, a range scan can be picked first and the optimizer decide that the same range scan can be used to skip ordering. In some cases, when the requested order is not the same as the index order, a reverse index scan is required. If the requested ordering is on the key part that is not already used by the range scan, we update the number of used key parts for the range scan to reflect the change. The issue occurred because the key part information was not also updated, and when it was necessary to access key part information based on the number of key parts used.

  We also now note when a reverse scan uses extended key parts, and set the correct flags for evaluation accordingly. (Bug #33615893)

  References: This issue is a regression of: Bug #33037007.
- In both optimizer trace and `EXPLAIN FORMAT=TREE` output, date ranges were printed as binary. Now in such cases we print temporal values as quoted strings. (Bug #33335079)
- When a condition was pushed down, the result of evaluating assignments to user variables in the `SELECT` list of the subquery were sometimes affected. For this reason, we now prevent condition pushdown for statements with assignments to user variables.

  Our thanks to Casa Zhang and the Tencent team for the contribution. (Bug #104918, Bug #33341080)

- When the join buffer was set to certain arbitrary sizes, the number of chunk files created for hash joins was too small. This meant that each file contained more rows than could fit in the join buffer, so that the probe chunks needed to be read multiple times. This was caused by using integer division when computing how many chunk files are needed; we fix this by using floating-point division instead.

  Our thanks to Øystein Grøvlen for the contribution. (Bug #104186, Bug #33073354)

- A query using aggregation on a `BIT` type could return different results depending on the indexes or join type employed. This was due to the fact that a DML statement using such an aggregation caches the `BIT` values using an integer type, and later looks up and converts to a string format for output. The current issue arose because this lookup treated the `BIT` value as an integer, resulting in an incorrect string value.

  This is fixed by adding a new internal class for cached `BIT` values which can convert bit values to string formats correctly.

  Our thanks to Hope Lee for the contribution. (Bug #100859, Bug #31894023)

- When a DML statement containing an outer `DISTINCT` query with a subquery inside a `HAVING` clause, the inner subquery attempts to use a column reference for a column from the outer `DISTINCT` query, but this should be allowed only if the subquery is used somewhere outside of the `HAVING`, or if the outer `SELECT` does not use grouping. The current issue came about because such a query was allowed to run even though neither of these conditions were met.

  To fix this, the column reference check is expanded to detect an invalid column reference of this sort, and to return an error if it does.

  Our thanks to Song Zhibai for the contribution. (Bug #97742, Bug #30617496)

## Packaging Notes

- The GnuPG build key used to sign MySQL downloadable packages has been updated. The previous GnuPG build key is set to expire on 2022-02-16. For information about verifying the integrity and authenticity of MySQL downloadable packages using GnuPG signature checking, or to obtain a copy of our public GnuPG build key, see Signature Checking Using GnuPG.

  Due to the GnuPG key update, systems configured to use `repo.mysql.com` may report a signature verification error when upgrading to MySQL 5.7.37 and higher or to MySQL 8.0.28 and higher using `apt` or `yum`. Use one of the following methods to resolve this issue:

  1. Manually reinstall the MySQL APT or YUM repository setup package from https://dev.mysql.com/downloads/.

  2. Download the MySQL GnuPG public key and add it your system GPG keyring.

     - For MySQL APT repository instructions, see Appendix A: Adding and Configuring the MySQL APT Repository Manually.

     - For MySQL YUM repository instructions, see Upgrading MySQL with the MySQL Yum Repository.

  (Bug #33587308)

- The bundled `libedit` library was upgraded to version 20210910-3.1. (Bug #33568767)

## Performance Schema Notes

- A new statement metric, `CPU_TIME`, is now available, enabling you to measure the CPU time spent on a query.

  The following changes were made to support this:

  - A timer, `THREAD_CPU`, was added to the Performance Schema `PERFORMANCE_TIMERS` table.
  - A consumer, `events_statements_cpu` was added to the Performance Schema `setup_consumers` table.

    `events_statements_cpu` is disabled by default.
  - A Performance Schema command option, `performance-schema-consumer-events-statements-cpu` to enable or disable the consumer, `events_statements_cpu`.

    See Performance Schema Command Options for more information.
  - The following columns were added:
    - `CPU_TIME` column was added to the following Performance Schema tables:
      - `events_statements_current`
      - `events_statements_history`
      - `events_statements_history_long`

      `CPU_TIME` is the time spent on CPU for the current thread, expressed in picoseconds.
    - `SUM_CPU_TIME` column was added to the following Performance Schema tables:
      - `events_statements_summary_by_thread_by_event_name`
      - `events_statements_summary_by_account_by_event_name`
      - `events_statements_summary_by_user_by_event_name`
      - `events_statements_summary_by_host_by_event_name`
      - `events_statements_summary_global_by_event_name`
      - `events_statements_summary_by_digest`
      - `events_statements_summary_by_program`
      - `prepared_statements_instances`

      SUM_CPU_TIME is the CPU time spent on the current thread, expressed in picoseconds, for the corresponding aggregations.
    - `CPU_LATENCY` column was added to the following sys schema tables:
      - `statement_analysis`
      - `user_summary_by_statement_type`

- `user_summary_by_statement_latency`
- `host_summary_by_statement_type`
- `host_summary_by_statement_latency`
- `processlist`
- `session`

CPU latency is the CPU time spent on the current thread, expressed in human-readable form.

- `CPU_LATENCY` column was added to the following sys schema `x$` tables:
  - `x$statement_analysis`
  - `x$user_summary_by_statement_type`
  - `x$host_summary_by_statement_type`
  - `x$processlist`
  - `x$session`
  - `x$user_summary_by_statement_latency`
  - `x$host_summary_by_statement_latency`

  CPU latency is the CPU time spent on the current thread, expressed in picoseconds.

Our thanks to Facebook for their contribution to this fix. (Bug #32202060, Bug #101764, WL #14779)

## Functionality Added or Changed

- **Important Change:** The number of distinct windows which can appear in a given `SELECT` is now limited to 127. The number of distinct windows is the sum of the named windows and the implicit windows specified as part of any window function's `OVER` clause.

  In order to use a large number of windows, it may be necessary to increase the value of the `thread_stack` server system variable. (Bug #33279604)

- **InnoDB:** `InnoDB` now supports `ALTER TABLE ... RENAME COLUMN` operations using `ALGORITHM=INSTANT`.

  Operations that support `ALGORITHM=INSTANT` only modify metadata in the data dictionary. Table data is unaffected, making the operations instantaneous. If not specified explicitly, `ALGORITHM=INSTANT` is used by default by DDL operations that support it.

  For more information about this and other DDL operations that support `ALGORITHM=INSTANT`, see Online DDL Operations. (WL #14785)

- It is theoretically possible for a user with sufficient permissions using MySQL Enterprise Audit to mistakenly create an “abort” item in the audit log filter that prevents themselves and other administrators from accessing the system. From MySQL 8.0.28, the `AUDIT_ABORT_EXEMPT` privilege is available to permit a user account’s queries to always be executed even if an “abort” item would block them. Accounts with this privilege can therefore be used to regain access to a system following an audit

misconfiguration. The query is still logged in the audit log, but instead of being rejected, it is permitted due to the privilege.

Accounts created in MySQL 8.0.28 or later with the `SYSTEM_USER` privilege have the `AUDIT_ABORT_EXEMPT` privilege assigned automatically when they are created. The `AUDIT_ABORT_EXEMPT` privilege is also assigned to existing accounts with the `SYSTEM_USER` privilege when you carry out an upgrade procedure with MySQL 8.0.28 or later, if no existing accounts have that privilege assigned. (WL #14670)

- The `tmp_table_size` variable now defines the maximum size of individual in-memory internal temporary tables created by the TempTable storage engine. An appropriate size limit prevents individual queries from consuming an inordinate amount global TempTable resources. See Internal Temporary Table Storage Engine. (WL #14647)

- The `innodb_open_files` variable, which defines the number of files `InnoDB` can have open at one time, can now be set at runtime using a `SELECT innodb_set_open_files_limit(N)` statement. The statement executes a stored procedure that sets the new limit.

  To prevent non-LRU manged files from consuming the entire `innodb_open_files` limit, non-LRU managed files are now limited to 90 percent of the `innodb_open_files` limit, which reserves 10 percent of the `innodb_open_files` limit for LRU managed files.

  The `innodb_open_files` limit now includes temporary tablespace files, which were not counted toward the limit previously. (WL #14591)

- The functions `FROM_UNIXTIME()`, `UNIX_TIMESTAMP()`, and `CONVERT_TZ()` now handle 64-bit values on platforms that support them, including 64-bit versions of Linux, MacOS, and Windows.

  On compatible platforms, `FROM_UNIXTIME()` now accepts a maximum argument of 32536771199.999999 seconds, corresponding to `'3001-01-18 23:59:59.999999'` UTC (including the optional fraction of up to 6 digits). If the argument is larger than this, the function returns `NULL`.

  On compatible platforms, `UNIX_TIMESTAMP()` now accepts a maximum value of `'3001-01-18 23:59:59.999999'` UTC, corresponding to 32536771199.999999 seconds since the Unix Epoch. If the argument is larger than this, the function returns `0`.

  In addition, on compatible platforms, `CONVERT_TZ()` now performs time zone conversion beyond 2038, up to `'3001-01-18 23:59:59.999999'` UTC. If the datetime argument exceeds this value, the argument is returned unchanged. This “no-op” behavior is the same as previously with values beyond `'2038-01-19 03:14:07.999999'` UTC.

  The behavior of these 3 functions on 32-bit platforms is unchanged.

  The behavior of the `TIMESTAMP` type is also unaffected by this change; its maximum allowed value remains `'2038-01-19 03:14:07.999999'` UTC, regardless of platform. For dates futureward of this, use the MySQL `DATETIME` type instead. (WL #14630)

- This release introduces monitoring and limiting of memory allocation on a global and per-user basis. You can now observe the total memory consumed by all user connections by checking the

value of the `Global_connection_memory` status variable, which must be enabled by setting `global_connection_memory_tracking = 1`.

The total includes memory used by system processes, or by the MySQL root user, although these users are not subject to disconnection due to memory usage.

Memory used by the `InnoDB` buffer pool is not included in the total.

You can control indirectly how often the status variable is updated by adjusting `connection_memory_chunk_size`; `Global_connection_memory` is updated only when the total memory usage varies by more than this amount.

You can specify limits on resource consumption per user connection by setting `connection_memory_limit`; any user whose memory usage exceeds this amount cannot issue additional queries. You can also impose a global memory limit by setting `global_connection_memory_limit`. Whenever `Global_connection_memory` exceeds the global limit, no regular users can issue new queries requiring memory usage. System users such as MySQL root are not bound by these limits. (WL #13458)

## Bugs Fixed

- **InnoDB:** The minimum I/O buffer size calculated during an index creation operation did not align with the I/O block size, permitting a record to exceed the buffer boundary. (Bug #33570629)
- **InnoDB:** The `sync_array_detect_deadlock` algorithm used by the semaphore deadlock checker in debug builds was simplified in terms of code and time complexity, and an implementation of the algorithm was introduced for use in release builds. (Bug #33538505)
- **InnoDB:** The `ut::make_unique` library function in the `InnoDB` sources now permits specifying the type of field allocated. (Bug #33538461)
- **InnoDB:** A Performance Schema instrumentation was added for tracking redo log buffer memory allocations. (Bug #33527660)
- **InnoDB:** Warnings printed to the error log for long semaphore waits did not provide information about the latch owner. (Bug #33509386)
- **InnoDB:** A latch release and reacquisition mechanism was introduced to reduce the amount of time that threads spend in critical sections protected by a global lock system latch. (Bug #33502610, Bug #33563523)
- **InnoDB:** A hole punch operation on Windows caused a failure. The operation was performed as an overlapped (asynchronous) operation, which requires a `OVERLAPPED` structure containing a handle to an event object. The `OVERLAPPED` structure was not provided. (Bug #33496778)
- **InnoDB:** The `ut_time()` infrastructure in the `InnoDB` sources was replaced with a type-checked standard library implementation. (Bug #33460092)
- **InnoDB:** Numerous `Trying to access missing tablespace` errors were printed to the error log following a restore operation. (Bug #33437625)
- **InnoDB:** Performance Schema aware `ut::make_unique` and `ut::make_shared` memory management library functions were added to the `InnoDB` sources. Similar functions (`ut::make_unique_aligned` and `ut::make_shared_aligned`) were added for types with extended alignment. (Bug #33420694)
- **InnoDB:** The `buf_validate()` function in the `InnoDB` sources was optimized, improving performance on debug builds.

Thanks to Hobert Lu for the contribution. (Bug #33417058, Bug #104967)

- **InnoDB:** On a NUMA-enabled system, the page size of memory chunks allocated to the buffer pool did not align with the system page size in certain scenarios, causing in the following error: `Failed to set NUMA memory policy of buffer pool page frames to MPOL_INTERLEAVE`. (Bug #33406701)

  References: This issue is a regression of: Bug #32714144.

- **InnoDB:** Two instances of `std::unique_ptr` with `mem_heap` in the `InnoDB` sources now use the `Scoped_heap()` wrapper, which uses a stateless function object instead of a pointer to a function. (Bug #33405520)

- **InnoDB:** The `m_end_range` flag in the `prebuilt` struct, which is set to true when the end of the range is exceeded while populating the prefetch cache, was not set to false when the prefetch cache was reset (initialized). As a result, in cases where the end of the range is not exceeded and the handler is reused, the `m_end_range` flag could be set incorrectly the next time the prefect cache is used. (Bug #33384537)

- **InnoDB:** Column metadata in the data dictionary was not updated when a new table ID was assigned to a table after discarding the table's tablespace during a table import operation. (Bug #33319149)

- **InnoDB:** Setting the `innodb_interpreter` debug-only system variable to `NULL` caused a failure. (Bug #33316661)

- **InnoDB:** Full-text index creation file management was improved. (Bug #33270893)

- **InnoDB:** An update operation that inserted a new row into a temporary table used for aggregation caused the temporary table to be moved to disk and the update operation to be retried on the new on-disk temporary table. A BLOB pointer in the record data prepared before the temporary table was moved to disk was rendered stale, causing a failure. (Bug #33242407)

- **InnoDB:** Memory allocation is now performed by a new standards-compliant custom memory allocator which is compatible with Performance Schema. (Bug #33159210)

- **InnoDB:** A race condition between threads attempting to deinitialize and initialize statistics for the same table raised and assertion failure. (Bug #33135425)

- **InnoDB:** An `innodb_flush_log_at_trx_commit` setting other than 1 or a long running transaction could have resulted in an inconsistent rate of redo log flushing. (Bug #33128461)

- **InnoDB:** Allocation of large pages is now handled by a library designed to handle this. In cases where the large page allocation mechanism cannot be used, a fallback mechanism allocates regular aligned pages. Fallback can occur when large page address space is exhausted, when large page support is disabled by the underlying hardware or operating system architecture, or when large page support in MySQL is disabled explicitly (`--large-pages=0`). Occurrences of `ut_allocator` functions for allocation and deallocation of large pages have been replaced by the new library functions. (Bug #33119009, Bug #33118309, Bug #33149501, Bug #32135935)

- **InnoDB:** Handling of regular 4K page-aligned allocations is now performed by a self-contained library which is compatible with Performance Schema. (Bug #33118362)

- **InnoDB:** Functions belonging to a new `InnoDB` library responsible for dynamic storage management of suitably aligned types has replaced the functions previously used for this purpose. (Bug #33110341)

- **InnoDB:** Dynamic allocation of suitably aligned types is now handled by a library which is compatible with Performance Schema. (Bug #33110312)

- **InnoDB:** While a purge thread was freeing LOB pages at end of a purge batch, a required index data structure was freed, causing a failure. (Bug #32918325)
- **InnoDB:** Inconsistencies in Performance Schema instrumentation logic for dynamic memory management functions (`ut_*` functions) were addressed. (Bug #32715466)
- **InnoDB:** `InnoDB` dynamic allocation routine limitations prevented dynamic allocation of an array of constructible types. The limitations have been addressed, permitting allocation of default constructible types, non-default constructible types, and types that are both default and non-default constructible. (Bug #32714047)
- **InnoDB:** When using `READ COMMITTED` or `READ UNCOMMITTED`, certain queries executed on a table inside of a trigger or function prevented concurrent transactions on the same table. The acquired locks were too restrictive. (Bug #32636792)
- **InnoDB:** Hole punch functionality was not working as expected for tablespace pages that were encrypted and compressed, for most pages on Windows, and for Windows volumes that do not implement Microsoft volume management functions. (Bug #32136255)
- **InnoDB:** Replacing the `ut_time()` infrastructure with a type-checked standard library (`std`) implementation in MySQL 8.0.28 caused a performance regression. The regression was due the time library used by `std` clocks, which generates system calls. Low resolution clocks are now used instead. (Bug #107763, Bug #34352870)

  References: This issue is a regression of: Bug #33460092.
- **Packaging:** Added an `icu-data-files` RPM that was separated from the `server` package. This is a package of ICU data files needed by MySQL regular expressions. (Bug #35162346)
- **Partitioning:** Creating a table with nondeterministic functions in generated column expressions should not be possible, but this was not enforced in all cases; a series of one or more `ALTER TABLE` statements could be employed to arrive at a partitioned table with one or more such generated columns. When attempting to execute the `CREATE TABLE` statement obtained by running `SHOW CREATE TABLE` against this table, MySQL rejected the statement with a misleading error message referring to the partitioning expression rather than to the problematic column, despite the fact that the partitioning expression itself was legal.

  This was caused by the result of a check for any unsafe expressions defined for a generated column (in the internal variable `thd->safe_to_cache_query`), which was later checked again without being cleared while parsing the partition expression, leading to an error even when the partition expression did not refer to the problematic generated column expression. Now in such cases, we reset `thd->safe_to_cache_query` before parsing the partition function.

  The issue of allowing the use of certain nondeterminstic functions (`AES_ENCRYPT()`, `AES_DECRYPT()`, `RANDOM_BYTES()`) in generated columns is handled separately. (Bug #29268656)

  References: See also: Bug #32592320.
- **Partitioning:** A query using an index other than the primary key of a partitioned table sometimes resulted in excessive CPU load. (Bug #104576, Bug #33238010)
- **Replication:** A `PURGE BINARY LOGS` statement could be issued while the instance was locked for backup, which contravened the rules of the backup lock by removing files from the server. The statement now cannot be used while a LOCK INSTANCE FOR BACKUP statement is in effect. (Bug #33437026)
- **Replication:** Replication stopped with an error when reading a table map event if the name of the table or database was over 64 bytes – the limit is 64 characters, so the use of multi-byte characters could

cause this situation. The replica now no longer checks the size of the table and database names, and supports the transmission of longer names. (Bug #33305975, Bug #104798)

- **Replication:** A lock conflict could occur if the Performance Schema table `replication_applier_status_by_worker` was queried while a `STOP REPLICA` command was in progress. The issue has now been resolved. (Bug #33290947)
- **Replication:** From MySQL 8.0.26, new versions of the plugins that implement semisynchronous replication are supplied to replace the terms “master” and “slave” with “source” and “replica”. Following this, the `UNINSTALL PLUGIN` statement incorrectly allowed the old versions of the plugins, `rpl_semi_sync_master` and `rpl_semi_sync_slave`, to be uninstalled when replication channels were currently using them. The issue has now been fixed. (Bug #33270401)
- **Replication:** When the `PAD_CHAR_TO_FULL_LENGTH` SQL mode was enabled on a replica server, trailing spaces could be added to a replication channel’s name in the replication metadata repository tables, resulting in errors in replication operations that identified the channel using that data. The issue has now been fixed in MySQL 8.0 by using VARCHAR for character columns, and in MySQL 5.7 by disabling the SQL mode when reading from those tables. Thanks to Brian Yue for the contribution. (Bug #33213841)
- **Replication:** If a replica was disconnecting while a `SHOW REPLICAS` statement was being issued, the server was able to access deleted data. (Bug #33206343, Bug #104566)
- **Replication:** If a replica server with the system variable `replica_preserve_commit_order = 1` set was used under intensive load for a long period, the instance could run out of commit order sequence tickets. Incorrect behavior after the maximum value was exceeded caused the applier to hang and the applier worker threads to wait indefinitely on the commit order queue. The commit order sequence ticket generator now wraps around correctly. Thanks to Zhai Weixiang for the contribution. (Bug #32891221, Bug #103636)
- **Replication:** The replication receiver thread stopped with an error if the replication source server sent a heartbeat event containing a binary log file position that was above the 4GB offset, due to the large size of the binary log file. A new heartbeat event (Heartbeat_log_event_v2, log event type 41) that handles the larger value correctly has been added for use in this situation. (Bug #29913991)
- **Group Replication:** Group Replication could stop unexpectedly during the auto-rejoin procedure while the Performance Schema replication group member statistics table was being checked. The concurrency of the operations is now handled correctly. (Bug #33609089)
- **Group Replication:** Group Replication's selection process for a group member to be the donor for distributed recovery involves the use of the standard random selector defined for the operating system. If this random device was not available and an exception was thrown, the member's joining process stopped. Group Replication now takes this possibility into account and uses a fallback random selector if the one on the operating system returns an error. (Bug #33596124)
- **Group Replication:** The `STOP GROUP_REPLICATION` statement stops asynchronous replication channels on the group member, but it does not implicitly commit transactions that are in progress on them like `STOP REPLICA` does. This is because on a Group Replication group member, an additional transaction committed during the shutdown operation would leave the member inconsistent with the group and cause an issue with rejoining. The server is left in a read-only state after the operation is completed. This situation leads to failed commits for transactions that are in progress while stopping Group Replication, so to avoid these, the `STOP GROUP_REPLICATION` statement now cannot be issued while a GTID is assigned as the value of the `gtid_next` system variable. (Bug #33412483)
- **Group Replication:** An expelled group member using the Group Replication auto-rejoin procedure reported its state as RECOVERING too early on, while it was still collecting information from other group members and before the compatibility checks were complete. The state change is now carried out while

the view is being installed, which is the point where the rejoining member is actually accepted as a group member. (Bug #33380840)

- **Group Replication:** Disabling the Group Replication plugin in a MySQL source distribution using the CMake option `DWITH_GROUP_REPLICATION=0` did not disable applications and tests related to Group Replication, which caused them to build incorrectly. (Bug #33308513)

- **Group Replication:** In Group Replication, if a `SET gtid_next` statement is used on a group member to set the GTID for the next transaction, it is possible for the same GTID to be used for a transaction that starts concurrently on another member. If both transactions reach the commit stage, the second one in the total order is rolled back, resolving the situation. However, when the transaction consistency level for Group Replication (the `group_replication_consistency` system variable) was set to `BEFORE` or `BEFORE_AND_AFTER`, members could reach a deadlock with one holding ownership of a GTID in the `gtid_owned` set, and another waiting for ownership to be released before committing the transaction. The wait function now only considers the GTIDs for committed transactions and not the GTIDs that are owned but not committed, except where a session owns a GTID that is concurrently committed, in which case the executing session errors out. (Bug #33051454, Bug #104096)

- **Group Replication:** The group communication engine for Group Replication (XCom, a Paxos variant) now logs more information in the situation where the existing group members have difficulty in communicating with a joining member, for example due to network issues. This can result in the group remembering an old incarnation of the joining member and not allowing it to attempt to join again. (Bug #32873315)

- **Group Replication:** Group Replication's Group Communication System (GCS) now differentiates in its records of expelled members between those that had successfully joined the group, and those that never did manage to join the group. (Bug #32630484)

- **Group Replication:** A race condition occurred if the Group Replication group member statistics in the Performance Schema were queried when Group Replication was being started or stopped. (Bug #32392468)

- **Microsoft Windows:** On Windows, added missing debug and test suite binaries for MySQL Server (commercial) and MySQL NDB Cluster (commercial and community). (Bug #32713189)

- **JSON:** When the first argument passed to `JSON_TABLE()` was a row instead of a single JSON value, an assertion was raised while trying to evaluate the row expression. We fix this by raising an error during function resolution if the first argument is a row, so that the row expression itself is never evaluated. (Bug #33414235)

- Using `LPAD()` or `RPAD()` in the expression for a generated column led to corrupted indexes on the parent table. (Bug #33661337)

  References: See also: Bug #32668730, Bug #33238711.

- In some cases where warnings were issued, rows were missing from the results of aggregation using a temporary table. (Bug #33603911)

- For openSUSE 15, added the libtirpc rpcgen build requirement in `mysql.spec` to now use TI-RPC. (Bug #33582982)

- An `UPDATE` statement acting on multiple tables sometimes adds elements to a list each time it is executed. Elements were never removed from this list, which increased the memory footprint for each execution of the statement. We fix this by clearing the element list following execution. (Bug #33574408)

- The size of the `HOST` column of the Performance Schema `processlist` table is increased from VARCHAR(255) to VARCHAR(261). (Bug #33570218)

- A keyring migration failure due to an OpenSSL error raised an assertion. The SSL error state was not flushed from the thread's OpenSSL error queue. (Bug #33546207)
- A process listing function call caused a failure. (Bug #33511690)
- The commercial Debian server packages contained two testing plugins (component_test_page_track_component.so and component_test_global_priv_registration.so); they were moved to the separate and optional testing package. (Bug #33504443)
- For Fedora, increased the release package version number from 1 to 10; this to help eliminate potential installation related problems with native Fedora dnf/yum packages of the same version. (Bug #33504443)
- Increased compat level to 11 for Debian-based packages as the previous level of 9 is deprecated; and replaced calls to dh_systemd_enable + dh_systemd_start with dh_installsystemd to satisfy requirements for compatibility level 10+. (Bug #33458752)
- A delete operation involving a full-text search query caused a failure. (Bug #33455243)
- An improperly handled error caused a startup failure when using the `keyring_okv` plugin. (Bug #33449117)
- For Debian, added a `mysql-community-server` dependency to the `mysql-community-server-debug` package so as to pull in all required packages needed by the debug build. (Bug #33426737)
- For virtual generated columns of type `DECIMAL`, we now always store some data, so that we avoid undefined behavior when trying to convert the field buffer to a decimal value. (Bug #33424846)
- MySQL now supports pushing a condition down to a derived table when an expression from the underlying derived table contains a variable set by a stored procedure. (Bug #33423394)
- `tls_version` and `admin_tls_version` settings are now validated server startup. (Bug #33390209)
- The `admin_tls_version` variable accepted an invalid value. (Bug #33389818)
- If two or more deprecated system variables were persisted using a `SET PERSIST` statement, when the server was restarted, a deprecation warning was only logged for the first of the deprecated system variables. (Bug #33388488)
- For an index range scan whose key parts are equal, the range is now shown as an equality. For example, `a = 3` is now displayed, instead of `3 <= a <= 3` previous to this change. (Bug #33351123)
- Replaced `/var/run` references with `/run` as `/var/run` usage is deprecated for `tmpfiles.d` configuration files. The symlink from `/var/run` to `/run` remains to keep current setups functional. (Bug #33351110, Bug #33588618)
- Executing `SHOW PROCESSLIST` or accessing `INFORMATION_SCHEMA.PROCESSLIST` on a server with a specific configuration caused a failure. (Bug #33341623)
- Added a mapping from ICU error code `U_FILE_ACCESS_ERROR` to the new MySQL error code `ER_REGEXP_MISSING_FILE`. (Bug #33326003)
- A failed keyring function argument validation check caused a failure. (Bug #33319782)
- The index range scan iterator did not always increment the number of rows examined as expected. (Bug #33305632)
- Enabling the `create_admin_listener_thread` system variable on the command line could cause a server exit during startup under specific error conditions. (Bug #33300587)

- The SUBSTR() function did not always correctly handle errors raised when trying to evaluate its arguments. (Bug #33290160)
- International Components for Unicode version 67 introduced a new implementation for \X (match a grapheme cluster), which requires locale data not currently included with MySQL.

  This means that, when using the version of ICU bundled with MySQL, a query using \X raises the error ER_REGEXP_MISSING_RESOURCE; when using ICU supplied by the system, we report ER_WARN_REGEXP_USING_DEFAULT as a Note. (Bug #33290090)
- A full-text search query on a table stored in the BLACKHOLE storage engine where the chosen query plan used a full-text index scan caused an error instead of returning an empty result set. (Bug #33250020)
- The LOCK_TIME returned by the performance schema was under evaluated, missing time spent in rows locks, and time spent when locking multiple tables. As of this release, LOCK_TIME accounts for:
  - all the time waited on SQL TABLES
  - all the time waited on DATA locks

  LOCK_TIME is now reported consistently in the slow log and the performance schema. (Bug #33236909)
- A new option \T for the mysql client prompt prints an asterisk (*) if the current session is inside a transaction block. You can use the option with the --prompt command-line option, in a MySQL option file, or with the MYSQL_PS1 environment variable. Thanks to Murakami Kohei for the contribution. (Bug #33214862, Bug #104598)
- Constant subqueries in RANGE INTERVAL expressions were not always handled correctly. (Bug #33197418)
- Decimal expressions which evaluated as NULL were not always handled correctly. (Bug #33169048)
- A user account that was granted a MySQL role with a global SELECT privilege was denied access to the mysql database. The user account's global privileges were not checked when the role was granted. (Bug #33159353, Bug #104423)
- When setting up an Item_ref to a SELECT alias, its cached properties are copied (including whether it is part of a ROLLUP expression or not). However, these might not yet be correctly computed, so the computation should to be done first or the values could be wrong. Having the wrong value could cause certain expressions to be materialized in an intermediate step when they should not (because they contain ROLLUP expressions that are not ready for computation, but having the wrong value is unknown at this point). The issue is fixed by forcing cached values to be recomputed when an item is designated as a rollup item. (Bug #33149402, Bug #104394)
- An invalid comment string detected while upgrading a table from MySQL 5.7 to MySQL 8.0 caused the upgrade to fail with errors that did not provide sufficient contextual information. (Bug #33148961)
- It was possible in some cases to create a generated column of type SERIAL, which is not allowed.

  See Numeric Data Type Syntax, and CREATE TABLE and Generated Columns, for more information (Bug #33141966)
- Statements which commit a transaction implicitly or explicitly are not allowed inside a trigger or a stored function. Both CREATE TRIGGER and CREATE FUNCTION should report an error (ER_COMMIT_NOT_ALLOWED_IN_SF_OR_TRG) in this case, but did not correctly handle DROP TABLESPACE. (Bug #33141958)
- A SHOW TABLE STATUS operation raised an assertion failure when run on a table defined with a very large AVG_ROW_LENGTH value. (Bug #33114368)

- When calculating the maximum number of rows likely to be read from a scan, the intermediate result was a double that could become greater than the maximum allowed value for a 64-bit unsigned integer. This triggered undefined behavior when converting the intermediate double value to an integer, which in some cases could lead to assert failures.

  We fix this by clamping the result in the range [1, `UINT64_MAX`]. (Bug #33066458)

- Queries using both `UNION` and `LIMIT 0` triggered an assert failure in debug builds. (Bug #33066455)

  References: This issue is a regression of: Bug #32302724.

- Renaming an event using `ALTER EVENT ... RENAME TO` did not delete the Performance Schema instrumentation for the original event. (Bug #33059358)

- An SSL handshake assertion was raised on debug builds when using the thread pool plugin. (Bug #33012398)

- Some prepared statements using either `GROUP BY WITH ROLLUP` or one or more window functions could be executed successfully only once. (Bug #33007266)

- An error occurred for statements of the form `INSERT INTO view VALUE(tuple) AS row_alias(id_list)`. When executing such statements, the server calls the internal function `Sql_cmd_insert_base::prepare_values_table()` in order to prepare the derived table created as a `VALUES` alias; this function populates `Sql_cmd_insert_base.values_field_list` with `Item_field` objects pointing to the columns of the underlying table. When inserting into a view rather than a table, an expected `real_item()` transform, needed to map from an `Item_view_ref` referencing the view column to an `Item_field` representing the corresponding column in the underlying table, was not performed. (Bug #32858783)

- Some multiply-nested subselects were not handled correctly, and could lead to an unplanned shutdown of the server. (Bug #32547812)

- Inspection of a session thread object during a `SHOW PROCESSLIST` operation and a concurrent change to the thread's security context resulted in a race condition. (Bug #32320541, Bug #102052)

- In cases where there are no operations to be performed on the result of a `UNION`, the rows are streamed without storing them in a temporary table, although a placeholder for the temporary table still exists in the query block. Since this table is not instantiated, a check for estimates of the cost of reading the rows from the table while calculating the access cost and optimizing for range based access had unpredictable results.

  We fix this by skipping retrieval of such row estimates in the case of an uninstantiated temporary table. (Bug #32255904)

- A multi-table `DELETE` statement using common table expressions was not always handled correctly. (Bug #32238558)

  References: This issue is a regression of: Bug #98330, Bug #30796015.

- If a `CR_UNKNOWN_ERROR` was to be sent to a client, an exception could occur. (Bug #31933415)

- SSL-related code was revised to avoid a potential memory leak. (Bug #31933295)

- In some cases, multiple-table `UPDATE` statements could block concurrent access. (Bug #31731752)

- Keyring system variables that use an internal slot for complex settings no longer accept a setting of `DEFAULT`. (Bug #30879700)

- The `Timestamp` column in the `mysql.tables_priv` and `myql.columns_priv` grant tables was set to a zero timestamp value (`"0000-00-00 00:00:00"`) for `GRANT` and `REVOKE` operations, preventing a logical restore of the grant tables. As of MySQL 8.0.28, a valid start time value is written to the Timestamp column.

  If you have existing grant table records with zero timestamp values that are preventing a logical restore of the grant tables, a workaround is to update records in the grant tables or in the dump files, replacing zero timestamp values with `CURRENT_TIMESTAMP`.

  Thanks to Venkatesh Prasad Venugopal for the contribution. (Bug #30863899, Bug #98495)

- Producing a per-table dump using `mysqldump` in MySQL 5.7 and 8.0 requires a longer execution time compared to MySQL 5.6. This is because the `information_schema.files` table, which is queried for information on log file groups by `mysqldump`, contains information about InnoDB data files as well as NDB data files from MySQL 5.7. In MySQL 8.0, the issue has been fixed by rewriting the query to select only the appropriate data files. In MySQL 5.7, Information Schema tables do not have indexes, so a full table scan is still required. (Bug #29210990, Bug #93875)

- Keyring memory management was improved. (Bug #25042167)

- Incorrect values for `FORCE INDEX FOR GROUP BY` could be set while saving and restoring the presence of `FORCE INDEX` hints within tables. (Bug #105694, Bug #33604416)

- If a query with the `sql_buffer_result` system variable enabled returned just one row, and an attempt was made to insert the result into a table, then an error in setting the output from the temporary table could produce a data exception. (Bug #105351, Bug #33515752)

  References: This issue is a regression of: Bug #33152269.

- Resetting of the active slice was not performed in `WindowIterator::Read()` at the end the end of the input result set for windowing. This led to reading wrong values on reaching the `ORDER BY` sort, since the number of the active slice was still set to 1—that is, to the items being read from the input table—while the `ORDER BY` sorting stage needs to read the values after computation of any window functions. For this, it needs the active slice to be that of the last window's output table.

  We fix this by moving the resetting of the slice to the output slice immediately following the read, so that it is already set correctly when returning at the end of the input set and moving on to the ordering.

  Our thanks to Casa Zhang and the Tencent team for the contribution. (Bug #105045, Bug #33399696)

- Code inspection revealed use of `strncpy()` in the internal function `set_parse_error_message()` without making sure that the last byte of the buffer being copied into was a null byte. We fix this by using `snprintf()` instead of `strncpy()`; this ensures that the result is valid even if it is truncated. (Bug #104856, Bug #33321787)

- When executing prepared statements that activated a trigger created with the `DEFINER` clause (or a stored function), invoker privileges were used for checking table access instead of definer privileges. This, in turn, could cause privilege checks on tables used by the trigger or stored function to fail. (Bug #104168, Bug #33064461)

- When a singleton histogram is constructed, its cumulative frequency is calculated by adding frequencies of previous buckets with the current bucket; because a floating-point value was used for the accumulator, this sometimes led to accumulated float errors, with the final cumulative frequency fractionally greater than 1.0.

  This fix accumulates the frequency with an integer type instead, to avoid intermediate floating-point errors.

Our thanks to Casa Zhang and the Tencent team for the contribution. (Bug #104108, Bug #33045336)

- With `index_condition_pushdown=ON` and `transaction_isolation='READ-COMMITTED'`, locks on secondary indexes were not released until the transaction was either committed or rolled back, even though the secondary indexes were unmatched. (Bug #102722, Bug #32554667)

- Multi-valued indexes were not used for queries executed from within stored functions. (Bug #102359, Bug #32427727)

  References: See also: Bug #104700, Bug #33268466.

- An error occurred for an SQL statement having the form shown here:

  ```
  INSERT INTO target_table
    SELECT aggregate_expression, non_aggregate_expression
    FROM empty_table;
  ```

  This happened when the query plan used aggregation from a temporary table, and when *`non_aggregate_expression`* was constant during one execution of the query, but could vary between executions. Such an expression might, for example, include a function such as `NOW()` or `USER()`. This resulted in the temporary table getting a column for *`non_aggregate_expression`*, which is unnecessary, since all rows have the same value. In addition, if there were no rows, there was no legal value to insert into *`target_table`*, which is what actually triggered the error.

  We fix this by not using a temporary table column when *`non_aggregate_expression`* is `const` for the current execution. (Bug #102252, Bug #32384355, Bug #33546083)

- When executing a prepared statement that included values passed in as strings, MySQL attempted to parse them as integers and could return an error unrelated to the input value.

  After a recent change, dynamic parameter handling was refactored so that the derived data type for parameters was determined based on context. For example, in a comparison such as *`int_col = ?`*, the parameter was given the same type as the (integer) column it was compared to. To preserve compatibility with existing MySQL applications, if a decimal or float value was supplied as parameter, the statement was automatically reprepared with new type assigned to the parameter based on the actual value. This handling preserved compatibility for numeric parameters.

  However, if a string parameter was supplied, it was still interpreted as an integer (the resolved data type) and this behavior was not compatible with older MySQL versions that detected the actual type of the value. The consequences being that if *`int_col = ?`* is executed with the parameter value `'1.7'`, only the integer part of the string was used, making the effective comparison *`int_col = 1`*.

  To fix the issue, now when a string parameter is supplied, the parameter is analyzed to determine if it is an integer, a decimal, or a float value and the actual data type of the parameter is updated accordingly. Later, the actual type is compared to the resolved type and if it is incompatible, the statement is reprepared with the new actual type. So, the previous statement now evaluates as *`int_col = 1.7`* and the comparison evaluates using decimal numbers. (Bug #101806, Bug #32213576, Bug #103364, Bug #32787037)

# Changes in MySQL 8.0.27 (2021-10-19)

- Audit Log Notes
- Authentication Notes
- Character Set Support

- Compilation Notes
- Connection Management Notes
- Data Type Notes
- Deprecation and Removal Notes
- Storage Engine Notes
- Firewall Notes
- SQL Function and Operator Notes
- Keyring Notes
- Optimizer Notes
- Performance Schema Notes
- Pluggable Authentication
- Security Notes
- Server Administration
- Spatial Data Support
- Functionality Added or Changed
- Bugs Fixed


## Audit Log Notes

- A `CREATE USER` statement `BY 'auth_string'` clause was written to the audit log and general query log as an `AS 'auth_string'` clause. (Bug #33184550)

## Authentication Notes

- Previously, MySQL user accounts authenticated to the server using a single authentication method. MySQL now supports multifactor authentication (MFA), which makes it possible to create accounts that have up to three authentication methods. MFA support entails these changes:

  - `CREATE USER` and `ALTER USER` syntax has been extended to permit specification of multiple authentication methods.

  - The `authentication_policy` system variable enables MFA policy to be established by controlling how many factors can be used and the types of authentication permitted for each factor. This places constraints on how the authentication-related clauses of `CREATE USER` and `ALTER USER` statements may be used.

  - Client programs have new `--password1`, `--password2`, and `--password3` command-line options for specifying multiple passwords. For applications that use the C API, the new `MYSQL_OPT_USER_PASSWORD` option for the `mysql_options4()` C API function enables the same capability.

  In addition, MySQL Enterprise Edition now supports authentication to MySQL Server using devices such as smart cards, security keys, and biometric readers. This authentication method is based on the Fast

Identity Online (FIDO) standard, and uses a pair of plugins, `authentication_fido` on the server side and `authentication_fido_client` on the client side. The server-side FIDO authentication plugin is included only in MySQL Enterprise Edition distributions.

Multifactor authentication can use existing MySQL authentication methods, the new FIDO authentication method, or a combination of both. For more information, see Multifactor Authentication. (Bug #33159968, WL #14183)

- In cases where an authentication plugin performed no hashing of the authentication string, `CREATE USER` statements with a `BY 'auth_string'` clause failed with an error. (Bug #33125289)

## Character Set Support

- Regular expression functions now report an error when an expression or pattern cannot be converted to a character set which is suitable for the ICU regular expression engine.

  In addition, error checking in several geometry functions has been improved. (Bug #33290245)

- The `gen_dictionary()` function now takes `latin1` as the character set of its argument, and returns the same character set. (Bug #30389649)

- Otherwise identical strings, using, respectively, the ASCII (collation `ascii_general_ci`) and UCS2 (collation `ucs2_general_ci`) character sets did not match as expected in join conditions. (Bug #104571, Bug #33204161)

  References: See also: Bug #24847620, Bug #30746908, Bug #32244631, Bug #32501472.

- Given the default collation `c1` of a character set `cs`, and a different collation `c2` (that is, not equal to `c1`), then the statement `CREATE DATABASE d COLLATE c2 CHARACTER SET cs` created a new database with the default collation set to `c1` instead of `c2`. (Bug #104504, Bug #33183590)

## Compilation Notes

- **InnoDB:** A workaround was implemented for a Clang issue that causes a build failure on Windows (Bugzilla – Bug 51538). (Bug #33217633)

- MySQL now can be compiled using C++17. The following minimum version requirements apply for compiler support:

  - GCC 7.1 or Clang 5 (Linux)
  - XCode 10 (macOS)
  - GCC 10 (Solaris)
  - Visual Studio 2019 Update 4 (Windows)

  In particular, on Solaris, GCC is now the only supported compiler. The code has been cleaned up to remove adaptations and workarounds for Sun Studio, Oracle Studio, and SunPro. (Bug #32907274, Bug #103757, Bug #32907475, Bug #32992125, Bug #32992242, Bug #33004840, Bug #33086882)

## Connection Management Notes

- Previously, if the server restricted a client to the sandbox mode used to handle client connections for accounts with expired passwords, the client could use the `SET` statement. This is no longer permitted. For more information about sandbox mode, see Server Handling of Expired Passwords. (Bug #16369085, WL #13214)

## Data Type Notes

- `YEAR` values were not always interpreted correctly. (Bug #33142669)

  References: This issue is a regression of: Bug #31994744.

## Deprecation and Removal Notes

- **Important Change:** The `default_authentication_plugin` variable is deprecated as of MySQL 8.0.27; expect support for it to be removed in a future version of MySQL.

  The `default_authentication_plugin` variable is still used in MySQL 8.0.27, but in conjunction with and at a lower precedence than the new `authentication_policy` system variable, which is introduced in MySQL 8.0.27 with the multifactor authentication feature. For details, see The Default Authentication Plugin. (Bug #27515356, WL #14138)

- **Important Change:** The `BINARY` operator is now deprecated, and subject to removal in a future release of MySQL. Use of `BINARY` now causes a warning. Use `CAST(... AS BINARY)` instead. (WL #13619)

## Storage Engine Notes

- The `BLACKHOLE` storage engine maximum key length has been increased from 1000 to 3072 bytes (the same as `InnoDB`). Thanks to Adam Cable for the contribution. (Bug #32788749, Bug #103371)

## Firewall Notes

- The new `FIREWALL_EXEMPT` privilege exempts a user from firewall restrictions. This is useful, for example, for any database administrator who configures the firewall, to avoid the possibility of a misconfiguration causing even the administrator to be locked out and unable to execute statements. See MySQL Enterprise Firewall. (WL #14517)

## SQL Function and Operator Notes

- The `SPACE()` function did not handle certain large or unsigned values correctly. (Bug #33180446)
- Function arguments were not always evaluated correctly during resolution of functions defined within views. (Bug #33142010)

  References: This issue is a regression of: Bug #29904087.

- Bit functions in window expressions assert that the runtime size of a bit mask is not bigger than its resolve time size. We found several violations of this rule, listed here:

  - `ENCRYPT()` sometimes computed the maximum size of the result incorrectly.
  - `CONVERT()`, `CONCAT()`, `CONCAT_WS()`, `EXPORT_SET()`, `INSERT()`, `REPLACE()`, and `WEIGHT_STRING()` did not compute the maximum result length properly for the binary character set.
  - During resolution of `REPLACE(str, from_str, to_str)` we assumed that the entire length of *`from_str`* would be replaced for each match in *`str`*, but since *`from_str`* may be only 1 character long, it is possible for *`str`* to be replaced with multiple copies of *`to_str`*.
  - `COMPRESS()` computed the maximum result length in an arbitrary fashion. Now we use `compressBound` from the `zlib` library instead.

  (Bug #32922688, Bug #33117410, Bug #33275424)

References: See also: Bug #33516898.

## Keyring Notes

- Diagnostics for `keyring_hashicorp` plugin configuration issues have been improved. (Bug #32075854)

## Optimizer Notes

- `EXPLAIN FORMAT=TREE` now shows more precise information than displayed previously about scans generated by the range optimizer. In particular, sub-iterators are now displayed explicitly, and are properly timed with `EXPLAIN ANALYZE`; index range scans now show the actual ranges being scanned. Descriptions in the output are also more user-friendly than before; for example, `index_for_group_by` shown for a query using `DISTINCT` is replaced by `index skip scan for deduplication`.

  In addition, a roundoff error causing inaccuracies in row count estimation for read over range intersection scans has been corrected, and optimizer traces for index range scans now correctly displays implicit key parts from `InnoDB` primary keys when they are used. (Bug #33037007, Bug #33062448)

- When transforming `EXISTS` to a semijoin, and when the query contained a view reference, the query was not processed correctly. (Bug #32813550)

  References: This issue is a regression of: Bug #30671329.

- In the case of a lateral derived table, if the creation of the cache invalidator was delayed, the table materialization was emitted without the invalidator, which kept rematerialization from occurring during execution and led to wrong results.

  The pending cache invalidator was emitted only when the index of the lateral table was less than that of the last table in the table list being considered. When the table index of the pending invalidator was equal to the last table of the join slice, the cache invalidator was skipped and the materialization was emitted without the invalidator.

  We fix this by creating the pending cache invalidator if the table index of the pending invalidator is less than or equal to that of the last table in the table list of the current join slice. (Bug #32407774)

## Performance Schema Notes

- To assist monitoring and troubleshooting, the Performance Schema instrumentation is now used to export names of instrumented threads to the operating system. This enables utilities that display thread names, such as debuggers and the Unix `ps` command, to display distinct `mysqld` thread names rather than “mysqld”. This feature is supported only on Linux, macOS, and Windows. For more information, see The setup_instruments Table. (WL #14587)

## Pluggable Authentication

- **Microsoft Windows:** The Kerberos authentication method added in MySQL 8.0.26 for MySQL server and client hosts running Linux is now supported on the client side for Windows. This enables MySQL client applications running on Windows to connect to MySQL accounts on Linux server hosts that authenticate using Kerberos. For details, see Kerberos Pluggable Authentication. (WL #14605)

## Security Notes

- For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.1.1l. Issues fixed in the new OpenSSL version are described at

https://www.openssl.org/news/cl111.txt and at http://www.openssl.org/news/vulnerabilities.html. (Bug #33273138, Bug #33309871)

## Server Administration

- Setting the session value of the following system variables is now a restricted operation and the session user must have privileges sufficient to set restricted session variables:

  - `low_priority_updates`
  - `max_delayed_threads`
  - `max_error_count`
  - `min_examined_row_limit`
  - `preload_buffer_size`
  - `select_into_buffer_size`
  - `select_into_disk_sync_delay`
  - `show_old_temporals`

  For information about the privileges required to set restricted session variables, see System Variable Privileges. (WL #14695)

## Spatial Data Support

- The `ST_SymDifference()` and `ST_Intersection()` functions now permit the geometry arguments to have a geographic spatial reference system (SRS). Previously, `ST_SymDifference()` and `ST_Intersection()` supported only geometry arguments in a Cartesian SRS. See Spatial Operator Functions. (WL #10996, WL #14273)

## Functionality Added or Changed

- **Important Change; Group Replication:** The system variable `group_replication_components_stop_timeout` specifies the time that Group Replication waits for each of its modules to complete ongoing processes while shutting down. The component timeout applies after a `STOP GROUP_REPLICATION` statement is issued, which happens automatically during server restart or auto-rejoin. The timeout is used to resolve situations in which Group Replication components cannot be stopped normally, which might happen if the member is expelled from the group while it is in an error state, or while a process such as MySQL Enterprise Backup holds a global lock on tables on the member. In such situations, the member cannot stop the applier thread or complete the distributed recovery process to rejoin. The `STOP GROUP_REPLICATION` statement does not complete until either the situation is resolved (for example, by the lock being released), or the component timeout expires and the modules are shut down regardless of their status.

  Previously, the timeout value defaulted to 31536000 seconds (365 days), which did not help in situations such as those just described. The new default value is 300 seconds, so that Group Replication components are stopped after 5 minutes if the situation is not resolved before that time, allowing the member to be restarted and to rejoin. (WL #14245)

  References: See also: Bug #31460690, Bug #31648211, Bug #32309647.

- **Replication:** When GTID-based replication is in use on a replica server, the replication applier and receiver threads still track and have some dependencies on binary log file names and file positions, as used for the alternative binary log file position based replication. A new option for the `CHANGE REPLICATION SOURCE TO` statement, `GTID_ONLY`, removes the persistence of file names and file positions from the replication metadata repositories. For replication channels with this setting, in-memory file positions are still tracked, and file positions can still be observed for debugging purposes in error messages and through interfaces such as `SHOW REPLICA STATUS` statements (where they are shown as being invalid if they are out of date). However, the writes and reads required to persist and check the file positions are avoided in situations where GTID-based replication does not actually require them, including the transaction queuing and application process. The GTID_ONLY setting also means that the replication metadata is flushed less frequently.

  The `GTID_ONLY` option is disabled by default for asynchronous replication channels, but it is enabled by default for group replication channels, and it cannot be disabled for them. To set `GTID_ONLY = 1` for a replication channel, GTIDs must be in use on the server (`gtid_mode = ON`), and row-based binary logging must be in use on the source (statement-based replication is not supported). The `CHANGE REPLICATION SOURCE TO` options `REQUIRE_ROW_FORMAT` and `SOURCE_AUTO_POSITION` must each be set to 1 for the replication channel. When `GTID_ONLY` is set to 1, the replica uses `replica_parallel_workers=1` if that system variable is set to zero for the server, so it is always technically a multi-threaded applier. This is because a multi-threaded applier uses saved positions rather than the replication metadata repositories to locate the start of a transaction that it needs to reapply.

  Thanks to Facebook for offering a contribution related to this issue. (Bug #94360, Bug #29364334, WL #7491)

- **Replication:** Multithreading is now enabled by default for replica servers. A multithreaded applier has a number of applier threads that execute transactions in parallel. This behavior can avoid many cases of unwanted replication lag that can cause temporary divergence between the source and replicas.

  The following default server settings are used to produce the multithreading behavior:

  - `replica_parallel_workers=4`. This setting enables multithreading and creates four applier threads on the replica, plus a coordinator thread to manage them. If you are using multiple replication channels, each channel has this number of threads. Four applier threads provide a base level of parallelism, and you can change the setting to specify up to 1024 applier threads.

  - `replica_preserve_commit_order=1`. This setting ensures that transactions are externalized on the replica in the same order as they appear in the replica's relay log, so the replica never enters a state that the master was not in, and there are no gaps in the sequence of transactions that have been executed from the relay log.

  - `replica_parallel_type=LOGICAL_CLOCK`. This setting specifies that transactions that are part of the same binary log group commit on a replication source server are applied in parallel on a replica. It is required when `replica_preserve_commit_order=1` is set.

  To override the new defaults and disable multithreading for a replica server, specify `replica_parallel_workers=0`. This setting disables parallel execution and gives the replica a single applier thread and no coordinator thread. When you apply this setting, the `replica_parallel_type` and `replica_preserve_commit_order` options have no effect and are ignored. (WL #10475)

- **Group Replication:** The asynchronous connection failover mechanism for MySQL replication now enables a replica that is part of a managed replication group to automatically reconnect to the sender if the current receiver (the primary of the group) fails. The new feature works with Group Replication, on a group configured in single-primary mode, where the group's primary is a

replica that has a replication channel with `SOURCE_CONNECTION_AUTO_FAILOVER` set to `ON`. The feature operates by default on a group in this situation, although you can disable it for the group by disabling the new member action `mysql_start_failover_channels_if_primary`, using the `group_replication_disable_member_action()` function. The feature is designed for a group of senders and a group of receivers to keep synchronized with each other even when some members are temporarily unavailable. It also synchronizes a group of receivers with one or more senders that are not part of a managed group. A replica that is not part of a replication group cannot use this feature.

To configure this feature, the replication channel and the replication user account and password for the channel must be set up on all the member servers in the replication group, and on any new joining members. You can do this using the `CHANGE REPLICATION SOURCE TO` statement, or if the new servers are provisioned using MySQL's clone functionality, this all happens automatically. The `SOURCE_CONNECTION_AUTO_FAILOVER` setting for the channel is broadcast to group members from the primary when they join, and also if it is changed. The source list is broadcast to all members when they join or when it is updated. If the primary goes offline or into an error state, the new primary that is selected for the group has the source list and the channel configuration already in place, and establishes a replacement asynchronous replication connection with the source.

A new function `asynchronous_connection_failover_reset()` is also provided for administrators to remove all settings relating to the asynchronous connection failover mechanism. Use this function to clean up a server that is no longer being used in a managed group. (WL #14020)

- **Group Replication:** The group communication engine for Group Replication (XCom, a Paxos variant) defaults to using every member of the group as a leader. When the Group Replication communication protocol version is set to 8.0.27 or later, the group communication engine can now use a single leader to drive consensus when the group is in single-primary mode. Operating with a single consensus leader improves performance and resilience in single-primary mode, particularly when some of the group's secondary members are currently unreachable.

  The single consensus leader is colocated with the group's primary, and changes when a new primary is elected. The Performance Schema table `replication_group_communication_information` shows the preferred and actual consensus leader, or leaders if all members are used as a leader, the communication protocol version, and the write concurrency.

  To enable the new behavior, set the system variable `group_replication_paxos_single_leader` to `ON` (the default is `OFF`). When Group Replication is running in multi-primary mode, or with earlier communication protocol versions, or when `group_replication_paxos_single_leader` is set to `OFF`, the group communication engine operates using every member of the group as a leader.

  Note that when you manually upgrade the members of a replication group to a new MySQL Server release, the group's communication protocol version is not automatically upgraded to match. If you no longer need to support members at earlier releases, you can use the `group_replication_set_communication_protocol()` function to set the communication protocol version to the new MySQL Server version to which you have upgraded the members. MySQL InnoDB Cluster manages the communication protocol version automatically for replication groups created using that function. (WL #9149)

- **Group Replication:** In previous releases, Group Replication secured group communication connections and distributed recovery connections between members using its own implementation of the security protocols, including TLS/SSL and the use of an allowlist for incoming Group Communication System (GCS) connections. Replication groups can now use the MySQL Server's own connection security in place of the Group Replication implementation. Using the MySQL protocol means that standard methods of user authentication can be used for granting (or revoking) access to the group in place of the

allowlist, and that the latest functionality of the server's protocol is always available on release. Network namespaces are supported for Group Replication when the MySQL communication stack is used.

To use the MySQL Server's implementation of the connection security management in place of the Group Replication implementation, set the new system variable `group_replication_communication_stack` to `MYSQL`. In addition, the network address set by `group_replication_local_address` for each group member must be changed to one of the IP addresses and ports which MySQL Server is listening on, as specified by `bind_address`. If a network namespace is used, this must be configured using the `CHANGE REPLICATION SOURCE TO` statement in the `group_replication_recovery` channel.

Authentication is carried out using the existing replication user account that Group Replication uses for distributed recovery, as set using `CHANGE REPLICATION SOURCE TO`, and this user must be given the new `GROUP_REPLICATION_STREAM` privilege. The TLS/SSL configuration for the connection is taken from Group Replication's existing settings for securing distributed recovery, plus the `group_replication_ssl_mode` system variable that specifies whether TLS/ SSL is enabled or disabled for group communications. These settings must be configured if they are not already in place. All these settings must be the same on all group members to avoid communication issues.

As part of this work, the default value for the `performance_schema_max_cond_classes` system variable is increased from 100 to 150.

See Group Replication Requirements For The MySQL Communication Stack for more details. (WL #9852)

- Programs that encounter issues while processing `include` or `includedir` directives in option files now produce error messages that are more informative about the cause of the errors. (Bug #32798288, Bug #103397)

- The default value for the `thread_stack` system variable has been increased to 1048576 on all supported platforms. (Bug #103912, Bug #32965326)

  References: See also: Bug #32934187.

- A default time zone can now be set for a server by using the server option `--default-time-zone` while starting a MySQL Server Docker container. Before, the container failed to start if the option was used. (WL #14703)

- For online DDL operations, storage is usually the bottleneck. To address this issue, CPU utilization and index building has been improved. Indexes can now be built simultaneously instead of serially. Memory management has also been tightened to respect memory configuration limits set by the user. See Configuring Parallel Threads for Online DDL Operations.

  The new `innodb_ddl_threads` variable defines the maximum number of parallel threads for the sort and build phases of index creation.

  The new `innodb_ddl_buffer_size` variable defines the maximum buffer size for DDL operations. The default setting is 1048576 bytes (approximately 1 MB). Defining a buffer size limit avoids potential out of memory errors for online DDL operations that create or rebuild secondary indexes. See Online DDL Memory Management. (WL #14283)

- The clone plugin now permits concurrent DDL operations on the donor MySQL Server instance while a cloning operation is in progress. Previously, a backup lock was held during the cloning operation, preventing concurrent DDL on the donor. To revert to the previous behavior of blocking concurrent DDL on the donor during a clone operation, enable the `clone_block_ddl` variable. See Cloning and Concurrent DDL. (WL #9683)

- Setting a session value for the `internal_tmp_mem_storage_engine` variable now requires the `SESSION_VARIABLES_ADMIN` or `SYSTEM_VARIABLES_ADMIN` privilege. (WL #14728)

## Bugs Fixed

- **Incompatible Change:** For all `SELECT` statements on a view, the query digest was based on the view definition. As a result, different queries had the same digest and aggregated together in the Performance Schema table `events_statements_summary_by_digest`, so statistics in that table were not usable for distinguishing distinct `SELECT` statements.

  The query digest for each `SELECT` statement on a view now is based on the `SELECT`, not the view definition. This enables distinguishing distinct `SELECT` statements in the `events_statements_summary_by_digest` table. However, tools that use query digests may need some adjustment to account for this change. For example, MySQL Enterprise Firewall and query rewrite plugins rely on query digests and existing rules for them that are associated with views may need to be updated. (Bug #27540213, Bug #89559, Bug #31761802)

- **Important Change:** `EXPLAIN FORMAT=TREE` now shows whether an index scan uses a covering index, and thus does not need to look up other columns from the table/clustered index. For example, if `idx1` is a covering index, the old output `Index scan on t1 using idx1` is now shown as `Covering index scan on t1 using idx1`. Previously, this information was shown only for `FORMAT=TRADITIONAL` and `FORMAT=JSON`.

  This fix also improves the wording used for full-text search to align with this change. For example, the old output `Indexed full text search on t1` (which was the same in both the covering and non-covering cases) is now `Full-text index search on t1` when there is no covering index, and `Full-text covering index search on t1` when a covering index is used. (Bug #32825235)

- **InnoDB:** An excessive number of notes were written to the error log when the `innodb_open_files` limit was temporarily exceeded. (Bug #33343690)

- **InnoDB:** An in-place DDL operation failed to flush all modified pages. (Bug #33290335, Bug #33238133)

- **InnoDB:** A parallel scan returned an incorrect partition ID when loading data into MySQL HeatWave from a subpartitioned `InnoDB` table. (Bug #33276021)

- **InnoDB:** The unused `os_event::event_iter` field in the `InnoDB` sources was removed to reduce memory use in the `os_event` structure.

  Our thanks to Facebook for the contribution. (Bug #33252468)

- **InnoDB:** The `srv_purge_thread` and `srv_worker_thread` threads were duplicated in the `performance_schema.threads` table.

  Thanks to Kaige Ye for the contribution. (Bug #33209066, Bug #104575)

- **InnoDB:** Truncation of an undo tablespace during use by an active transaction raised an assertion failure. The transaction was prematurely marked as complete, permitting the truncation operation. (Bug #33162828)

- **InnoDB:** When loading data into MySQL HeatWave from a partitioned table with concurrent DML modifying the primary key, the partition ID reported in the load callback was found to be incorrect for some records. (Bug #33139692)

- **InnoDB:** Instances of `MY_ATTRIBUTE((noreturn))` and `MY_ATTRIBUTE((unused))` in the `InnoDB` sources were replaced by C++17 `[[noreturn]]` and `[[maybe_unused]]` attributes. (Bug #33112971)

- **InnoDB:** Each buffer pool block includes a `block->lock_hash_val` field. Caching of this value was determined to be unnecessary, as it introduced unnecessary coupling of the buffer and lock system and unnecessary memory usage. (Bug #33072415)

- **InnoDB:** A query that performed an index merge with retrieval ordered by row ID raised an assertion failure. The record buffer set up for the index merge could not be used due to the scanned table containing a primary key with a BLOB component. A record buffer cannot be used for reading BLOBs, which are stored outside of the record. The BLOB primary key was not detected when the record buffer was set up, as the primary key column was not yet in the read set. Retrievals ordered by row ID temporarily add the primary key at a later stage when needed. To address this issue, a record buffer is no longer requested for row-ordered retrievals if the primary key has a BLOB component. (Bug #33067554)

- **InnoDB:** Deleting or updating a row from a parent table initiated a cascading `SET NULL` operation on the child table that set a virtual column value to NULL. The virtual column value should have been derived from the base column value.

  Thanks to Yin Peng at Tencent for the contribution. (Bug #33053297)

- **InnoDB:** On a system that was nearing disk capacity, an `InnoDB` recovery operation involving application of file extension redo log records (`MLOG_FILE_EXTEND`) could cause a failure. (Bug #33002492)

- **InnoDB:** Conflicting explicit locks granted on the same record raised an assertion failure. (Bug #33000142)

- **InnoDB:** Freeing the first page of LOB at the end of purge batch raised an assertion failure. The failure was due an invalid root page number. (Bug #32958624)

- **InnoDB:** To facilitate failure reporting and resolution, the `ib::fatal()` function in the `InnoDB` sources was revised to include the caller's location. (Bug #32957311)

- **InnoDB:** Recovery on the clone recipient server failed with the following error: `Error reading encryption for innodb_undo_007`. The encryption key was not written to encrypted spaces created during the page copy phase of the clone operation. (Bug #32950216)

- **InnoDB:** To avoid generating unwanted warning messages, the `fil_space_acquire()` function in the `InnoDB` sources was replaced by the `fil_space_acquire_silent()` function where possible. Both functions are used by background threads to acquire a tablespace. (Bug #32944543)

- **InnoDB:** `InnoDB` CRC32 checksum algorithm implementations have now been optimized for use with ARM and x86/x64 architectures. (Bug #32887066)

- **InnoDB:** Startup on an instance with thousands of tables took an excessive amount of time due a large amount of traffic on the error logging subsystem. (Bug #32846656)

- **InnoDB:** The `INFORMATION_SCHEMA.FILES` view did not show the current path of the temporary tablespace file, and the file name shown was different from the one defined by the `innodb_temp_data_file_path` variable. (Bug #32840635, Bug #103553)

- **InnoDB:** On Windows, keeping a file open without a shared write lock while attempting to acquire the `fil_shard` mutex caused a deadlock with another thread that had acquired the `fil_shard` mutex and was attempting to access the same file. (Bug #32808809)

- **InnoDB:** Starting a MySQL Server instance using the same `InnoDB` data files as an another running MySQL Server instance resulted in an initialization failure. (Bug #32777654, Bug #103338)

- **InnoDB:** The `InnoDB` recovery process did not recognize that page compression had been applied to data that was being recovered, causing the tablespace data file to increase in size during the redo log apply phase, which could lead to a recovery failure for systems approaching a disk-full state. (Bug #32771259)

- **InnoDB:** An assert that traversed a list of file segments which were not full to calculate the number of used pages for comparison with the number of used pages tracked by a field in the file segment inode failed sporadically. (Bug #31685095)

- **InnoDB:** A transaction failed to roll back when the server was restarted after failure occurred during an online DDL operation. Table locks could not be resurrected for the uncommitted transaction and the data dictionary table object could not be loaded for the affected table.

  Thanks to Shaohua Wang for the contribution. (Bug #31131530, Bug #99174)

- **InnoDB:** A query that used a temporary table for aggregation exhausted the memory available to the `TempTable` storage engine, causing an update operation to fail with a table is full error. (Bug #31117893, Bug #99100)

- **Replication:** The replication applier (SQL) thread overrode retryable errors (such as deadlocks and wait timeouts) from storage engines with a `key not found` error, causing replication to stop without retrying the transaction. These errors are no longer overridden. (Bug #33107663)

- **Replication:** When the Group Replication distributed recovery process synchronized a joining member with the donor, the Performance Schema table `replication_group_member_stats` table was not updated with the current number of transactions queued on the `group_replication_applier` channel (the `COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE` field.) The count is now tracked while new transactions are arriving during distributed recovery, although it remains zero until the joiner has certified the transactions that were received in the first phase of distributed recovery. (Bug #33067441)

- **Replication:** For multithreaded replicas (replicas on which `replica_parallel_workers` is set to a value greater than 0), if `replica_parallel_workers` is set to 1, the setting for `replica_preserve_commit_order` is now ignored. When there is a single applier, transactions are always executed and committed in the same order as in the replica's relay log; ignoring the process to preserve the commit order avoids potential performance degradation. (Bug #33048169)

- **Replication:** An assertion was raised if a replica MySQL Server instance with unpopulated time zones attempted to replicate a statement that set a time zone value that was unknown to the replica. Replicas now handle this situation correctly. (Bug #32986721)

- **Replication:** The error messages issued by MySQL Replication when GTIDs required for auto-positioning have been purged could be incorrectly assigned or scrambled in some situations. (Bug #32965864)

- **Replication:** If a group member was elected as the primary right before or while it was shutting down, the shutdown process hung while waiting on the primary election process, which was attempting to make the server leave the group since the election had failed due to the shutdown. The error handling process for primary elections now takes this into account, and does not take any further actions if the member is already leaving the group. (Bug #32884709)

- **Replication:** Querying the Performance Schema table `replication_asynchronous_connection_failover` could return an error if a row was deleted during the query process. In this situation, the row count is now returned as zero, and the query can be retried. (Bug #32701593)

- **Replication:** In some situations, a replica that used connection compression was not able to re-establish a lost connection to the source server. The issue has now been fixed. (Bug #32494609)

- **Group Replication:** During the Group Replication auto-rejoin procedure, a group member sets its status to `RECOVERING`. If the group member does not manage to rejoin, it should change the status to `ERROR`, but if a view change occurred in the meantime, it was possible for the status to remain in `RECOVERING`. The member status is now set to `ERROR` after an unsuccessful auto-rejoin procedure, regardless of any ongoing or stuck view changes. (Bug #33276418)

- **Group Replication:** Garbage collection for certification information has been moved from the Group Replication Group Communication System (GCS) thread to a background thread, so that sending and receiving of messages are not blocked while garbage collection is in progress. (Bug #33190276)

- **Group Replication:** When `group_replication_consistency` is `BEFORE_ON_PRIMARY_FAILOVER`, in the event of a primary failover, client connections are held until the new primary has the same state as the previous primary. Some monitoring and administration statements are exempt from this hold, so that the new primary can be inspected during the failover process.

  Previously, `DO` statements had a blanket exemption from this hold, but now only `DO` statements that do not use tables or loadable functions are exempt; this is the same as for `SELECT`. (Bug #33130768)

- **Group Replication:** MySQL Server incorrectly permitted reads from Performance Schema tables relating to Group Replication while Group Replication was stopping or restarting, and the data concerned should not have been used. The server now checks whether Group Replication is in `OFFLINE` status or uninitialized before executing the query. (Bug #33085494)

- **Group Replication:** When `group_replication_consistency` is set to `BEFORE_ON_PRIMARY_FAILOVER`, in the event of a primary failover, client connections are held until the new primary has the same state as the previous primary. Some monitoring and administration statements are exempt from this hold, so that the new primary can be inspected during the failover process. Previously, `SHOW` statements had a blanket exemption from the hold (with the exception of `SHOW CREATE USER`), but now only `SHOW` statements that do not depend on data (only on status or configuration) are exempt. See Configuring Transaction Consistency Guarantees, for a complete listing of exempt `SHOW` statements. (Bug #33082509)

- **Group Replication:** A deadlock could occur when a statement referencing Access Control Lists (ACLs), such as `CREATE USER`, was executed on the primary of a Group Replication group, and a member joined the group immediately afterwards before the transaction commit was confirmed by the other group members. The distributed recovery process needs a read lock on the ACL cache which is locked by the ACL statement. This situation blocked Group Replication's Group Communication System (GCS) thread until the ACL statement timed out, making the primary unreachable and possibly preventing the new member from joining. The ACL cache lock is now no longer required for the distributed recovery process, although the lock in the situation described is only released after the view change is complete and the ACL statement is committed. Any new connections or statements that require the ACL cache lock, including a member join when Group Replication uses the MySQL communication stack, must therefore wait on this or fail and retry. (Bug #33025231)

- **Group Replication:** If the thread that runs the Group Replication applier module is stopped, the group cannot function properly because it cannot exchange group transactions and messages. Previously, a member in this situation remained in `ONLINE` status and ignored the internal errors. The member now changes to `ERROR` status if the thread is stopped, and takes the action specified by `group_replication_exit_state_action`. (Bug #32934479)

- **Group Replication:** In MySQL 8.0.22 and later, a replication source server writes a `TRUNCATE TABLE` statement to the binary log to notify replicas to empty a `MEMORY` table the first time it is used after a

server restart. Previously, the thread where the statement was logged was not registered with the global thread manager, so Group Replication was not able to acknowledge it. The issue has now been corrected. (Bug #32355801)

- **JSON:** Made additional improvements in JSON function error handling to those made in MySQL 8.0.23. (Bug #32864910)

  References: See also: Bug #31856260.

- **JSON:** `JSON_TABLE()` allowed duplicate column names when the names differed in case only, although column names are case-insensitive in MySQL.

  Now this function compares column names in case-insensitive fashion. (Bug #102824, Bug #32591074)

- Added Ubuntu 21.10 packages. (Bug #33501583, Bug #105274)

- The MySQL client library could contribute to a memory leak if MySQL was linked against OpenSSL 1.0.1, as is the case for builds on EL6. (Bug #33335046)

- Implicitly grouped queries sometimes calculate aggregates during optimization when their values can be easily retrieved from indexes. When a predicate referenced a column that was declared with a `NO PAD` collation, that predicate might be evaluated using `PAD SPACE` semantics, and so return wrong results. This was because an internal function that checked for insignificant trailing spaces made the assumption that all nonbinary collations had `PAD SPACE` semantics, which was true of MySQL 5.7, but is not the case for MySQL 8.0, which has added many collations having `NO PAD` semantics, including the default collation (`utf8mb4_0900_ai_ci`).

  We fix this by explicitly checking the padding attribute of the collation in such cases. (Bug #33282123)

- A query containing a common table expression with a `MATCH() AGAINST()` clause executed on a table defined without a full-text index raised an assertion failure. (Bug #33264864)

- Several Performance Schema tables contained default timestamp values of 0 (zero) which conflicted with the default `sql_mode` values `NO_ZERO_IN_DATE` and `NO_ZERO_DATE`.

  For example, attempting to create a new table based on such a Performance Schema table resulted in an error similar to the following: `ERROR 1067 (42000): Invalid default value for 'FIRST_SEEN'`

  Default timestamp values have been removed from the following tables:

  - `performance_schema.events_errors_summary_by_account_by_error`
  - `performance_schema.events_errors_summary_by_host_by_error`
  - `performance_schema.events_errors_summary_by_thread_by_error`
  - `performance_schema.events_errors_summary_by_user_by_error`
  - `performance_schema.events_errors_summary_global_by_error`
  - `performance_schema.events_statements_summary_by_digest`
  - `performance_schema.host_cache`
  - `performance_schema.replication_applier_filters`
  - `performance_schema.replication_applier_global_filters`

(Bug #33240123, Bug #104643)

- A failed write to the `NOTIFY_SOCKET` environment variable caused a failure. The `ER_SYSTEMD_NOTIFY_WRITE_FAILED` error associated with the failed write has two parameters, but only one parameter was passed to the error logging routine. (Bug #33239183)
- An incorrectly type-casted variable was used when setting the `--ssl-fips-mode` option. (Bug #33223230)
- The following threads were not present in the `performance_schema.threads` table:
  - `buf_resize_thread`
  - `fts_optimize_thread`

  Thanks to Kaige Ye for the contribution.

  Thanks to Kaige Ye for the contribution. (Bug #33214130, Bug #104582, Bug #33214136, Bug #104583)
- A recursive call to an internal save function led to an unexpected error. (Bug #33198164)
- The internal `mysqld_list_fields()` function failed to remove temporary tables created to evaluate `JSON` table functions. (Bug #33177686)
- The code to produce minimal TAR packages added debug symbols to the packages, which caused larger (roughly by 10x) builds. Now DEB/RPM compiler flags are on by default for debug symbol builds, and off by default for minimal sized release builds. (Bug #33151629, Bug #104402)
- Some multi-table `DELETE` statements were found to leak memory. (Bug #33151275)

  References: See also: Bug #18684036.
- The return value for a copy function internal to the server was not handled as expected. (Bug #33142669)

  References: This issue is a regression of: Bug #31982292.
- Empty range frames were not always handled correctly. (Bug #33142418)

  References: This issue is a regression of: Bug #90300, Bug #27808099.
- In debug builds, the `ALTER TABLE` statement could produce an error if it added a new virtual column with the same name as one of the columns later referred to by a foreign key. This fix now ignores a virtual column if the name is duplicated and instead uses existing, non-virtual column names to check conditions. (Bug #33114045)
- An assert condition to ensure that execution of a stored program instruction is started when there are no errors did not work properly for a `CASE` statement in a loop. (Bug #33079184)
- In debug builds, `ANALYZE TABLE` with the `UPDATE HISTOGRAM` clause could return a non-success value to the caller, instead of a success value, after successfully clearing the diagnostics area. (Bug #33079073)
- The `mecab_charset` system status variable now reports its value as `utf8mb4` rather than `utf8`, which is deprecated. (Bug #33078623)
- In debug builds, MySQL Enterprise Encryption UDFs did not set the nullable flag when returning NULL. (Bug #33077931)

- The range optimizer was sometimes called when a plan lock was in force. This caused issues since the range optimizer can call itself, but a plan lock does not allow for recursion. (Bug #33076462)

  References: This issue is a regression of: Bug #18684036.

- `CAST()` and `DEFAULT()`, when used inside stored routines, were not always handled correctly. (Bug #33075828)

- String functions that use temporary string buffers during evaluation could lead to unexpected shutdowns. (Bug #33073951)

- The error message emitted after a host name failed to resolve to an IP address did not include a meaningful `errno` value. Now, `(-2)` indicating `EAI_NONAME` is returned in the message instead of `(0)`. (Bug #33064143)

- A statement such as `CREATE TABLE t SELECT 1` created an `InnoDB` table that was written incorrectly to the binary log if the value of `binlog_format` was set to `ROW` and `sql_mode` was in ANSI mode. As a result, replication of the statement failed with an error on the replica. Applying the `mysqlbinlog` utility to such a binary log could also fail.

  The atomic `CREATE...SELECT` was implemented by adding a new clause to `CREATE TABLE` called `START TRANSACTION`. However, this clause was not added when ANSI mode was enabled. This in turn caused the execution of an ordinary implicitly committed `CREATE TABLE` in the middle of the transaction and produced an error in GTID mode if the transaction had an assigned GTID. The issue is fixed by removing the SQL mode dependency from the new clause. (Bug #33064062, Bug #104153)

- A log file containing a malformed ISO8601 timestamp was processed incorrectly. (Bug #33060440)

- String conversion warnings that previously referred to `utf8` now reference `utf8mb3` instead. (Bug #33059330)

- When building MySQL from source on Unix platforms, `.bz2` files are now used for Boost archive downloads rather than `.tar.gz` files. (Bug #33052171)

- For Enterprise Linux 8 (and Fedora), fixed the debuginfo RPMS packages by disabling REPRODUCIBLE_BUILD in fprofile.cmake. (Bug #33037380)

- For a query with rollup, when setting an expression as nullable because it had a grouping column, we missed setting all expressions within that expression as nullable, doing so only for the topmost expression. This meant that, during evaluation, a `NULL` generated by rollup was not always propagated correctly. To fix this, we now set all the expressions having a grouping column as nullable when the query uses rollup. (Bug #33036184)

- During execution of `EXPLAIN`, when crossing into a different query block through a streaming or materialization node, this node was counted as the root, rather than the actual root node. (Bug #33030136)

- Fixed an undefined conversion from double to `int64` in `sql/join_optimizer/cost_model.cc`. (Bug #33024410)

- The internal function `find_in_group_list()` did not match up match up all items correctly during `ROLLUP` processing. We fix this by adding casts to `GROUP BY` expressions. (Bug #33022742, Bug #33123934)

  References: This issue is a regression of: Bug #30969045.

- A missing test for success of a memory allocation in the MySQL client library could lead to a client exit. (Bug #33019026)

- An audit log function call from a prepared statement caused an error. (Bug #33016004)
- Avoid adding column names prefixed with `!hidden!` to ensure that new names do not collide with names used by existing hidden columns for functional indexes. Generated hidden column names now have the following new form that extends the use of functional indexes into environments that do not support names generated by `MD5()`:

  `!hidden!`*`index_name`*`!`*`key_part_number`*`!`*`counter`*

  The *`counter`* value of a generated name is zero unless a column with that name already exists in the table. In this case, the value is incremented until the name becomes unique. (Bug #32983024)
- Removed an unnecessary hard-coded dependency on the range optimizer from `sql_help.cc`. (Bug #32976042)
- Insufficient buffer space allocation during window function execution could cause an assertion to be raised. (Bug #32975889)
- When finding the list of tables under a hash join, we did not take into account those that were also hidden under ZERO_ROWS iterators. This could lead to NULL row flags not being set correctly, which also caused problems when weedout wanted to save row IDs for them. (Bug #32975168)
- The `gen_dictionary()`, `gen_range()`, and `gen_rnd_pan()` data masking functions each could generate the same value if executed in close temporal proximity multiple times. (Bug #32970772)
- Creation and deletion of temporary tables used in resolution of common table expressions and having table references created within subqueries were not always managed correctly. (Bug #32962511)
- When the `--binary-as-hex` option is enabled for the `mysql` client, empty strings are now printed as `0x` instead of `NULL`. (Bug #32961656, Bug #103906)
- The resolver usually terminates the analysis and exits after encountering an error in a statement. In the case of duplicate column analysis, the resolver continued to the end of the column list, possibly adding multiple error messages to the diagnostics object. (Bug #32960158)
- When a scalar subquery returned multiple rows, the resulting error was not always handled correctly. (Bug #32956779)
- Changing the server SQL mode after creating a table containing generated columns could cause spurious messages to be written to the error log. (Bug #32954466)
- Manifest file reading could fail on Windows. (Bug #32950322)
- Evaluation of the values in an `IN()` list did not stop immediately on error, which led to assert failures. We fix this by stopping evaluation in such cases as soon as an error has been raised. (Bug #32942328)
- If an error was raised while evaluating a comparison of two non-nullable values as strings, the result of the comparison was set to `NULL`, even though the result was non-nullable according to the comparison operator metadata. The error was correctly returned to the user, but an assertion was raised by this inconsistency when running in debug mode.

  This is fixed by causing `Arg_comparator` not to set its owner to `NULL` when the owner is not nullable. (Bug #32942327)
- An unset variable referenced in an SQL script executed during an upgrade operation caused a failure. (Bug #32939819)
- Improper error propagation in `filesort` operations could raise an assertion. (Bug #32932969)

- In the internal `WalkAndReplace()` function, errors from `set_cmp_func()` were not correctly propagated. (Bug #32918927, Bug #33007298)

  References: This issue is a regression of: Bug #32548377.

- A deadlock could occur if a `RESET REPLICA ALL` statement was used while the channel configuration was being read. (Bug #32906709)

- A potential race condition in accessing the persisted variables cache has been eliminated. (Bug #32901419)

- The constant propagation performed by the MySQL optimizer could in some cases replace references to a column that was not nullable with a nullable expression. When this occurred, the parent item of the replaced column reference could sometimes have the wrong nullability, leading assert failures later, during execution, when a non-nullable item unexpectedly returned `NULL`.

  We fix this by skipping constant propagation in cases where a non-nullable column reference is replaced by a nullable expression. (Bug #32895824)

  References: This issue is a regression of: Bug #32371039.

- A column name provided in a query could differ in collation details, or because the name was provided as an expression alias in the query, and still match a column name in the dictionary. The query output contained the column name specified in the query (for example, `aaa`) rather than the column name from the dictionary (for example, `AAA`). (Bug #32892045)

- When the server SQL mode is other than strict mode, certain string functions return `NULL` to indicate that the result is too large for the result buffer, which could lead to in inconsistent behaviour such as incorrectly sorted output. In addition, the functions `LAST_INSERT_ID()` and `CAST(... AS CHAR)` did not maintain nullability properly for all cases. (Bug #32864958)

- Hidden items added as part of an `ORDER BY`, windowing function, or a reference to a view were not always handled correctly in implicitly grouped queries. (Bug #32863279, Bug #33079592)

- Type resolution for negation did not set the proper precision when converting the type from integer to decimal. This is fixed by assigning the same precision as the argument. (Bug #32863037)

  References: This issue is a regression of: Bug #31348202.

- Improper error propagation for failed `CREATE TABLE ... SELECT` statements caused rollback not to occur. (Bug #32855882)

- When used in a subquery, a `VALUES` having more than one `ROW()` was not always handled correctly. (Bug #32851684)

- The error packet that MySQL Server sends to a client program when the wait timeout expires (`ER_CLIENT_INTERACTION_TIMEOUT`) used an incorrect sequence number of 2 instead of 0 in the packet header when protocol compression was used. (Bug #32835205, Bug #103412)

- Concurrent insert operations on multiple tables with full-text indexes caused a large number of full-text index synchronization requests, resulting in an out of memory condition. (Bug #32831765, Bug #103523)

- The fix for a previous issue, following subsequent work which made it redundant and which led to invalid results from expressions used in window functions, has been reverted. (Bug #32820802)

  References: Reverted patches: Bug #26389508.

- In prepared statements, `NULLIF()` result type determination could be incorrect. (Bug #32816305, Bug #103458)

- Creating and dropping of views within stored routines were not always handled correctly. (Bug #32807430)

- The fix for a previous issue included a minor refactoring of how the precision and scale of a decimal expression were determined. It later emerged that, for the `TRUNCATE()` function, we might end up with a precision of zero, which is invalid.

  We fix this problem by treating a precision of zero as one. (Bug #32802251)

  References: See also: Bug #31348202.

- For legacy reasons, we can have composite access paths including `Filter` and `Sort` inside `table_path`. For ease of analysis and better formatting, we move the `EXPLAIN` output for these previous to the `Materialize` access path.

  We show here examples of an `EXPLAIN` statement run both prior to and following this change:

```
# Table created as follows:
mysql> DROP TABLE IF EXISTS t1;
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE t1 ( f1 INTEGER );
Query OK, 0 rows affected (0.03 sec)


# Previous to change:
mysql> EXPLAIN FORMAT=TREE
    ->   SELECT * FROM ( SELECT * FROM t1 LIMIT 2 OFFSET 1 ) AS alias1
    ->   WHERE f1 <= ANY ( SELECT f1 FROM t1 ) ORDER BY f1\G
*************************** 1. row ***************************
EXPLAIN: -> Sort: alias1.f1
-> Filter: <nop>((alias1.f1 <= (select #3)))  (cost=2.62 rows=2) [other sub-iterators not shown]
    -> Table scan on alias1  (cost=2.62 rows=2)
        -> Materialize  (cost=0.35..0.35 rows=0)
            -> Limit/Offset: 2/1 row(s)  (cost=0.35 rows=0)
                -> Table scan on t1  (cost=0.35 rows=1)

# Following change:
mysql> EXPLAIN FORMAT=TREE
    ->   SELECT * FROM ( SELECT * FROM t1 LIMIT 2 OFFSET 1 ) AS alias1
    ->   WHERE f1 <= ANY ( SELECT f1 FROM t1 ) ORDER BY f1\G
*************************** 1. row ***************************
EXPLAIN:
-> Sort: alias1.f1  (cost=0.35..0.35 rows=0)
    -> Filter: <nop>((alias1.f1 <= (select #3)))  (cost=2.62 rows=2)
        -> Table scan on alias1  (cost=2.62 rows=2)
            -> Materialize  (cost=0.35..0.35 rows=0)
                -> Limit/Offset: 2/1 row(s)  (cost=0.35 rows=0)
                    -> Table scan on t1  (cost=0.35 rows=1)
        -> Select #3 (subquery in condition; run only once)
            -> Aggregate: max(t1.f1)  (cost=0.45 rows=1)
                -> Table scan on t1  (cost=0.35 rows=1)
```

  After this change, the only legal access paths within `table_path` are `TABLE_SCAN`, `REF`, `REF_OR_NULL`, `EQ_REF`, and `ALTERNATIVE`. (Bug #32788576, Bug #32915233)

- Constant folding did not always handle errors correctly when evaluating decimal expressions. (Bug #32785804)

- A call order mismatch in `Query_block::prepare_values()` caused `setup_order()` to be called after `resolve_subquery()`, which meant that, for a `VALUES` clause that was a subquery, the subquery could be merged into the outer query block before calling `setup_order()`, leading to inconsistent data structures and an error.

We fix this issue by performing `setup_order()` earlier, and, if the column is not found, resolution is aborted. (Bug #32783943)

References: This issue is a regression of: Bug #31387510.

- In the Performance Schema table `variables.info`, the system variable `skip_slave_start` was incorrectly listed as `COMPILED` when the global value was actually loaded from the persisted variables file, so `PERSISTED` should have been used. (Bug #32640588)
- A `SELECT` query on the `INFORMATION_SCHEMA.PROCESSLIST` view with concurrent MySQL Server load caused a failure. (Bug #32625376)
- When a query uses a temporary table for aggregation, the group by item is used as a unique constraint on the temporary table: If the item value is already present, the row is updated; otherwise, a new row is inserted into the temporary table. If the item has a result field or reference item, it evaluated twice, once to check whether the result exists in the temporary table and, if not, again while constructing the row to be inserted. When the group by item was nondeterministic, the result value used to check for existence differed from that with which an insert was attempted, causing the insert to be rejected if the value already existed in the table.

  We fix this by using the hash of any nondeterministic items as the unique constraint, so that the hash is evaluated once only. (Bug #32552332)
- Privilege-checking for table-specific roles was in some contexts not restrictive enough. (Bug #32400788)
- Inconsistencies in how certain comparison predicates were evaluated (for example, when part of a `WHERE` clause) could return different results if a function was used instead of a string literal. (Bug #32345941, Bug #102151)
- Columns of type `ENUM` or `SET` are ordered based on numeric comparison, but the comparison function for range expressions (that is, expressions used for ordering in case of a range frame specification) of a window function is set based on the result type of the column, which for `ENUM` and `SET` is `String`. As a result, processing of rows for a window frame (to see whether a row is before or after the frame) did not work correctly; for example, a string comparison might determine that a row occurs before a frame, while a numeric comparison would have placed the row after.

  To fix this problem, we implement integer cache items for `ENUM` and `SET`, as well as integer comparison functions for use when `ENUM` or `SET` types are involved in range expressions. (Bug #32328576)
- A DML statement, when accessing a subquery which had been optimized away and cleaned up, led to an unplanned shutdown of the server. (Bug #32244822)
- When resolving columns, their names are compared in case-insensitive fashion using `utf8_general_ci`, which does not always follow the same comparison rules as those for the collation actually used for the table. Previously, when a table had in excess of 32 columns, name lookup was performed using a hash table. Hashing is collation-aware, and so follows the collation's comparison rules; this caused name lookup and duplication detection to be done in an inconsistent fashion. We solve this problem by removing the hash, and performing column name resolution in the same way in all cases regardless of the number of columns. (Bug #32169656)
- For a nullable column, when adjacent ranges were rounded off to the same value by range optimizer, wrong results were returned. (Bug #31870920)

  References: See also: Bug #98826, Bug #30988735.
- Quote handling was improved for the `SHOW GRANTS` statement. (Bug #31716706)

- An attempt could be made to write a `JSON_TABLE()` expression to the optimizer trace before the temporary table backing the table function had been created, causing an assertion to be raised. Now when the column type is not yet available, `<column type not resolved yet>` is written. (Bug #31578783)

- Validity checks for `mandatory_roles` system variable settings are now synchronized with validity checks performed for `GRANT role` statements. (Bug #31218040)

- The `keyring_hashicorp_update_config()` function was not safe for concurrent execution. (Bug #31205028)

- The query rewrite plugin failed when refreshing the rewrite rules and the table holding the rewrite rules contained rows that had been marked as deleted, but not physically removed.

  We fix this by causing the query rewrite plugin to skip the deleted rows instead of failing when it sees them. (Bug #22654105)

- Refactoring done as part of implementing window functions in MySQL made it possible to refer to aliases of aggregates in `ORDER BY` clauses but also allowed direct references to such aggregates, even though this should not be allowed. Now the server checks explicitly for such illegal references. (Bug #13633829, Bug #30106081)

- In certain cases, the view reference cloned when pushing a condition down to a derived table was not always resolved in the desired context. In addition, a check for a null condition was not performed correctly. (Bug #104574, Bug #33209907, Bug #33197276)

- Some queries using `HAVING COUNT(DISTINCT ...)` did not return any rows when one was expected. (Bug #104411, Bug #33152269)

  References: This issue is a regression of: Bug #31790217.

- Multi-valued indexes were not used in the following cases:

  - In views

  - In prepared statements

  - In a `WHERE` containing `MEMBER OF()` combined using `OR` with another predicate

  In addition, MySQL wrongly reported `impossible condition` for a `WHERE` clause in the form *`f()`* `AND` *`f()`*, where *`f()`* was any of `MEMBER OF()`, `JSON_CONTAINS()`, or `JSON_OVERLAPS()`.

  Our thanks to Yubao Liu for the contribution. (Bug #104325, Bug #104700, Bug #104721, Bug #33123079, Bug #33268466, Bug #33275457)

  References: See also: Bug #102359, Bug #32427727. This issue is a regression of: Bug #30838807.

- When `NULL` was passed to a user-created function that called `REGEXP_INSTR()`, the first invocation of the function returned `NULL` as expected, but each subsequent invocation of the function also returned `NULL` without regard to the value passed to it. (Bug #104239, Bug #33089668)

- Some of the functions defined in `mbr_utils.cc` threw heap-allocated exceptions in some situations. Memory allocated for the exception object in these cases was never freed, which meant that a small amount of memory leaked each time an exception was thrown.

  This is fixed by allocating the exception on the stack in such cases, instead. (Bug #104214, Bug #33086286)

- Column names were not displayed correctly in the results of `ROLLUP` queries when the `subquery_to_derived` optimization was enabled. (Bug #104139, Bug #33057397, Bug #33104036)
- A stored procedure containing an `IF` statement using `EXISTS`, which acted on one or more tables that were deleted and recreated between executions, did not execute correctly for subsequent invocations following the first one. (Bug #103607, Bug #32855634)
- When executing a range query with multiple identical ranges joined by `OR` (for example, a query with `WHERE (a=1 AND b=2 AND c=3) OR (a=1 AND b=2 AND c=3)`), the optimizer lost part of the range, and so chose a query plan that was not optimal.

  Our thanks to Facebook for the contribution. (Bug #102634, Bug #32523520)
- While evaluating a loose index scan as a possible option for performing grouping and finding the minimum value, the cost calculation did not reflect the fact that the query looked at one group only, due to the equality predicates on the grouping attributes. This resulted in examination of additional rows since grouping is performed after reading the rows from the index.

  We fix this by determining whether a query produces only one group by checking for the presence of equality predicates on grouping attributes and using these for calculating the cost. This causes the optimizer to pick loose index scan for such cases when doing so is found to be beneficial. (Bug #101838, Bug #32266286)

  References: See also: Bug #18109609.
- When resolving integer division, the precision of the result is taken from the dividend. When the divisor is a decimal number, it may be less than 1, which may cause the result to use more digits than the dividend. This yielded incorrect values in some cases in which the result of integer division was a decimal or float. (Bug #100259, Bug #31641064)
- Added an in-memory estimate to the optimizer trace to indicate how much of a given table is buffered in the buffer pool.

  Our thanks to Øystein Grøvlen for the contribution. (Bug #99993, Bug #31544522)
- The `EXPLAIN` output for a DML statement contains the table identifier, which normally includes the database name, in the output of `SHOW WARNINGS`. For some statements such as `CREATE VIEW`, the database name should be omitted, which is enforced by setting the `alias_name_used` flag to true in the cached table object, but when the cached table was reused following `CREATE VIEW`, the flag was not reset, which caused the database name to be omitted from the warnings following `EXPLAIN` for statements run after a `CREATE VIEW` which access the same cached table as the view.

  We fix this by ensuring that the `alias_name_used` flag is always set to an appropriate value during table initialization.

  Our thanks to Kaiwang Chen for the contribution. (Bug #98635, Bug #30909064)

# Changes in MySQL 8.0.26 (2021-07-20)

- Audit Log Notes
- Authentication Notes
- Compilation Notes
- Component Notes
- Deprecation and Removal Notes

- Error Handling
- Event Scheduler Notes
- Firewall Notes
- Packaging Notes
- Pluggable Authentication
- Server Administration
- Spatial Data Support
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed

## Audit Log Notes

- Previously, each event logged by MySQL Enterprise Audit included the SQL statement literal text. To provide an alternative (because it is possible that statements contain sensitive information), the audit log filtering language now supports logging a statement's digest rather than its literal text. For example, instead of logging this statement:

  ```
  SELECT * FROM orders WHERE some_sensitive_column=1234567
  ```

  The audit log plugin can log this digest:

  ```
  SELECT * FROM `orders` WHERE `some_sensitive_column` = ?
  ```

  This is similar to what is already logged for prepared statements, for which parameter markers appear rather than actual data values.

  To perform digest logging, use audit filter definitions that replace the statement literal text by its corresponding digest, as discussed in Replacement of Event Field Values.

  Because text replacement occurs at an early auditing stage (during filtering), the choice of whether to log statement literal text or digest values applies regardless of log format written later (that is, whether the audit log plugin produces XML or JSON output). (Bug #31482609, WL #14267, WL #14724)

- For MySQL Enterprise Audit, the new `audit_log_format_unix_timestamp` system variable enables inclusion of a `time` field in each audit record. The field value is an integer that represents the UNIX timestamp value indicating the date and time when the audit event was generated. The `time` field is supported only for JSON-format log files. (WL #14600)

- For MySQL Enterprise Audit, the new `audit_log_max_size` system variable enables audit log file pruning based on combined log file size. To have an effect, `audit_log_max_size` requires that `audit_log_rotate_on_size` be greater than 0. If that is true, the pruning algorithm uses `audit_log_max_size` in conjunction with `audit_log_prune_seconds`, with nonzero values of `audit_log_max_size` taking precedence over nonzero values of `audit_log_prune_seconds`. For details, see Space Management of Audit Log Files. (WL #14525)

## Authentication Notes

- Previously, as part of the “hello” packet sent by the server to clients, the server sent the name of the server-side authentication plugin rather than the client-side plugin. The server now sends the client-side

name, which is more appropriate for the client's needs and may help to avoid extra protocol round trips. (WL #14308)

## Compilation Notes

- **macOS:** It is now possible to build MySQL for macOS 11 on ARM (that is, for Apple M1 systems). (Bug #32386050, Bug #102259)
- Building on openSUSE 15 and SLES 15 now requires GCC 9, found in packages `gcc-9` and `gcc9-c++`.

  Building on SLES 12 now requires GCC 10, found in packages `gcc-10` and `gcc10-c++`.

  It is also recommended to use the named GCC version when building third-party applications that are based on the `libmysqlclient` C API library. (Bug #32886268, Bug #32886439)
- Building on Ubuntu 18.04 (bionic) now requires GCC 8, found in packages `gcc-8` and `g++-8`. It is also recommended to use GCC 8 when building third-party applications that are based on the `libmysqlclient` C API library. (Bug #32877062)
- It is now possible to build MySQL on Solaris using GCC 10, which becomes the default and recommended compiler. It is also recommended to use GCC 10 when building third-party applications that are based on the `libmysqlclient` C API library. (Bug #32552988)

## Component Notes

- A new component service enables server components to set system variable values. For information about this service, see the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html (search for `s_mysql_mysql_system_variable_update_string` and `mysql_system_variable_update_string_imp`). (WL #12002)

## Deprecation and Removal Notes

- The TLSv1 and TLSv1.1 connection protocols now are deprecated and support for them is subject to removal in a future MySQL version. (For background, refer to the IETF memo Deprecating TLSv1.0 and TLSv1.1.) It is recommended that connections be made using the more-secure TLSv1.2 and TLSv1.3 protocols. TLSv1.3 requires that both the MySQL server and the client application be compiled with OpenSSL 1.1.1 or higher.

  On the server side, this deprecation has the following effects:

  - If the `tls_version` or `admin_tls_version` system variable is assigned a value containing a deprecated TLS protocol, the server produces a warning for each deprecated protocol:
    - If the assignment occurs during server startup, the warning appears in the error log.
    - If the assignment occurs at runtime, the warning is added to the result of executing the `ALTER INSTANCE RELOAD TLS` statement.
  - If a client successfully connects using a deprecated TLS protocol, the server writes a warning to the error log.

  On the client side, the deprecation has no visible effect. Clients do not issue a warning if configured to permit a deprecated TLS protocol. This includes:

  - Client programs that support a `--tls-version` option for specifying TLS protocols for connections to the MySQL server.

  - Statements that enable replicas to specify TLS protocols for connections to the source server. (`CHANGE REPLICATION SOURCE TO` has a `SOURCE_TLS_VERSION` option and `CHANGE MASTER TO` has a `MASTER_TLS_VERSION` option.)

  - The `group_replication_recovery_tls_version` system variable that enables joining members to specify TLS protocols for distributed recovery connections.

  (Bug #32565996, WL #14519)

- The system variable `transaction_write_set_extraction` is now deprecated, and a warning message is issued if you attempt to set it or read its value; expect it be removed in a future MySQL release. This system variable was used on a replication source server that has multithreaded replicas, to specify the algorithm used to hash the writes extracted for a transaction's write set. The `XXHASH64` algorithm, which is the default in MySQL 8.0 and is required for Group Replication, is selected when the system variable is not used. (WL #13950)

- The `temptable_use_mmap` variable is now deprecated and subject to removal in a future MySQL version. (WL #14124)

- TLS support in MySQL has been moving toward a channel model using named sets of TLS parameters that apply to different securable ports or protocols. For example, to query the state of a particular TLS channel, use the Performance Schema `tls_channel_status` table:

```
mysql> SELECT VALUE FROM performance_schema.tls_channel_status
       WHERE CHANNEL = 'mysql_main' AND PROPERTY = 'Enabled';
+-------+
| VALUE |
+-------+
| Yes   |
+-------+
```

  This makes monolithic parameters that apply to TLS support as a whole less applicable, so the following options and system variables are now deprecated and subject to removal in a future MySQL version:

  - The `--ssl` and `--admin-ssl` server options.

  - The `have_ssl` and `have_openssl` system variables.

  The `--ssl` and `--admin-ssl` options are enabled by default, so it is normally unnecessary to specify them. As an alternative to specifying those options in negated form, if it is desired to disable encrypted connections for the main or administrative interface, set the corresponding TLS version system variable to the empty value to indicate that no TLS versions are supported. For example, these lines in the server `my.cnf` file disable encrypted connections for both interfaces:

```
[mysqld]
tls_version=''
admin_tls_version=''
```

  (WL #14481)

## Error Handling

- Information written to the server error log for client timeouts now includes (if available) the timeout value, and client user and host. (Bug #31581289, Bug #100112)

## Event Scheduler Notes

- If the Event Scheduler is enabled, enabling the `super_read_only` system variable prevents it from updating event “last executed” timestamps in the `events` data dictionary table. This causes the Event Scheduler to stop the next time it tries to execute a scheduled event, after writing a message to the server error log.

  Previously, if enabling `super_read_only` caused the Event Scheduler to stop, then after subsequently disabling `super_read_only`, it was necessary to manually restart the Event Scheduler by enabling it again. As a convenience, the server now automatically restarts the Event Scheduler as needed when either `read_only` or `super_read_only` is disabled. (Bug #31633859)

## Firewall Notes

- In MySQL 8.0.23, MySQL Enterprise Firewall implemented group profiles that each can apply to multiple accounts, in addition to the previously implemented account profiles that each apply to a single account. See Using MySQL Enterprise Firewall.

  A group profile with a single member account is logically equivalent to an account profile for that account, so it is possible to administer the firewall using group profiles exclusively, rather than a mix of account and group profiles. For new firewall installations, that is accomplished by uniformly creating new profiles as group profiles and avoiding account profiles. For upgrades from firewall installations that already contain account profiles, MySQL Enterprise Firewall now includes a stored procedure named `sp_migrate_firewall_user_to_group()` for converting account profiles to group profiles.

  Due to the greater flexibility offered by group profiles, all aspects of the firewall related to account profiles are now deprecated and subject to removal in a future MySQL version:

  - `INFORMATION_SCHEMA` tables: `MYSQL_FIREWALL_USERS`, `MYSQL_FIREWALL_WHITELIST`
  - `mysql` system schema tables: `firewall_users`, `firewall_whitelist`
  - `mysql` system schema stored procedures: `sp_reload_firewall_rules()`, `sp_set_firewall_mode()`
  - Loadable functions: `read_firewall_users()`, `read_firewall_whitelist()`, `set_firewall_mode()`

  Additionally, if the server detects account profiles at startup, it writes a warning for every successfully loaded account profile.

  For information about converting account profiles to group profiles (which you should do at your earliest convenience), see Migrating Account Profiles to Group Profiles. (WL #14174)

## Packaging Notes

- Binary packages that include `curl` rather than linking to the system `curl` library have been upgraded to use `curl` 7.77.0. (Bug #33077562)
- For Ubuntu packages, the AppArmor profile for `mysqld` was too restrictive regarding PID and socket file names and failed for servers not using the exact names in the profile. Now the profile applies to the directories in which the files live, enabling it to apply for different file names, and multiple servers. (Bug #32857611)
- The `dh-systemd` package has been removed from Ubuntu 21.04, so the dependency on it has been removed from MySQL packages built for that distribution. (Bug #32688072)

- For Debian packages, an `EnvironmentFile` directive was added to enable the systemd service to read environmental variables from the `/etc/default/mysql` file if it is present. (Bug #32082863, Bug #101363)
- Debian packages now use `/run` rather than `/var/run` for path names. (Bug #31955638)
- The bundled `lz4` library was upgraded to version 1.9.3. (Bug #29747853)
- For Debian packages, the `update-alternatives` priority of the MySQL configuration file was increased to ensure it replaces an existing file from a previously installed distribution. (Bug #29606955)

## Pluggable Authentication

- **Linux:** MySQL Enterprise Edition now supports an authentication method that enables users to authenticate to MySQL Server using Kerberos, provided that appropriate Kerberos tickets are available or can be obtained. It is available only on MySQL server and client hosts running Linux, but can access Kerberos services running on non-Linux hosts. For details, see Kerberos Pluggable Authentication.

  This authentication method uses a pair of plugins, `authentication_kerberos` on the server side and `authentication_kerberos_client` on the client side. The server-side Kerberos authentication plugin is included only in MySQL Enterprise Edition. It is not included in MySQL community distributions. The client-side plugin is included in all distributions, including community distributions. This enables clients from any distribution to connect to a server that has the server-side plugin loaded. (WL #14336)

## Server Administration

- Setting the session value of the `innodb_strict_mode` system variable is now a restricted operation and the session user must have privileges sufficient to set restricted session variables.

  For information about the privileges required to set restricted session variables, see System Variable Privileges. (Bug #32944980, WL #14610)

## Spatial Data Support

- The `ST_Buffer()` function now permits the geometry argument to have a geographic spatial reference system (SRS), if the geometry is a point value. Previously, `ST_Buffer()` supported only geometry arguments in a Cartesian SRS. The `ST_Difference()` and `ST_Union()` functions now permit the geometry arguments to have a geographic SRS. Previously, `ST_Difference()` and `ST_Union()` supported only geometry arguments in a Cartesian SRS. See Spatial Operator Functions. (WL #14126, WL #14188, WL #14266)

## X Plugin Notes

- When the X DevAPI `Session.run_sql()` method was used to execute a query that returned multiple results, due to a caching issue, the `result.columns` property was not updated to reflect the columns present in the active result, although the `result.column_names` property was. (Bug #32887586)
- During an upgrade process, X Plugin logged a message stating that it was ready for connections once the TCP port and UNIX socket had been allocated. However, connections could not actually be accepted until after the upgrade process was complete. The message is now issued only after the upgrade has finished. (Bug #32814997)

## Functionality Added or Changed

- **Incompatible Change:** From MySQL 8.0.26, new aliases or replacement names are provided for most remaining identifiers that contain the terms “master”, which is changed to “source”; “slave”, which is

changed to “replica”; and “mts” (for “multithreaded slave”), which is changed to “mta” (for “multithreaded applier”). Help text is also changed where applicable to use the new names.

The following name replacements are visible in the Performance Schema tables, the process list, and the replica status information. These changes are incompatible with earlier releases. Monitoring tools that work with these instrumentation names might be impacted:

- Instrumented locks (mutexes), visible in the `mutex_instances` and `events_waits_*` Performance Schema tables with the prefix `wait/synch/mutex/`
- Read/write locks, visible in the `rwlock_instances` and `events_waits_*` Performance Schema tables with the prefix `wait/synch/rwlock/`
- Instrumented condition variables, visible in the `cond_instances` and `events_waits_*` Performance Schema tables with the prefix `wait/synch/cond/`
- Instrumented memory allocations, visible in the `memory_summary_*` Performance Schema tables with the prefix `memory/sql/`
- Thread names, visible in the `threads` Performance Schema table with the prefix `thread/sql/`
- Thread stages, visible in the `events_stages_*` Performance Schema tables with the prefix `stage/sql/`, and without the prefix in the `threads` and `processlist` Performance Schema tables, the output from the `SHOW PROCESSLIST` statement, the Information Schema `processlist` table, and the slow query log
- Thread commands, visible in the `events_statements_history*` and `events_statements_summary_*_by_event_name` Performance Schema tables with the prefix `statement/com/`, and without the prefix in the `threads` and `processlist` Performance Schema tables, the output from the `SHOW PROCESSLIST` statement, the Information Schema `processlist` table, and the output from the `SHOW REPLICA STATUS` statement

If the incompatible changes do have an impact for you, you can set the new system variable `terminology_use_previous` to `BEFORE_8_0_26` to make MySQL Server use the old versions of the names for the objects specified in the previous list. This enables monitoring tools that rely on the old names to continue working until they can be updated to use the new names. The system variable can be set with session scope to support individual functions, or global scope to be a default for all new sessions. When global scope is used, the slow query log contains the old versions of the names.

For semisynchronous replication, you can choose whether to use the new or the old versions of the system variables and status variables. New versions of the plugins that implement semisynchronous replication, one for the source server and one for the replica, are supplied that replace the terms master and slave with source and replica, and you can install these versions instead of the old ones:

- The `rpl_semi_sync_master` plugin (`semisync_master.so` library) for the source has a new version `rpl_semi_sync_source` (`semisync_source.so` library)
- The `rpl_semi_sync_slave` plugin (`semisync_slave.so` library) for the replica has a new version `rpl_semi_sync_replica` (`semisync_replica.so` library)

You cannot have both the new and the old version of the relevant plugin installed on an instance. If you use the new version of the plugins, the new system variables and status variables are available but the

old ones are not. If you use the old version of the plugins, the old system variables and status variables are available but the new ones are not.

The following internal-use items are converted to use the new terms but are not externalized to users or monitoring tools, and MySQL Server handles any necessary resolution internally:

- C++ filenames in source code
- Header guards in C++ files
- Debug symbols
- User variables passed in the replication protocol handshake by the replica when it connects to a replication source server (the replica sets both the old and the new name)

The following categories of identifiers have a new alias, and a deprecation warning is issued when the old name is used, although the old name continues to work. Both names are available in Performance Schema tables and status displays, and no deprecation warning is issued when reading these. The new aliases are not affected by the new system variable `terminology_use_previous`, and can still be used when it is set:

- System variables that contain the terms “master”, “slave”, or “mts”, with the exception of some that have already been deprecated or are scheduled for deprecation, and those defined by NDB. If these system variables are persisted using a SET PERSIST statement, both the old and the new name are

persisted, regardless of which was specified in the statement. With a RESET PERSIST statement, both are reset.

- Status variables that contain the terms “master”, “slave”, or “mts”, with the exception of those defined by NDB.
- Command-line options for `mysqld` that contain the terms “master”, “slave”, or “mts”, with the exception of some that have already been deprecated or are scheduled for deprecation, and those defined by NDB.
- Command-line options for `mysqladmin` that contain the terms “master”, “slave”, or “mts”.
- Command-line options for `mysqlbinlog` that contain the terms “master”, “slave”, or “mts”.
- Command-line options for `mysqldump` that contain the terms “master”, “slave”, or “mts”.
- An SQL function that contains the term “master”.

The complete list of identifiers with new aliases (or in the case of semisynchronous replication, replacements) is as follows:

- System variables:
  - `master_verify_checksum` now has the alias `source_verify_checksum`
  - `sync_master_info` now has the alias `sync_source_info`
  - `init_slave` now has the alias `init_replica`
  - `rpl_stop_slave_timeout` now has the alias `rpl_stop_replica_timeout`
  - `log_slow_slave_statements` now has the alias `log_slow_replica_statements`
  - `slave_max_allowed_packet` now has the alias `replica_max_allowed_packet`
  - `slave_compressed_protocol` now has the alias `replica_compressed_protocol`
  - `slave_exec_mode` now has the alias `replica_exec_mode`
  - `slave_type_conversions` now has the alias `replica_type_conversions`
  - `slave_sql_verify_checksum` now has the alias `replica_sql_verify_checksum`
  - `slave_parallel_type` now has the alias `replica_parallel_type`
  - `slave_preserve_commit_order` now has the alias `replica_preserve_commit_order`
  - `log_slave_updates` now has the alias `log_replica_updates`
  - `slave_allow_batching` now has the alias `replica_allow_batching`
  - `slave_load_tmpdir` now has the alias `replica_load_tmpdir`
  - `slave_net_timeout` now has the alias `replica_net_timeout`
  - `sql_slave_skip_counter` now has the alias `sql_replica_skip_counter`
  - `slave_skip_errors` now has the alias `replica_skip_errors`

  - `slave_checkpoint_period` now has the alias `replica_checkpoint_period`
  - `slave_checkpoint_group` now has the alias `replica_checkpoint_group`
  - `slave_transaction_retries` now has the alias `replica_transaction_retries`
  - `slave_parallel_workers` now has the alias `replica_parallel_workers`
  - `slave_pending_jobs_size_max` now has the alias `replica_pending_jobs_size_max`
  - `pseudo_slave_mode` now has the alias `pseudo_replica_mode`
  - `skip_slave_start` now has the alias `skip_replica_start`
- If the new `rpl_semi_sync_source` and `rpl_semi_sync_replica` plugins are used for semisynchronous replication:
  - `rpl_semi_sync_slave_enabled` is replaced by `rpl_semi_sync_replica_enabled`
  - `rpl_semi_sync_slave_trace_level` is replaced by `rpl_semi_sync_replica_trace_level`
  - `rpl_semi_sync_master_wait_for_slave_count` is replaced by rpl_semi_sync_source_wait_for_replica_count
  - `rpl_semi_sync_master_enabled` is replaced by `rpl_semi_sync_source_enabled`
  - `rpl_semi_sync_master_timeout` is replaced by `rpl_semi_sync_source_timeout`
  - `rpl_semi_sync_master_trace_level` is replaced by `rpl_semi_sync_source_trace_level`
  - `rpl_semi_sync_master_wait_point` is replaced by `rpl_semi_sync_source_wait_point`
- The following system variables are not changed:
  - `ndb_slave_conflict_role` (NDB system variables are not changed)
  - `binlog_rotate_encryption_master_key_at_startup` (“master key” is an accepted term)
  - `slave_rows_search_algorithms` (This system variable is already deprecated)
  - `master_info_repository` (This system variable is already deprecated)
- Status variables:
  - `Slave_open_temp_tables` now has the alias `Replica_open_temp_tables`
  - `Slave_rows_last_search_algorithm_used` now has the alias `Replica_rows_last_search_algorithm_used`
- If the new rpl_semi_sync_source and rpl_semi_sync_replica plugins are used for semisynchronous replication:
  - `Rpl_semi_sync_slave_status` is replaced by `Rpl_semi_sync_replica_status`
  - `Rpl_semi_sync_master_status` is replaced by `Rpl_semi_sync_source_status`

  - `Rpl_semi_sync_master_clients` is replaced by `Rpl_semi_sync_source_clients`
  - `Rpl_semi_sync_master_yes_tx` is replaced by `Rpl_semi_sync_source_yes_tx`
  - `Rpl_semi_sync_master_no_tx` is replaced by `Rpl_semi_sync_source_no_tx`
  - `Rpl_semi_sync_master_wait_sessions` is replaced by `Rpl_semi_sync_source_wait_sessions`
  - `Rpl_semi_sync_master_no_times` is replaced by `Rpl_semi_sync_source_no_times`
  - `Rpl_semi_sync_master_timefunc_failures` is replaced by `Rpl_semi_sync_source_timefunc_failures`
  - `Rpl_semi_sync_master_wait_pos_backtraverse` is replaced by `Rpl_semi_sync_source_wait_pos_backtraverse`
  - `Rpl_semi_sync_master_tx_wait_time` is replaced by `Rpl_semi_sync_source_tx_wait_time`
  - `Rpl_semi_sync_master_tx_waits` is replaced by `Rpl_semi_sync_source_tx_waits`
  - `Rpl_semi_sync_master_tx_avg_wait_time` is replaced by `Rpl_semi_sync_source_tx_avg_wait_time`
  - `Rpl_semi_sync_master_net_wait_time` is replaced by `Rpl_semi_sync_source_net_wait_time`
  - `Rpl_semi_sync_master_net_waits` is replaced by `Rpl_semi_sync_source_net_waits`
  - `Rpl_semi_sync_master_net_avg_wait_time` is replaced by `Rpl_semi_sync_source_net_avg_wait_time`

- NDB-related status variables are not changed.
- For `mysqld`, the command-line versions of all the aliased and replaced system variables in the lists above have equivalent command-line aliases or replacements, plus the following command-line option that is not a system variable:
  - `show-slave-auth-info` has the alias `show-replica-auth-info`
- The following command-line options are not changed:
  - `abort-slave-event-count` (This command-line option is scheduled for deprecation)
  - `disconnect-slave-event-count` (This command-line option is scheduled for deprecation)
  - `master-info-file` (This command-line option is already deprecated)
  - `master-retry-count` (This command-line option is already deprecated)
- For `mysqladmin`, the option `start-slave` now has the alias `start-replica`, and the option `stop-slave` now has the alias `stop-replica`.
- For `mysqlbinlog`, the option `read-from-remote-master` now has the alias `read-from-remote-source`.

  - For mysqldump, the following command-line options have new aliases:
    - `apply-slave-statements` now has the alias `apply-replica-statements`
    - `delete-master-logs` now has the alias `delete-source-logs`
    - `dump-slave` now has the alias `dump-replica`
    - `include-master-host-port` now has the alias `include-source-host-port`
    - `master-data` now has the alias `source-data`
  - The built-in SQL function `MASTER_POS_WAIT` has a new alias `SOURCE_POS_WAIT`.

  (WL #14194, WL #14628)

- **InnoDB:** The new `innodb_segment_reserve_factor` system variable permits configuring the percentage of tablespace file segment pages that are reserved as empty pages. For more information, see Configuring the Percentage of Reserved File Segment Pages.

  Thanks to Facebook for the contribution. (Bug #32312743, Bug #102044, WL #14521)

- **Group Replication:** For Group Replication, a group in single-primary mode can now be configured to stay in super read-only mode, so that it only accepts replicated transactions and does not accept any direct writes from clients. This setup means that when a group's purpose is to provide a secondary backup to another group for disaster tolerance, you can ensure that the secondary group remains synchronized with the first. You can configure the group to remain in super read-only mode when a new primary is elected, by disabling the action that normally takes place to remove that mode on the primary.

  Administrators can configure a group in this way using the new Group Replication functions `group_replication_enable_member_action()` and `group_replication_disable_member_action()`, which can enable and disable actions for members of a group to take in specified situations. The functions can also be used on servers that are not part of a group, as long as the Group Replication plugin is installed. Member actions are configured on the primary and propagated to other group members and joining members using group messages. Another function `group_replication_reset_member_actions()` is available to reset the member actions configuration to the default setting for all member actions. (WL #13855)

- **Group Replication:** You can now select an alternative UUID to form part of the GTIDs that are used when internally generated transactions for view changes (`View_change_log_event`) are written to the binary log. The system variable `group_replication_view_change_uuid` added in this release specifies a UUID that is used instead of the group name (the value of `group_replication_group_name`). The alternative UUID makes it easier to distinguish view change events from transactions received by the group from clients. This can be useful if your setup allows for failover between groups, and you need to identify and discard transactions that were specific to the backup group. Note that all members of the group must have the same alternative UUID specified, so groups set up in this way cannot include members from releases prior to MySQL 8.0.26. (WL #14539)

- The URL for downloading Boost was updated. Thanks to Marcelo Altmann for the contribution. (Bug #32856104, Bug #103611)

- The clone plugin now permits cloning from a donor MySQL server instance to a hotfix MySQL server instance of the same version and release. Previously, the hotfix server instance was not recognized as the same MySQL version and release. (Bug #32523635)

- These statements now report `utf8mb3` rather than `utf8` when writing character set names: `EXPLAIN`, `SHOW CREATE PROCEDURE`, `SHOW CREATE EVENT`.

  Stored program definitions retrieved from the data dictionary now report `utf8mb3` rather than `utf8` in character set references. This affects any output produced from those definitions, such as `SHOW CREATE` statements.

  This error message now reports `utf8mb3` rather than `utf8` when writing character set names: `ER_INVALID_CHARACTER_STRING`. (Bug #32233614, Bug #32392077, Bug #32392209, Bug #32428538, Bug #32428598)

- On platforms that support `fdatasync()` system calls, the new `innodb_use_fdatasync` variable permits using `fdatasync()` instead of `fsync()` for operating system flushes. An `fdatasync()` system call does not flush changes to file metadata unless required for subsequent data retrieval, providing a potential performance benefit. The `innodb_use_fdatasync` variable can be set dynamically using a `SET` statement. (WL #14452)

## Bugs Fixed

- **Incompatible Change:** Within trigger bodies, `INSERT` or `UPDATE` statements containing a `SET` clause that used `OLD` or `NEW` values as assignment targets could raise an assertion or lead to a server exit. Such assignments are no longer permitted. (Bug #32803211)

- **Performance:** Internal functions used to copy values between columns have been improved such that computations not necessary when the values are of similar types are no longer performed. Queries using temporary tables should be noticeably faster with this enhancement. Our internal testing has shown such queries being executed up to 11% faster than previously; as always, your results may differ from these depending on environment, configuration, and other factors. (Bug #32742537)

- **InnoDB:** To reduce the number of unnecessary warning messages in the error log, instances of the `fil_space_acquire()` function in the `InnoDB` sources were replaced by the `fil_space_acquire_silent()` function where possible. (Bug #32944543)

- **InnoDB:** The `TRX_FORCE_ROLLBACK_ASYNC` flag in the `InnoDB` sources, which indicates whether a transaction was rolled back asynchronously or by the owning thread, was found to be redundant and has been removed. (Bug #32912595)

- **InnoDB:** Use of the `ut_delete` symbol instead of the `UT_DELETE` macro in the InnoDB sources caused a failure in builds that disable Performance Schema memory tracing (`-DDISABLE_PSI_MEMORY=ON`). (Bug #32910699)

- **InnoDB:** Dictionary system `mutex_enter` and `mutex_exit` calls in the `InnoDB` sources were renamed to `dict_sys_mutex_enter()` and `dict_sys_mutex_exit()`, respectively. (Bug #32907980)

- **InnoDB:** Legacy `UNIV_INLINE` and `UNIV_MATERIALIZE` artifacts were removed from `InnoDB` sources. `UNIV_HOTBACKUP` was added to method declarations in some header files. (Bug #32894165)

- **InnoDB:** The `lock_sys` sharded `rw_lock` index used random index values generated by the `ut_rnd_interval()` function, which was not optimal for low-concurrency workloads. (Bug #32880577)

- **InnoDB:** A string value setting for the `innodb_redo_log_encrypt` variable was not handled properly. (Bug #32851525)

- **InnoDB:** Read-write transaction set (`trx_sys->rw_trx_set`) shards, each with a dedicated mutex, were introduced to alleviate transaction system mutex (`trx_sys->mutex`) contention caused by

transaction set insertions and removals. Related enhancements include moving transaction set modifiers to less critical locations, eliminating heap allocation inside of the `TrxUndoRsegs` constructor, converting transaction state (`trx->state`) and transaction start time (`trx->start_time`) fields to `std::atomic` fields, and new assertion code to validate threads that operate on transactions. (Bug #32832196)

- **InnoDB:** Record buffer logic for the `InnoDB` memcached `GET` command was revised. (Bug #32828352)
- **InnoDB:** The `ut_list` base member in the `InnoDB` sources now locates list nodes using the element portion of the list type rather than storing a member pointer in the base node of a list at runtime, which waisted resources. The patch also includes other `ut_list` related code improvements. (Bug #32820458)
- **InnoDB:** After upgrading from MySQL 5.6 to MySQL 5.7 and starting the server with undo log truncation enabled (`innodb_undo_log_truncate=ON`), a deadlock occurred when an undo tablespace truncate operation was initiated. The deadlock caused a long semaphore wait and an eventual failure. A direct upgrade from MySQL 5.6 to MySQL 5.7.35 or later avoids this potential issue. (Bug #32800020)
- **InnoDB:** Type-safe enhancements for PSI_memory_key identifiers were introduced. PSI_memory_key identifiers are used by Performance Schema for instrumentation of memory operations. With this enhancement, `ut::aligned_`*`name`* library functions are able to report type errors at compile time. (Bug #32797838)
- **InnoDB:** The `buf_get_LRU_mutex()` function was optimized to avoid acquiring the LRU mutex unnecessarily when flushing from the flush list. (Bug #32797451, Bug #103391)
- **InnoDB:** In debug builds, an access to `Fil_shard::m_deleted_spaces` (deleted tablespaces vector) was not protected by the `Fil_shard` mutex, causing a failure. (Bug #32792816)
- **InnoDB:** Attempting to start MySQL with `--innodb-dedicated-server` on a machine with 2GB of RAM resulted in the creation of a single redo log file, which prevented the server from starting. Now in such cases, at least two log files are created.

  Our thanks to Adam Cable for the contribution. (Bug #32788772, Bug #103372)
- **InnoDB:** A failure occurred when truncating a partitioned table after an operation that added too many columns to the table, exceeding the column limit. The number of columns added is now evaluated before an `ADD COLUMN` operation is permitted. (Bug #32788564, Bug #103363)
- **InnoDB:** On platforms that support punch hole where the disk is near full, creating a tablespace with a large `AUTOEXTEND_SIZE` setting could lead to a no space on device failure and a subsequent `InnoDB` recovery failures. (Bug #32771235)
- **InnoDB:** The list of table locks requested by a transaction (`trx->lock.table_locks`), which is a subset of the transaction lock list (`trx->lock.trx_locks`), was removed. Table locks requested by a transaction are now are now listed at the beginning of the transaction lock list instead. (Bug #32762881)
- **InnoDB:** A failure occurred during recovery with the disk being near full, leaving the data in an inconsistent state. The failure occurred in the `fil_tablespace_redo_extend()` function, which is used to redo a tablespace extension operation. (Bug #32749974, Bug #32748733)
- **InnoDB:** After relocating a file-per-table tablespace offline and making the new location known to `InnoDB` using the `innodb_directories` option, an `ALTER TABLE` operation that used the `COPY` algorithm failed with a storage engine error. The failure was due to a renaming check, which searched the data directory instead of the new directory location. (Bug #32721533)
- **InnoDB:** `ut_allocator()` compliance issues with the C++ standard template library (STL) were addressed. (Bug #32715698)

- **InnoDB:** Instances of `ut_allocator::allocate()` instantiated by `std::vector` in the `InnoDB` sources failed to trace memory allocations and deallocations performed implicitly by `std::vector`. The same issue was found for other C++ standard template library (STL) and STL-like data-structures. (Bug #32715688)

- **InnoDB:** The `ut_allocator::construct()` interface in the `InnoDB` sources, which is a custom interface implemented in a pre-C++11 style, caused unnecessary overhead. The interface was not necessary and has been removed. (Bug #32715381)

- **InnoDB:** The `ut_list` length member variable in the `InnoDB` sources was replaced by an atomic field to permit lock free access without undefined behavior. The default `ut_list` constructor was replaced by a new constructor that performs all list initialization. (Bug #32715371)

- **InnoDB:** The `ut_allocator()` out of memory reporting mechanism in the `InnoDB` sources was not reliable and has been removed. (Bug #32715359)

- **InnoDB:** Implicit handling of Performance Schema metadata was implemented for `ut_allocator::allocate_large()` and `ut_allocator::deallocate_large()` functions in the `InnoDB` sources. This modification aligns Performance Schema metadata handling with that of similar allocation functions. (Bug #32714144)

- **InnoDB:** Stalls were caused by concurrent `SELECT COUNT(*)` queries where the number of parallel read threads exceeded the number of machine cores. A patch for this issue was provided for Windows builds in MySQL 8.0.24. The MySQL 8.0.26 patch addresses the same issue on other affected platforms. (Bug #32678019)

  References: See also: Bug #32224707.

- **InnoDB:** To avoid costly calls to the `rec_get_offsets()` function, which determines the offsets for each field in a record, caching of offsets is extended to indexes that meet certain requirements such as having a fixed length, no virtual columns, instant columns, and so on. (Bug #32673649)

- **InnoDB:** When starting the server with a data directory that was restored by MySQL Enterprise Backup, the doublewrite buffer (controlled by the `innodb_doublewrite` variable) remained disabled until the next server restart. (Bug #32642866)

- **InnoDB:** “Too many open files” errors were encountered when creating a large number of tables. (Bug #32634620)

  References: This issue is a regression of: Bug #32541241.

- **InnoDB:** `InnoDB` recovery was unable to proceed due to a page tracking system recovery failure, which should have been non-blocking. (Bug #32630875)

- **InnoDB:** An integer underflow issue was addressed in the `InnoDB` mecached plugin sources. (Bug #32620378, Bug #32620398)

- **InnoDB:** When a transaction started waiting for lock, the `InnoDB` lock system provided information to the server about the transaction currently holding the lock but failed to inform the server after releasing the lock and grating it to another waiting transaction. As a result, the replication applier thread coordinator was unable to detect potential deadlocks in the intended transaction commit order that could occur if the third transaction in this scenario committed after the initial waiting transaction. (Bug #32618301)

- **InnoDB:** When inserting a record into a unique secondary index, the index record locks taken to prevent concurrent transactions from inserting a conflicting record into the affected range included an unnecessary gap lock on the first record after the range. (Bug #32617942)

- **InnoDB:** `InnoDB` code that creates dynamically allocated over-aligned types was replaced by `ut::aligned_name` library functions. (Bug #32601599)

- **InnoDB:** Memory allocation functions belonging to the API that handles dynamic storage of over-aligned types (`ut::aligned_name` library functions) were extended to support memory tracing using Performance Schema. The `HAVE_PSI_MEMORY_INTERFACE` source configuration option enables the memory tracing module.

  An `aligned_zalloc()` library function, which provides support for zero-initialized memory allocations, was added to the API. (Bug #32600981)

  References: See also: Bug #32601599, Bug #32246061, Bug #32246200.

- **InnoDB:** Sampling of `InnoDB` data for the purpose of generating histogram statistics is now supported with all transaction isolation levels supported by `InnoDB`. Previously, sampling was performed using only the `READ UNCOMMITTED` isolation level. (Bug #32555575)

- **InnoDB:** An index with a key prefix length greater than 767 bytes was permitted on a table defined with the `REDUNDANT` row format, exceeding the index key prefix length limit for that row format. The `ALTER TABLE` operation that added the index validated the index key prefix length for the row format defined by the `innodb_default_row_format` variable instead of the actual row format of the table. The fix ensures that index key prefix length is validated for the correct row format. (Bug #32507117, Bug #102597)

- **InnoDB:** Adaptive hash index latches did not provide meaningful latch location information. (Bug #32477773)

- **InnoDB:** A dependency related to redo and undo log encryption at server initialization time was removed. (Bug #32476724)

- **InnoDB:** An online buffer pool resizing operation freed the previous buffer pool page hash, conflicting with a concurrent buffer pool lookup that required the previous page hash. (Bug #32460315)

- **InnoDB:** When using the TempTable storage engine (`internal_tmp_mem_storage_engine=TempTable`), more than 255 aggregate functions in a `SELECT` list caused errors due to overflow of an internal variable that stores indexed column field positions. (Bug #32458104, Bug #102468)

- **InnoDB:** A workload stalled while executing a undo tablespace truncation operation on an instance with a large buffer pool. The function that truncates marked undo tablespaces now takes a shared latch instead of an exclusive latch, and the shared latch is taken for a shorter period of time. (Bug #32353863, Bug #102143)

- **InnoDB:** A programming interface was added for handling dynamic storage of over-aligned types. (Bug #32246200, Bug #32246061)

- **InnoDB:** Under certain circumstances, a failure during `InnoDB` recovery could have caused a loss of committed changes. Checkpoints permitted during recovery did not handle page flushing, flush list maintenance, or persisting changes to the data dictionary table buffer as necessary for a proper checkpoint operations. Checkpoints and advancing the checkpoint LSN are therefore no longer permitted until redo log recovery is complete and data dictionary dynamic metadata (`srv_dict_metadata`) is transferred to data dictionary table (`dict_table_t`) objects. Should the redo log run out of space during recovery or after recovery (but before data dictionary dynamic metadata is transferred to data dictionary table objects) as a result of this change, an `innodb_force_recovery` restart may be required, starting with at least the `SRV_FORCE_NO_IBUF_MERGE` setting or, in case that fails, the `SRV_FORCE_NO_LOG_REDO` setting. If an `innodb_force_recovery` restart fails in this scenario, recovery from backup may be necessary. (Bug #32200595)

- **InnoDB:** Rollback segments are now initialized in parallel during startup. Previously, rollback segments were initialized serially.

  Thanks to Zheng Lai and the Tencent Cloud-Native Database team for submitting the contribution this bug fix is based on. (Bug #32170127, Bug #101658)

- **InnoDB:** A failure occurred during testing of `innodb_log_writer_threads` variable configuration. The failure was caused by a race condition. (Bug #32129814)

  References: This issue is a regression of: Bug #30088404.

- **InnoDB:** A race condition occurred between a purge thread that was truncating an undo tablespace and a server thread that queried the `INFORMATION_SCHEMA.FILES` table. As a result, the truncated undo tablespace did not appear in the `INFORMATION_SCHEMA.FILES` table when queried, which in turn caused a MySQL Enterprise Backup failure due to a dependency on the `INFORMATION_SCHEMA.FILES` table for undo tablespace file locations. (Bug #32104924, Bug #32654667)

- **InnoDB:** When DML operations are concentrated on a single table, purge work was performed by a single purge thread, which could result in slowed purge operations, increased purge lag, and increased tablespace file size if DML operations involve large object values. To address this issue, purge work is now automatically redistributed among available purge threads when the purge lag exceeds the `innodb_max_purge_lag` setting. (Bug #32089028)

- **InnoDB:** The function that populates the `INFORMATION_SCHEMA.INNODB_TABLESPACES` table accessed the space for a file and performed an unprotected `stat()` operation and retrieval of file size information. (Bug #32025344)

- **InnoDB:** Code related to the `trx_t::is_recovered` flag in the `InnoDB` sources was revised to address various complexity and correctness issues. One of the issues addressed caused an XA transaction to be described incorrectly as “recovered”, which occurred when a client session disconnected from an XA transaction after `XA PREPARE`. (Bug #31870582)

- **InnoDB:** TempTable debug assertion code for an `Indexed_cells` member function (`cell_from_mysql_buf_index_read()`) did not account for non-nullable columns with zero length. (Bug #31091089)

- **InnoDB:** Using the InnoDB memcached plugin, attempting to retrieve multiple values in a single get command returned an incorrect value. (Bug #29675958, Bug #95032)

- **InnoDB:** The `trx_sys_t::serialisation_mutex` was introduced to reduce contention on the on the `trx_sys_t::mutex`. The new mutex protects the `trx_sys_t::serialisation_list` when a transaction number is assigned, which was previously protected by the `trx_sys_t::mutex`.

  Thanks to Zhai Weixiang for the contribution. (Bug #27933068, Bug #90643)

- **Partitioning:** When a table was partitioned by `TIMESTAMP` and a timestamp literal with a time zone offset was used in the `WHERE` clause of a `SELECT` statement, it was possible for a partition to be omitted from the result set.

  When a time zone offset is specified in a timestamp literal, it is expected to be converted to a timestamp without a time zone offset, and then compared against a timestamp column, but this was not done properly in all cases, with the result that a partition could be pruned while selecting the partitions to be scanned for the query.

  We fix this by making sure that a timestamp with a time zone offset is always converted as described before comparing with values from the column. (Bug #101530, Bug #32134145)

- **Replication; Group Replication:** When the system variable `replication_optimize_for_static_plugin_config` was set, the plugins for Group Replication and semi-synchronous replication could not be uninstalled cleanly on server shutdown. (Bug #32798287)

- **Replication:** When the `SOURCE_CONNECTION_AUTO_FAILOVER=1` option was set on the `CHANGE REPLICATION SOURCE TO` statement for a replication channel, a `STOP REPLICA IO_THREAD` statement did not stop the monitor thread that monitors for connection failures on the channel, and could incorrectly stop the applier thread. (Bug #32892977)

- **Replication:** A deadlock could occur when `START GROUP_REPLICATION` and `STOP GROUP_REPLICATION` statements were issued at the same time that a view change was taking place for the group. (Bug #32738137, Bug #32836868)

- **Replication:** An incorrect default value in code meant that the allowlist of IP addresses permitted for a replication group was implicitly reconfigured although no value had been supplied. (Bug #32714911)

- **Replication:** A deadlock could occur if a `STOP GROUP_REPLICATION` statement was issued when a replication channel on a group member was attempting to commit a transaction. The server now rolls back the transaction immediately if it cannot acquire the relevant lock, rather than waiting for the lock and the commit to complete and causing the deadlock. (Bug #32633176)

- **Replication:** On a multithreaded replica, the reference to the active event was sometimes managed incorrectly when retrying a transaction. (Bug #32590974)

- **Replication:** After a DML operation was performed on the last partition of a table with more than 128 partitions, MySQL Server and MySQL clients (such as `mysqlbinlog`) parsed the event information from the binary log incorrectly, resulting in an inaccurate partition ID being stated. The information is now read using an event reader function that is endianness independent. (Bug #32358736, Bug #102191)

- **Replication:** In a new MySQL Server installation, the `mysql.gtid_executed` system table was missing the property `STATS_PERSISTENT=0` to disable persistent statistics, which is present for the other replication-related tables. (Bug #32250735)

- **Replication:** When the same row in a table was updated multiple times by the same event, the replication applier's hash scan algorithm omitted to check for JSON partial updates, which are logged when `binlog_row_value_options=PARTIAL_JSON` is set. This could result in replication stopping with a "key not found" error. (Bug #32221703)

- **Replication:** Replica servers now check and validate the transaction ID part of a GTID before applying and committing the transaction associated with it. (Bug #32103192)

- **Replication:** Replication could stop on a multithreaded replica if a unique secondary key was omitted from the writeset hashes used to compute transaction dependencies, leading to errors when executing the transactions on the multithreaded replica. Write set hashes now always include unique secondary keys even if they are not included in the read set and write set. (Bug #31636339)

- **Replication:** MySQL semisynchronous replication did not clean up connection artifacts correctly after sending a reply to the source, causing a mismatch that meant the connection had to be re-established. The issue has now been fixed. (Bug #31363518, Bug #32441220, Bug #32759421)

- **Replication:** On a multithreaded replica (with `slave_parallel_workers > 0` ), the algorithm used by GTID auto-positioning to check for missing transactions briefly sets a low value (4) for the event position in the course of its calculations. If the operation is stopped at that moment, the recovery process

that resolves gaps in the sequence of transactions can fail. The process to resolve gaps is not actually necessary when GTID auto-positioning is used, so the process has now been disabled in that situation.

As a result, on a multithreaded replica, when `GTID_MODE = ON` is set for the instance and `SOURCE_AUTO_POSITION` is set for the channel using the `CHANGE REPLICATION SOURCE TO` statement, the following behaviors now apply:

- A `START REPLICA UNTIL SQL_AFTER_MTS_GAPS` statement just stops the applier thread when it finds the first event to execute, and does not attempt to check for gaps in the sequence of transactions.
- A `CHANGE REPLICATION SOURCE TO` statement does not automatically fail if there are gaps in the sequence of transactions.

These changed behaviors only apply on a multithreaded replica that uses GTIDs and GTID auto-positioning, and not on a replica that uses binary log position-based replication. (Bug #30571587, Bug #97694)

- **Group Replication:** When running Group Replication in single-primary mode, unnecessary copies of the transaction data were created during data serialization. This is now done in a single step to reduce the memory footprint. (Bug #32781945)
- **Group Replication:** Replicated transactions in GTID-based replication include an `original_commit_timestamp` to show when the transaction was committed on the original source server, and an `immediate_commit_timestamp` to show when it was committed on the replica. Previously, the `original_commit_timestamp` was not set for Group Replication view change events, which are agreed by the group but then generated and applied by each group member, so the timestamp appeared as zero in viewable output. For improved observability, group members now set local timestamp values for transactions associated with view change events. (Bug #32668567)
- **Group Replication:** Previously, a warning message was written to the error log if the original commit timestamp on a replicated transaction was more recent than the immediate commit timestamp on the replica applying it. The message could occur inappropriately if the fluctuation in the replication lag had a similar value to the clock difference between the machines involved, which could be made more likely by better quality connections between them. During Group Replication failover, it was possible for a message to be returned for every transaction, flooding the log. To avoid this situation, the warning message has been withdrawn. (Bug #32554807)
- **Group Replication:** When `group_replication_consistency` was set to `AFTER`, and a view change event was delayed until after a locally prepared transaction was completed, a different GTID could be applied to it, causing errors in replication. The data is now processed in the same sequence it is received to avoid the situation. (Bug #31872708)
- **Microsoft Windows:** Writing to Windows event logs could be unsuccessful. (Bug #32890918)
- **JSON:** Reading JSON values from tables that used the `CSV` storage engine raised an error such as `Cannot create a JSON value from a string with CHARACTER SET 'binary'`. This happened because the `CSV` engine uses `my_charset_bin` as the character set for the record buffer but creation of `JSON` values includes an explicit check for `my_charset_bin`, and raises an error if this character set is given.

  We handle this issue by passing the actual character set of the column instead of the character set of the buffer holding the data, which is always binary. (Bug #102843, Bug #32597017)
- A query for which the derived condition pushdown optimization could be applied was not so optimized when the query was part of `INSERT ... SELECT`. (Bug #32959186)

- Import operations for access privilege information became very slow for large numbers of accounts and schemas. (Bug #32934351)
- Queries which needed to sort the results of a full-text index scan were in some circumstances not handled correctly. (Bug #32924198)
- Thanks to Xiaoyu Wang for a code correction to `PFS_notification_registry::is_empty()`. (Bug #32919118, Bug #103788)
- Queries containing `GROUP BY`, `ORDER BY`, and `LIMIT` in a subquery and accessed using a cursor could cause a server exit. (Bug #32918240)
- When invoked with the `--help` and `--verbose` options, `mysqld` created an `auto.cnf` file in the current directory. (Bug #32906164)
- Queries that involved pushing a condition with view references down to a materialized derived table could cause a server exit. (Bug #32905044, Bug #32324234)
- Messages for `ER_CANT_INITIALIZE_UDF` errors could be truncated. (Bug #32891703)
- A regression was found in the simplification of streaming aggregation (`GROUP BY` of data already sorted) that was performed in MySQL 8.0.23.

  We fix this issue as follows: When there is an implicit grouping on a single table which is the subject of a fulltext search, we now force insertion of a temporary table to materialize `MATCH()` temporaries before they are sent to the `AggregateIterator`, since it tries to save and restore the rows it receives, but cannot properly include the fulltext search information, as it is hidden. (Bug #32889491)

  References: This issue is a regression of: Bug #31790217.
- For conversion of `-DBL_MAX` to string and back to double, the new double value differed from the original and was rejected as out of bounds. (Bug #32888982, Bug #103709)
- Now, whenever the `JSON_LENGTH()` function includes the optional *`path`* argument, the server rewrites it as `JSON_LENGTH(JSON_EXTRACT(doc, path))`. This means that `JSON_LENGTH()` now supports wildcards (such as `$`, `.`, and `*`) and array ranges in the path, as shown here:

```
mysql> SELECT JSON_LENGTH('[1,2,3,4,5,6,7]', '$[2 to 4]') AS x;
+------+
| x    |
+------+
|    3 |
+------+
```

  (Bug #32877703)
- For flags typically used for RPM and Debian packages, the new `WITH_PACKAGE_FLAGS` `CMake` option controls whether to add those flags to standalone builds on those platforms. The default is `ON` for nondebug builds. This is used to “harden” the builds; for example, by adding `-D_FORTIFY_SOURCE=2`. (Bug #32876974)
- The `NULLIF()` function did not perform all necessary checks for errors. (Bug #32865008)
- For views that depended on other views, output from the `SHOW CREATE VIEW` statement used during production of dump files could cause an error at restore time. (Bug #32854203, Bug #103583)
- The query-attributes code did not properly handle large 64-bit numbers. (Bug #32847269)
- Information retrieved from the Performance Schema `metadata_locks` table could be incorrect for foreign keys and `CHECK` constraints. (Bug #32832004, Bug #103532)

- When generating unique names for view columns, the server now considers only those objects whose names are visible. (Bug #32821372)
- When a condition is pushed down to a materialized derived table, a clone of the derived table expression replaces the column (from the outer query block) in the condition. When the cloned item included a `FULLTEXT` function, it was added to the outer query block instead of the derived table query block, which led to problems. To fix this, we now use the derived query block to clone such items. (Bug #32820437)
- A common table expression which was used more than once in a statement, at least once within a subquery that was subsequently removed during resolution due to being always true, or always false. (Bug #32813547, Bug #32813554)
- If a statement is prepared against a table that is persistent at preparation time but temporary at the time of first execution, an assertion could be raised. (Bug #32799797)
- An internal function used by spatial functions could reference memory after freeing it. (Bug #32793104)
- The `impossible filter` optimization removed MRR access paths that were required by the corresponding BKA access paths. (Bug #32787415)
- The new `WITH_AUTHENTICATION_CLIENT_PLUGINS` `CMake` option is enabled automatically if any corresponding server authentication plugins are built. Its value thus depends on other `CMake` options and it should not be set explicitly. (Bug #32778378)
- The MRR iterator normally filters out `NULL` keys by checking `impossible_null_ref()`, but when a join condition either contained an `IS NULL` predicate, or used the `NULL`-safe equals operator `<=>`, the optimizer had to check whether the join condition used the predicate terms as part of its join condition, and not set the internal flag `HA_MRR_NO_NULL_ENDPOINTS` in such cases. Now we check, using a bitmask, whether the each column in the key rejects `NULL`, in which case we can set `HA_MRR_NO_NULL_ENDPOINTS` without further checks. (Bug #32774281)
- For system variables with an enumeration type, `SET PERSIST_ONLY` *`var_name`* `= DEFAULT` persisted the numeric value and not the symbolic name. (Bug #32761053)
- The X DevAPI operations Collection.replaceOne and Collection.addOrReplaceOne now verify that the value of the `_id` field in a document matches the `id` parameter specified for the operation. If it does not, an error is returned. (Bug #32753547)
- For applications that use the C API to execute prepared statements, query attributes could not be used for prepared statements with no parameters. (Bug #32753030, Bug #32790714, Bug #32955044)
- The arguments to `IN()` were not always converted to the correct character set. (Bug #32746552)
- The `LOCATE()` function unconditionally returned `NULL` when an argument could not be evaluated. Now, when used in an expression that is determined to be non-nullable, the function returns zero instead. (Bug #32746536)
- The internal function `my_well_formed_len_utf32()` asserted when presented with a string of invalid length. Now in such cases, the function reports an invalid string instead. (Bug #32745294)
- The functions `TRIM()`, `RTRIM()`, and `LTRIM()` did not always perform proper error checking. (Bug #32744772)
- A previous fix in an internal resolver function ensured that it raises an error when a generated column cannot be resolved. This worked without any problem when the generated column is part of a `CREATE TABLE` statement, but in the case where the table with the generated column was generated on a MySQL 5.7 database and then upgraded to MySQL 8.0, an error was reported and the upgrade terminated.

We fix this by using the correct pointer when reporting the table causing the error in the function `fix_generated_columns_for_upgrade()`. (Bug #32738972)

- When looking inside rollup wrappers in the `SELECT` list, and trying to find the same item in the `GROUP BY` list, the server did not take into account that a cache might have been added around the expression. Now any such caches are unwrapped before looking for the item. (Bug #32729377, Bug #32918400)
- A prepared statement that used `MIN()` or `MAX()` could return an incorrect result if it also included dynamic parameters. (Bug #32713860, Bug #103170)
- Replication could fail if a DML statement was executed immediately after an XA transaction was rejected or forced to rollback due to a deadlock. (Bug #32707060)
- Queries containing multiple instances of `GROUPING()` were not always handled correctly. (Bug #32695103)
- When executing `EXPLAIN ANALYZE`, materialization iterators count every single `init()` call, even those that only retain existing data, causes materializations to appear to cost too little compared to the number of underlying calls. We fix this by allowing the materialization iterator to override the call and row counts with its own data. (Bug #32688540)
- A race condition in the metadata locking code could result in a server exit. (Bug #32686116)
- An index-only scan on a covering full-text index could return incorrect results for queries with multiple calls to `MATCH()` function depending on the order in which the `MATCH()` calls were evaluated. (Bug #32685616)
- Including query attributes for a prepared statement could cause a statement execution failure. (Bug #32676878)
- Executing DDL statements on a system table could cause a server exit. (Bug #32665990)
- With the `--online-migration` option, `mysql_migrate_keyring` could exit during connection establishment to the migration server. (Bug #32651203)
- `mysql_migrate_keyring` failed to enforce the condition that the source and destination keystores must differ. (Bug #32637784)
- System cache size checking could be inaccurate on Ubuntu. (Bug #32619199, Bug #102926)
- The server did not process correctly some queries that used `MATCH ... AGAINST` on a column with a fulltext index in a `HAVING` clause. (Bug #32617181)

  References: This issue is a regression of: Bug #30969045.
- Replicated transactions in GTID-based replication include an `original_commit_timestamp` to show when the transaction was committed on the original source server, and an `immediate_commit_timestamp` to show when it was committed on the replica. Previously, Group Replication servers set the `original_commit_timestamp` when they committed locally generated transactions, but they did not set the `immediate_commit_timestamp`. For improved observability, group members now set this timestamp. If the transaction originated in the group, the `original_commit_timestamp` and `immediate_commit_timestamp` are both generated by Group Replication, and are equal. If the transaction originated outside the group, the `original_commit_timestamp` is preserved, and an `immediate_commit_timestamp` is set. (Bug #32613896)
- The internal function `Item_func_match::eq()` raised an assert failure in debug builds when it was called with an argument that was an `Item_func_match_predicate`. The assertion was

added with the expectation that an `Item_func_match` object would not be compared with an `Item_func_match_predicate` object, but it was later found that this can happen during the `ONLY_FULL_GROUP_BY` check when the predicate is in a `HAVING` clause.

We fix this by removing the assertion so that the function returns false, instead. (Bug #32611913)

- `SELECT` using `DISTINCT` with a `GROUP BY ... WITH ROLLUP` on a primary key column returned a different result than when the column was not a primary key. (Bug #32565875)
- When an item in the `SELECT` list came from a table that was found to be constant, it was possible to add caches around it before adding `ROLLUP` wrappers, causing it to be unfindable in the group list (which had no such wrappers). This is addressed by unwrapping the caches. (Bug #32559356)
- The C client library could produce a `Malformed communication packet` error if a prepared statement parameter was longer than 8KB and the next parameter represented a `NULL` value. (Bug #32558782)
- When comparing arguments of different types, and the arguments are deemed constants, the server may insert its own hidden item cache to save on repeated conversion (and possible warnings). This is not visible in the output of any statements such as `EXPLAIN`.

  An issue arose due to the fact that resolution of `ROLLUP` runs after the comparisons are set up, and may replace the arguments, which could lead to incorrect results from comparisons. Now when this happens, we signal that the comparison needs to reconsider its caches, and not possibly re-use stale values. (Bug #32548377)
- Incorrect reference counting for a destructor in the implementation of a loadable function could raise an assertion for debug builds or report that the function does not exist for nondebug builds. (Bug #32528898, Bug #102649)
- MySQL source distributions now bundle the Google Test source code, which no longer need be downloaded to run Google Test-based unit tests. Consequently, the `WITH_GMOCK` and `ENABLE_DOWNLOADS` `CMake` options have been removed and are ignored if specified.

  This change also corrects a couple of build issues whereby source distributions incorrectly included an empty `source_downloads` directory, and `CMake` did not fail when `WITH_UNIT_TESTS` was enabled but the Google Test source was not found. (For the latter issue, it can no longer occur because the source is always present.) (Bug #32512077, Bug #27326599, Bug #29935278)
- When outer batched key access and block-nested loop (or hash join) occurred together in a query, both were reverted to nested loops in the plan interpretation to access paths. Problems arose in some cases in which non-equality join conditions had been pushed to a special kind of BKA index condition, and by converting to a regular index lookup (which does not check the BKA condition), it was possible to drop conditions that should have been checked.

  We fix this by removing the BKA index condition; since its use is very rare, potential gains in most practical queries are very low, and removing it decreases complexity significantly. (Bug #32424884)
- `SHOW CREATE USER` could cause an unexpected server exit if the structure of the `mysql.user` system table was manually altered. (Bug #32417780, Bug #31654586)
- For large values of the `group_concat_max_len` system variable, prepared statements that used the `GROUP_CONCAT()` function could be unnecessarily re-prepared across executions. (Bug #32413530, Bug #102344)
- Changes were not properly rolled back on re-execution of prepared statements or stored procedures for which walk and replace was performed as part of optimization. (Bug #32384324, Bug #32573871)

- Client programs using the asynchronous C API functions could leak memory if a connection timeout occurred. Thanks to Facebook for a contribution used in fixing this issue. (Bug #32372038, Bug #102189, Bug #32800091, Bug #103409, Bug #33171164, Bug #104461)
- The dynamic statistics cache exposed by the `INFORMATION_SCHEMA` could be updated in the middle of a transaction before it was known whether the transaction would commit, potentially caching information corresponding to no committed state. (Bug #32338335)
- Repreparation of a prepared statement at the beginning of an implicit transaction could cause an `ER_GTID_NEXT_TYPE_UNDEFINED_GROUP` error. (Bug #32326510, Bug #102031)
- `USING HASH` was removed from the definitions of Performance Schema thread pool tables because the Performance Schema does not support hash indexes. (Bug #32194617)
- Inserting a datetime literal with an explicit time zone offset into a `TIMESTAMP` column could produce the wrong time if `time_zone=SYSTEM` and the system time zone has DST transitions. (Bug #32173664, Bug #101670)
- While setting up `ORDER BY` for a window function, a window definition including a subquery was removed along with the subquery in the `ORDER BY`. (Bug #32103260)
- The use of a `SET PERSIST` or `SET PERSIST_ONLY` statement is now disallowed for the `group_replication_force_members` system variable. This system variable must be cleared after use before you can issue a `START GROUP_REPLICATION` statement, so it should not be persisted to the MySQL server instance's option file. (Bug #31983203)
- Upgrades from MySQL 5.7 to 8.0 failed if the MySQL installation had `FEDERATED` tables. (Bug #31973053)
- Persisting component-related system variables could result in unnecessary "unknown variable" error messages. (Bug #31968366)
- An out-of-memory error occurred when loading large amounts of data into tables with full-text search indexes. Not all of the memory allocated to the full-text search cache was accounted for when inserting data into the full-text search auxiliary tables. (Bug #31576731)
- The precision for a `UNION` between integer and decimal values could be calculated incorrectly, leading to truncation of the integer in the result. (Bug #31348202, Bug #99567)
- `InnoDB` wrote unnecessary warnings to the error log indicating that table statistics could not be calculated. (Bug #30865032)
- A secondary index over a virtual column became corrupted when the index was built online.

  For `UPDATE` statements, we fix this as follows: If the virtual column value of the index record is set to `NULL`, then we generate this value from the cluster index record. (Bug #30556595)
- `DROP DATABASE` for a database with a large number of tables (one million) could result in an out-of-memory condition. (Bug #29634540)
- The `system_time_zone` system variable is initialized at server startup from the server host settings and the `mysqld` environment, but if the server host time zone changed (for example, due to daylight saving time), `system_time_zone` did not reflect the change. Now it does. (Bug #29330089, Bug #94257)
- For upgrades on Ubuntu, if an existing `my.cnf` file is found, it is renamed to `my.cnf.bak` and a warning is issued. (Bug #24486556, Bug #82639)
- Boolean system variables could be assigned a negative value. (Bug #11758439, Bug #50643)

- Prepared statements did not always make use of index extensions (see Use of Index Extensions). (Bug #103711, Bug #32897525)

  References: See also: Bug #103710, Bug #32897503.

- When enabled, the `prefer_ordering_index` optimizer switch had a negative effect on the performance of prepared statements. (Bug #103710, Bug #32897503)

  References: See also: Bug #103711, Bug #32897525.

- Successive `INSERT()` function calls could sometimes yield invalid `NULL` results. (Bug #103513, Bug #32824978)

- Some `NOT IN` subqueries did not return correct results due to a regression in `NULL` handling. (Bug #103331, Bug #32773801)

  References: This issue is a regression of: Bug #30473261.

- Whenever a `SELECT ... FOR UPDATE NOWAIT` statement was unable to obtain a record lock, a message `Got error 203 when reading table ...` was written to the error log, even though though this is a relatively common occurrence, the logging of which led to excessive use of disk space.

  Our thanks to Brian Yue for this contribution. (Bug #103159, Bug #32705614)

- `0 - (MAX(BIGINT) + 1)` returned `-(MAX(BIGINT) + 1)`. Now an out of range error is returned instead. (Bug #103093, Bug #32693863)

# Changes in MySQL 8.0.25 (2021-05-11)

- Packaging Notes
- Bugs Fixed

## Packaging Notes

- Binary packages that include `curl` rather than linking to the system `curl` library have been upgraded to use `curl` 7.76.0.

## Bugs Fixed

- On Fedora 34, builds from source failed due to an undefined reference to symbol `crc32_z@@ZLIB_1.2.9`. (Bug #32702860)

- Docker containers for MySQL Enterprise Edition no longer need to run with root privileges. (Bug #32472242)

- For a prepared, implicitly grouped `SELECT` statement in which the `WHERE` clause was determined always to be false, the result of some aggregate functions could sometimes be picked up from the previous execution of the statement. (Bug #103192, Bug #32717969)

# Changes in MySQL 8.0.24 (2021-04-20)

- Audit Log Notes
- Compilation Notes
- Connection Management Notes

- Error Handling
- Keyring Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Pluggable Authentication
- Security Notes
- Spatial Data Support
- Functionality Added or Changed
- Bugs Fixed


## Audit Log Notes

- MySQL Enterprise Audit now supports audit log file pruning, for JSON-format log files. See Space Management of Audit Log Files. (WL #14268)

## Compilation Notes

- GCC 10 is now a supported compiler for building MySQL on EL7 or EL8. This compiler is available in the `devtoolset-10` (EL7) or `gcc-toolset-10` (EL8) package. It is also recommended to use GCC 10 when building third-party applications that are based on the `libmysqlclient` C API library. (Bug #32381003)

## Connection Management Notes

- Previously, if a client did not use the connection to the server within the period specified by the `wait_timeout` system variable and the server closed the connection, the client received no notification of the reason. Typically, the client would see `Lost connection to MySQL server during query` (`CR_SERVER_LOST`) or `MySQL server has gone away` (`CR_SERVER_GONE_ERROR`).

  In such cases, the server now writes the reason to the connection before closing it, and client receives a more informative error message, `The client was disconnected by the server because of inactivity. See wait_timeout and interactive_timeout for configuring this behavior.` (`ER_CLIENT_INTERACTION_TIMEOUT`).

  The previous behavior still applies for client connections to older servers and connections to the server by older clients. (WL #12999)

## Error Handling

- Client connection failure messages now include the port number. For example: `Can't connect to MySQL server on '127.0.0.1:63333'.`. Thanks to Daniël van Eeden for the contribution. (Bug #30787660, Bug #98311)

## Keyring Notes

- MySQL Keyring previously implemented keystore capabilities using server plugins, but now is transitioning to use the MySQL component infrastructure, beginning with these keyring components:

- `component_keyring_file` stores keyring data in a file local to the server host. This component is available in MySQL Community Edition and MySQL Enterprise Edition distributions. See Using the component_keyring_file File-Based Keyring Component.

- `component_keyring_encrypted_file` stores keyring data in an encrypted, password-protected file local to the server host. This component is available in MySQL Enterprise Edition distributions. See Using the component_keyring_encrypted_file Encrypted File-Based Keyring Component.

The new keyring components have similarities to the existing `keyring_file` and `keyring_encrypted_file` plugins, but are configured differently, use distinct on-disk storage formats, and have fewer restrictions on key type and key size.

Keyring components are not loaded using the `--early-plugin-load` server option during startup or configured during startup or at runtime using system variables:

- During startup, the server determines which keyring component to load using a manifest file, and the loaded component consults its own configuration file when it initializes. See Keyring Component Installation.

- At runtime, the new `ALTER INSTANCE RELOAD KEYRING` statement enables reconfiguring an installed keyring component after changes to its configuration file. See ALTER INSTANCE Statement.

If a keyring component is installed, the new Performance Schema `keyring_component_status` table provides status information about it. See The keyring_component_status Table.

Key migration capabilities have been expanded. Previously, key migration occurred only from one keyring plugin to another. The new `--keyring-migration-to-component` server option enables key migration from a keyring plugin to a keyring component; this facilitates transitioning a MySQL installation from keyring plugins to keyring components. The new `mysql_migrate_keyring` utility enables key migration from one keyring component to another. See Migrating Keys Between Keyring Keystores. There is no provision for migrating keys from a keyring component to a keyring plugin.

Existing keyring plugins remain available with no change to user-visible characteristics, but their implementation was revised to use the component infrastructure. This is facilitated using the built-in plugin named `daemon_keyring_proxy_plugin` that acts as a bridge between the plugin and component service APIs. See The Keyring Proxy Bridge Plugin. (WL #13446)

## Optimizer Notes

- The MySQL query optimizer can now apply the derived table optimization to correlated scalar subqueries, whenever the `subquery_to_derived` flag of the `optimizer_switch` variable is enabled. This is done by applying an extra grouping and then an outer join on the lifted predicate. For example, a query such as `SELECT * FROM t1 WHERE (SELECT a FROM t2 WHERE t2.a=t1.a) > 0` can be rewritten as `SELECT t1.* FROM t1 LEFT OUTER JOIN (SELECT a, COUNT(*) AS ct FROM t2 GROUP BY a) AS derived ON t1.a = derived.a WHERE derived.a > 0`.

  If the subquery already has an explicit grouping, MySQL adds the extra grouping to the end of the existing grouping list.

  MySQL performs a cardinality check to ensure that the subquery does not return more than one row, and raises `ER_SUBQUERY_NO_1_ROW` if it does. The check is performed as part of evaluating any `WHERE` or `JOIN` clause in the rewritten query, before evaluating the lifted predicate.

  For more information, see Correlated Subqueries, as well as Derived Tables. (WL #13520)

## Packaging Notes

- The bundled `libedit` library was upgraded to version 20190324-3.1. (Bug #32433089)
- Binary packages that include `curl` rather than linking to the system `curl` library have been upgraded to use `curl` 7.74.0.

## Performance Schema Notes

- **Incompatible Change:** Instrumentation in the Performance Schema `setup_instruments` table was revised as follows:
  - There are new memory instruments to account for memory allocated to data dictionary infrastructure and objects:

    ```
    memory/sql/dd::infrastructure
    memory/sql/dd::objects
    ```

  - Some instruments were renamed for improved uniformity of instrument naming. Affected instrument names are grouped using a key prefix in the same style as C++ namespaces. For example, error-related instruments use a `error::` prefix and partition-related instruments use a `Partition::` prefix. The following table displays the affected instruments.

    | Old Instrument Name | New Instrument Name |
    |---|---|
    | `debug_sync_control::debug_sync_action` | `THD::debug_sync_action` |
    | `errmsgs` | `errmsgs::server` |
    | `handler::errmsgs` | `errmsgs::handler` |
    | `handlerton` | `handlerton::objects` |
    | `log_error_loaded_services` | `log_error::loaded_services` |
    | `log_error_stack` | `log_error::stack` |
    | `LOG_name` | `LOG::file_name` |
    | `MDL_context_backup_manager` | `MDL_context::backup_manager` |
    | `Partition_admin` | `Partition::admin` |
    | `Partition_share` | `Partition::share` |
    | `partition_sort_buffer` | `Partition::sort_buffer` |
    | `partition_syntax_buffer` | `Partition::syntax_buffer` |
    | `Prepared_statement_map` | `Prepared_statement::infrastructure` |
    | `prune_partitions::exec` | `Partition::prune_exec` |
    | `Recovered_xa_transactions` | `XA::recovered_transactions` |
    | `thd::main_mem_root` | `THD::main_mem_root` |
    | `XID` | `XA::transaction_contexts` |

    In addition, the `servers` instrument is a duplicate of `servers_cache` and has been removed.

    Applications that use the old or removed instrument names should be adjusted to account for this change.

- Several instruments were given a value in the `DOCUMENTATION` column (it is no longer `NULL`), improving runtime instrumentation documentation availability.

(WL #14079)

## Pluggable Authentication

- The new `caching_sha2_password_digest_rounds` system variable enables configuring the number of hash rounds used by the `caching_sha2_password` authentication plugin for password storage. (WL #14281)

## Security Notes

- For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.1.1k. Issues fixed in the new OpenSSL version are described at https://www.openssl.org/news/cl111.txt and https://www.openssl.org/news/vulnerabilities.html. (Bug #32680637)

## Spatial Data Support

- The new `ST_LineInterpolatePoint()` and `ST_LineInterpolatePoints()` functions return a single point or multiple points interpolated between the start point and endpoint of a `LineString` geometry. The new `ST_PointAtDistance()` function returns a single point a given distance along a `LineString` geometry. See Spatial Operator Functions.

  The new `ST_Collect()` aggregate function takes multiple geometry arguments and produces from them a single geometry collection value. See Spatial Aggregate Functions

  The `CAST()` and `CONVERT()` functions have been extended to support casting geometry values from one spatial type to another. See Cast Functions and Operators. (WL #13454, WL #14160, WL #14161, WL #14178, WL #14127)

## Functionality Added or Changed

- **InnoDB:** The `AUTOEXTEND_SIZE` maximum setting was increased from 64M to 4GB. The `AUTOEXTEND_SIZE` option, introduced in MySQL 8.0.23, defines the amount by which `InnoDB` extends the size of a tablespace when it becomes full. The option is supported with the `CREATE TABLE`, `ALTER TABLE`, `CREATE TABLESPACE`, and `ALTER TABLESPACE` statements. For more information, see Tablespace AUTOEXTEND_SIZE Configuration. (Bug #32438606)

- **Group Replication:** It is now possible to use `START REPLICA SQL_THREAD` and `STOP REPLICA SQL_THREAD` statements for the `group_replication_applier` channel when Group Replication is stopped. This enables an operator to apply any remaining unapplied transactions on a server that left the group, without having to rejoin the server to the group. (Bug #32027612, Bug #32414767)

- **Group Replication:** Group Replication's allowlist of hosts from which an incoming Group Communication System connection can be accepted, can now be updated while Group Replication is still running. You can therefore add new members to a group controlled by an allowlist without needing to stop and restart Group Replication. The allowlist is specified by the `group_replication_ip_allowlist` system variable on each group member. (WL #14563)

- **Microsoft Windows:** The `/RTC1` compiler flag was removed from the debug compiler flags to reduce the time needed to run all tests invoked by the `mysql-test-run.pl` script when using MSVC on Windows. Initially, this compiler flag was introduced with the `CMake` program (by default for debug builds) and it generated code to test for stack corruption around function calls and the use of uninitialized variables at runtime. No loss of bug discovery is expected with the removal of the `/RTC1` compiler flag. Memory errors, such as stack corruption, are more likely to be found by Address Sanitizer (ASAN) testing and the use of uninitialized variables are detected by compiler warnings. (Bug #32525732)

- **Microsoft Windows:** The `named_pipe_full_access_group` system variable now defaults to an empty string (`''`), making named pipe connections secure. Previously, `'*everyone*'` was the default value. A valid Windows local group name may be substituted. (WL #12671)

- The new `clone_donor_timeout_after_network_failure` Clone plugin variable defines the amount of time the donor allows for the recipient to reconnect and restart a cloning operation following a network failure. The timeout was previously fixed at 5 minutes. The timeout can now be extended up to 30 minutes to provide more time for network issues to be resolved.

  Thanks to Daniël van Eeden for the contribution. (Bug #32335434, Bug #102103)

- Client applications and test suite plugins now report `utf8mb3` rather than `utf8` when writing character set names. (Bug #32164079, Bug #32164125)

- The `--skip-slave-start` command line parameter is used to prevent the replication I/O thread and replication SQL thread from starting when a replica server restarts. However, using the parameter on the command line or in a `my.cnf` option file might require login access to the server host. The `skip_slave_start` system variable is now provided to give access to this feature using MySQL Server's privilege structure, so that database administrators do not need any privileged access to the operating system. The new global system variable is read-only and can be set using a `SET PERSIST_ONLY` statement. As a system variable, its value can also be queried from a MySQL client and used by MySQL APIs. The `--skip-slave-start` command line parameter can still be used as an alternative, and it sets the new global system variable. (WL #14450)

## Bugs Fixed

- **Important Note:** When a `utf8mb3` collation was specified in a `CREATE TABLE` statement, `SHOW CREATE TABLE`, `DEFAULT CHARSET`, the values of system variables containing character set names, and the binary log all subsequently displayed the character set as `utf8` which is becoming a synonym for `utf8mb4`. Now in such cases, `utf8mb3` is shown instead, and `CREATE TABLE` raises the warning `'collation_name' is a collation of the deprecated character set UTF8MB3. Please consider using UTF8MB4 with an appropriate collation instead`. (Bug #27225287, Bug #32085357, Bug #32122844)

  References: See also: Bug #30624990.

- **InnoDB:** Stale pages encountered by the `buf_page_create()` function were freed and then read from disk again without being reinitialized. (Bug #32622548)

- **InnoDB:** C++ `enum` type values in buffer pool page and buffer pool block data structures in the `InnoDB` source code were changed from `int` to `uint8_t` in MySQL 8.0.23, causing the data to be printed as `ascii` characters instead of integers in different outputs including messages and `INFORMATION_SCHEMA` tables.

  The function that populates the `INNODB_BUFFER_PAGE_LRU` table did not print the `IO_PIN` state for the `IO_FIX` field. (Bug #32575469)

- **InnoDB:** The `btr_pcur_t::copy_stored_position` function, which copies the position of one index tree persistent cursor to another, performed unnecessary free and malloc calls. (Bug #32574835)

- **InnoDB:** A function that checks for locks on a table tried to acquire an exclusive lock system (`lock_sys`) latch, causing long semaphore waits. (Bug #32545030)

- **InnoDB:** Counting temporary tablespaces as open files caused the `innodb_open_files` limit to be exceeded, preventing other files from being opened. Temporary tablespaces are now ignored when counting open files. (Bug #32541241)

- **InnoDB:** The `buf_flush_ready_for_flush()` function identified a stale page as ready to flush without first checking if the page was dirty or I/O fixed, causing an assertion failure. (Bug #32530147)
- **InnoDB:** Rollback of a transaction that modified generated columns raised an assertion failure. The failure occurred when attempting to free space occupied by externally stored columns. The update vector containing the externally stored columns did not account for the generated columns. (Bug #32529561)
- **InnoDB:** When starting the server with `innodb_validate_tablespace_paths` disabled, user-created undo tablespaces, which are temporarily set to inactive during the initial stages of the startup process, were not reactivated. (Bug #32493885)
- **InnoDB:** Disabling `innodb_extend_and_initialize`, which causes `InnoDB` to use `fallocate()` when allocating space to tablespace files, caused a performance regression for insert-oriented workloads. (Bug #32437020)
- **InnoDB:** An unnecessary full flush list scan was removed, improving the speed with which session temporary tablespaces are created. Previously, depending on the size of the buffer pool and the number of dirty pages, creation of session temporary tablespaces could take a long time, affecting write transaction performance. (Bug #32423860)
- **InnoDB:** A function that checks if a given page type is valid raised an assertion when testing valid but undefined page type for an undo tablespace. (Bug #32366301)
- **InnoDB:** Sharded read-write lock function instrumentation was improved. (Bug #32333168)
- **InnoDB:** On Windows, when creating a table with the `COMPRESSION` option, `InnoDB` failed to check the error status of a system call before checking the punch hole support flag set by the same system call, which resulted in reading an uninitialized flag. (Bug #32322645)
- **InnoDB:** Geometry columns created in MySQL 5.6 caused failure when restarting the server after upgrading from MySQL 5.7 to MySQL 8.0 due to a geometry type change introduced in MySQL 5.7. (Bug #32299738)
- **InnoDB:** During recovery, log records for multiple record group mini transactions were parsed twice. The log record from the first parsing operation is now saved to avoid the second pass, which improves recovery performance.

  Thanks to Zhai Weixiang for the contribution. (Bug #32293797, Bug #102010)
- **InnoDB:** In debug builds, an end range condition check for a secondary index raised an assertion failure, and Valgrind testing uncovered an end range condition check for secondary index on a virtual column that read an uninitialized value. (Bug #32291506)
- **InnoDB:** With the `innodb_log_writer_threads` variable disabled, extensive log writing by one thread could block log flushing opportunities for other threads. (Bug #32255538)
- **InnoDB:** In `SHOW ENGINE INNODB STATUS` deadlock information, in cases where a single `rw-lock` reader thread holds a shared latch, the reader thread id was not printed. (Bug #32252477)
- **InnoDB:** Running concurrent `SHOW CREATE TABLE` and `ALTER TABLE` operations on the same table raised an assertion failure. The `SHOW CREATE TABLE` thread referenced a tablespace object that had been made stale by the `ALTER TABLE` operation. (Bug #32235621)
- **InnoDB:** The introduction of sharded `rw_lock_stats` counters in MySQL 5.7 caused a regression in CPU cache efficiency. To address this issue in MySQL 5.7, the sharding method was changed. For optimal performance, the `rw_lock_stats` counter is removed in MySQL 8.0. (Bug #32225367)

- **InnoDB:** On Windows, stalls were caused by concurrent `SELECT COUNT(*)` queries where the number of parallel read threads exceeded the number of machine cores. (Bug #32224707, Bug #101789)
- **InnoDB:** An `IS_STALE` column, which indicates whether a buffer pool page is stale, was added to the `INFORMATION_SCHEMA.INNODB_BUFFER_PAGE` table. (Bug #32194434)
- **InnoDB:** In debug builds, the `log_free_check()` function in the redo log code, which ensures that there is available space in the logs when holding latches on dirty pages, now uses a list of mini-transactions opened by the current thread to search for potential constraint violations. (Bug #32189367)
- **InnoDB:** During a slow shutdown on a system with a 64k `InnoDB` page size and `innodb_max_undo_log_size` setting that was less than the initial undo tablespace size, the two undo tablespaces were truncated in an endless loop. Undo tablespace truncation is now restricted to undo tablespaces that are larger than the initial undo tablespace size and the `innodb_max_undo_log_size` setting.

  The function that retrieves the next redo rollback segment for an undo tablespace was modified to ensure that undo logs are evenly split between two active undo tablespaces when an inactive tablespace is found. (Bug #32173457)
- **InnoDB:** Starting the server in upgrade mode following an unexpected stoppage while the `undo_001` undo tablespace was being truncated caused failure. When upgrade processing was completed and shutdown initiated, the function that determines if undo truncation is required could not find the `undo_001` undo tablespace. To address this issue, undo tablespace truncation is no longer performed when starting the server in upgrade mode. (Bug #32127912)
- **InnoDB:** A delete operation on a parent table that initiated a cascading update on a child table with an indexed virtual column and indexed foreign key constraint column caused a virtual column corruption. (Bug #32124113)
- **InnoDB:** The open and close sequence for table share instances (`m_share` objects) and dictionary table instances was modified to prevent accessing old `m_share` objects that could point to stale dictionary indexes.

  Thanks to Yuxiang Jiang for the contribution. (Bug #31899685)
- **InnoDB:** An debug assertion failure occurred when issuing a `TRUNCATE TABLE` operation after a successful server restart following a server initialization failure. (Bug #31763837)
- **InnoDB:** A fast shutdown (`innodb_fast_shutdown=0`) raised an assertion failure. The assertion code checks for transactions in the transaction list (`mysql_trx_list`) before halting purge operations. The check occurred before shutting down the GTID persister background thread, which was still inserting transactions into the transaction list. (Bug #31693627)
- **InnoDB:** Undo tablespace truncation error handling was improved, and a Windows-specific `InnoDB` file deletion procedure was made more POSIX-compatible. (Bug #31684783)
- **InnoDB:** An online `ALTER TABLE` operation failed with an `Incorrect key file for table` error due to an unnecessary encryption status check that was performed when reading online DDL row logs. (Bug #31529221, Bug #99938)
- **InnoDB:** Initializing the server with an with a 4K `InnoDB` page size caused “key too long” errors to be written to the error log. (Bug #31496943, Bug #99892)
- **InnoDB:** A table created with a reserved name caused a failure. (Bug #31382599)
- **InnoDB:** Creating an index on virtual columns raised an invalid debug assertion failure. (Bug #31279528, Bug #99421, Bug #27168185)

- **InnoDB:** In debug mode, the `buf_pool_validate_instance()` function accessed the `buf_page_t::io_fix` field without properly synchronizing with a thread that was modifying the same field in the `buf_page_io_complete()` function, causing an assertion failure. In addition to correcting that issue, the `buf_page_get_io_fix_unlocked()` function was replaced by several specific functions that test the `buf_page_t::io_fix` field for particular values. A series of other small, related code changes were also implemented. (Bug #31027553)
- **InnoDB:** The TempTable memory allocator did not track RAM consumption when allocating blocks of memory shared by different queries for a given session, which could result in nonadherence to the `temptable_max_ram` limit. (Bug #29890126)
- **Partitioning:** In some cases, invalid `PARTITION` clauses were not handled correctly in `ALTER TABLE` statements. (Bug #32235085)
- **Replication:** With row-based replication in use (`binlog_format=ROW`), stored generated columns that existed only on the replica were not being updated when the other fields in the row were inserted or updated. These columns are now evaluated when the row image is unpacked, and their value is updated in the row image before it is applied. (Bug #32292413, Bug #101978)
- **Replication:** Binary log transaction compression could not continue if a row event included a `BLOB` column containing uncompressable data, and the compressed size of the row event was higher than its uncompressed size. The function now handles the additional post-compression bytes correctly. (Bug #32174715, Bug #101611)
- **Replication:** If all previous binary log files were purged at startup because their retention period had expired, the new binary log file contained an empty `Previous_gtids` event, which could cause errors in replication. The order of initialization has now been changed so that previous binary log files are only purged after the previous GTID set has been written to the new binary log file that is created at startup. (Bug #32134875, Bug #101533)
- **Replication:** When MySQL Server counted the number of GTIDs in a set, it was possible for the return value to wrap, returning an incorrect result. This could lead to an incorrect decision to use state transfer from the binary log for Group Replication distributed recovery, when a remote cloning operation would have been more efficient. The logic has now been corrected. (Bug #32086209)
- **Replication:** An assertion was raised in debug builds relating to lost GTIDs if binary log files were removed at startup because their retention period had expired. (Bug #32008512, Bug #101137)
- **Replication:** The `sql_slave_skip_counter` system variable, which is used to make a replica skip a specified number of transactions, ignored transaction payloads that had been compressed using binary log transaction compression (activated using the `binlog_transaction_compression` system variable). Only events in uncompressed transaction payloads were counted towards the total skipped. (Bug #31973055)
- **Replication:** A deadlock could occur if the binary log file was rotated while system variables were being updated and read by different clients. (Bug #31774422)
- **Replication:** The output of a `SHOW PROCESSLIST` statement for a replica's SQL thread sometimes showed the last query as currently being applied when the replica was actually caught up. (Bug #30521198, Bug #97560)
- **Group Replication:** A race condition could occur if a `STOP GROUP_REPLICATION` statement was used to stop a group member, while the Performance Schema statistics for Group Replication were being queried by another client. Group Replication now prevents `STOP GROUP_REPLICATION` statements and Performance Schema queries from running concurrently. (Bug #32100147)
- **Microsoft Windows:** Running MySQL Server (64-bit) on a Windows system with more than 32 logical processors and setting the VCPU attribute of a resource group to greater than 30 produced

an incorrect CPU mask value, which is used to set the thread affinity. Under these conditions, the MSVC compiler reported warnings that the 32-bit shift was converted implicitly to 64-bit in the `thread_attrs_api_win.cc` file. The conversion resulted in the wrong CPU mask calculation on systems that have more than 32 logical processors. This fix ensures that a 64-bit shift is used when calculating the CPU mask by replacing argument `1` of the shift with `1LL`. (Bug #32079726)

- **JSON:** The `IF()` function sometimes hit an assertion in debug builds when an error was raised from its first argument; this also could occur under similar circumstances with the additional condition that the function's return type was `JSON`. (Bug #32231393, Bug #32231620)

- **JSON:** A number of JSON functions did not propagate errors correctly, which could lead to assert failures in debug builds. (Bug #32047630)

- **JSON:** `JSON_TABLE()` inside triggers was sometimes handled incorrectly when re-used in different sessions. Fixed by ensuring that `JSON_TABLE()` and its temporary table are processed in the context of the current session. (Bug #31644193)

- **JSON:** A multi-valued index defined on an expression that cast a column into a typed array was not used for speeding up queries. This was because the server, when substituting an expression with an equivalent indexed generated column, did not attempt to replace a reference to a column with a reference to an equivalent generated column; for multi-valued indexes, it makes sense to replace the column reference with a reference to a generated column backing a multi-valued index on an expression that cast that column into a typed array.

  This fix lifts the restriction that the server attempts to substitute only function expressions and conditional expressions with indexed generated columns, by allowing substitution of column references when they appear in a context where they can make use of a multi-valued index, that is, when they are used as arguments to `MEMBER OF()`, `JSON_CONTAINS()`, or `JSON_OVERLAPS()`. The restriction remains in force for cases in which the column reference is used in a non-array context. (Bug #30838807)

- **JSON:** The `JSON_SEARCH()` function interpreted all search string and path values as `utf8mb4` strings, regardless of their actual encoding, which could lead to wrong results. (Bug #102443, Bug #32443143)

- **JSON:** In some cases, when used in a left join, some MySQL JSON functions caused the optimizer to transform it into an inner join, even though the inner join was not equivalent to the original left join. This was due to the fact that they return a value which is not `NULL`, even though one of their arguments is `NULL`, and the optimizer expected them to return `NULL` on `NULL` input.

  Functions affected by this issue included `JSON_ARRAY()`, `JSON_OBJECT()`, `JSON_ARRAY_APPEND()`, `JSON_ARRAY_INSERT()`, `JSON_INSERT()`, `JSON_REPLACE()`, and `JSON_SET()`. See JSON Functions, for more information about these functions. (Bug #101861, Bug #32248705)

- Community packages for generic Linux were not built with the necessary LDAP/SASL/Kerberos dependencies, and did not bundle the associated required libraries. (Bug #32619858)

- The functions `BIT_AND()`, `BIT_OR()`, `BIT_XOR()`, and `JSON_ARRAYAGG()` did not always provide proper error handling. (Bug #32594813)

- The server did not always provide proper error messages when `IN` was incorrectly used with `UNION`. (Bug #32593846)

- It was possible for casting from `DECIMAL` to a signed integer type to hit an assertion when the result had more digits than the metadata in `Item_typecast_signed` indicated it would have. This happened because the decimal value is rounded when converted to an integer, and `Item_typecast_signed` did not take into account that the rounding might increase the number of digits in what was the integer part of the decimal value, such as when rounding 9.9 up to 10.

This is fixed by removing the logic that tried to compute a minimal maximum length for the result of the cast, and instead using the default set by the parent class `Item_int_func`. That default is the maximum width of a 64-bit integer, which should be safe regardless of the input value. (Bug #32591589)

References: This issue is a regression of: Bug #32371039.

- The name `my_row_id` was not permitted for invisible columns. This restriction has been lifted. (Bug #32586231)
- On a system with many concurrent connections, execution of grant statements could take excessively long waiting for metadata locks, causing the server to become unresponsive. (Bug #32483597)
- Windows binaries and libraries were not being properly signed. (Bug #32458533)
- Privilege checks for `INFORMATION_SCHEMA` tables could fail to take the `lower_case_table_names` value into account. (Bug #32448364)
- `LIKE ... ESCAPE`, where `ESCAPE` did not reference a constant value, was not handled correctly within a prepared statement. (Bug #32446728)
- MySQL has traditionally interpreted an empty value in the `ESCAPE` clause (that is, `ESCAPE ''`) for `LIKE` as “no escape character”. Problems could arise when either of the first two arguments to `LIKE` was a string using a multibyte character set, because the empty value in such cases was interpreted as meaning that the backslash (`\`) should be used as the escape character, breaking the expected behavior.

  This fix causes `LIKE` to interpret `ESCAPE ''` as meaning that there is no escape character regardless of character set, restoring the previous, expected behavior.

  In addition, we now raise an error if the specified escape character cannot be converted to the target character set. This supersedes the original behavior in such cases, which was to fall back silently to using the backslash as the escape character. (Bug #32446508)
- Windows packages were subject to an OpenSSL vulnerability. (Bug #32431519)
- Loadable function arguments containing window functions or subqueries could produce unexpected results. (Bug #32424455)
- Improper handling of temporary tables used for cursors within stored procedures could result in unexpected server behavior. (Bug #32416811)
- Use of the symbol `TRUE` in the source resulted in a build failure on some platforms. This was replaced by `true`. (Bug #32406197, Bug #102308)
- The privilege check used to determine whether users can see view definitions in the `INFORMATION_SCHEMA.VIEWS` table worked incorrectly. (Bug #32405811)
- For recursive common table expressions, an assertion could be raised if it became necessary to convert an in-memory temporary table to on-disk. (Bug #32404597)
- A prepared statement employing a user-created function was not handled correctly when the function took no arguments. (Bug #32404542)
- A clone plugin installation failure could cause subsequent installation attempts to fail. (Bug #32402158, Bug #102240)
- Some internal functions used with temporal column types did not provide proper handling for `YEAR` values. (Bug #32395335)

- The internal constants `MAX_FLOAT_STR_LENGTH` = 12 and `MAX_DOUBLE_STR_LENGTH` = 22 represent the maximum lengths of strings representing `FLOAT` and `DOUBLE` values, respectively. The heuristic employed by the conversion routine `my_gcvt` for determining whether to use scientific notation sometimes generated strings that were longer than these.

  These long strings caused problems for some string conversion routines, since their actual lengths could exceed the expected maximums. We fix this by explicitly telling `my_gcvt` the desired length whenever we fetch a `FLOAT` or `DOUBLE` in a string context. (Bug #32385934)

- Within triggers, use of `RAND()` with no arguments could lead to unexpected server behavior. (Bug #32372805)

- When an `Item_cache` object was used in a prepared statement or stored procedure, it might point to a `Field` object that was used in a previous execution. We solve this issue by replacing the cached field member with an `Item_field`, which is persistent for the lifetime of the procedure. (Bug #32367019)

- A missing tablespace error was reported on the recipient MySQL server instance after a remote cloning operation. The tablespace was not cloned due to the `innodb_validate_tablespace_paths` variable being disabled on the donor instance, which resulted in the associated tablespace object not being loaded. A check is now performed before a cloning operation to ensure that all tablespace objects are loaded. (Bug #32354908, Bug #102137)

- Some query blocks containing large numbers of `EXISTS` subqueries were not always handled correctly. (Bug #32343143)

- `mysqlpump` could exit unexpectedly if a `SHOW CREATE TABLE` statement failed. (Bug #32340208)

- A long running remote cloning operation failed due to a low `wait_timeout` setting on the donor MySQL Server instance. Donor threads use the MySQL Server `wait_timeout` setting when listening for Clone protocol commands. To avoid timeout failures on donor instances with a low `wait_timeout` setting, the Clone idle timeout is now set to the default `wait_timeout` setting, which is 28800 seconds (8 hours). Clone network read and write timeout values were also increased. (Bug #32340112, Bug #102097)

- Replication threads running in the server were visible in the Performance Schema `threads` table, but failed to appear in the `variables_by_thread` or `status_by_thread` tables. Now they appear in all three tables. Thanks to Facebook for the contribution. (Bug #32335496, Bug #102115)

- A query string was displayed before it had been rewritten. (Bug #32335263, Bug #32628376)

- For builds compiled using the `libedit` library, if the `mysql` client was invoked with the `--default-character-set=utf8` option, libedit rejected input of multibyte characters. (Bug #32329078, Bug #32583436, Bug #102806)

- On Windows, large result sets could cause the `mysql` client to exit unexpectedly. (Bug #32316323, Bug #102051)

- Preparing a query expression for repeated execution could raise an assertion if an error occurred during preparation. (Bug #32291841)

- Functional index creation did not handle the column name as not case-sensitive. (Bug #32287186, Bug #101994)

- Temporary tables bound to triggers during statement execution could cause an unexpected server exit. (Bug #32267749, Bug #32288089, Bug #32299045)

- Improved `NULL` and error handling in calculations involving decimal values. (Bug #32258228, Bug #32497850)

- An assertion was raised if there was an open handler to a table in a schema when the schema was altered to be read only. (Bug #32248313)
- Selecting from the Performance Schema `replication_group_members` table in some cases raised Error 1267 `Illegal mix of collations ...`. This was due to a change in MySQL 8.0.22 in the handling of comparisons between a column value and a system variable. (Bug #32244631)

  References: See also: Bug #32501472, Bug #32579184.
- MySQL produced invalid metadata for a number of temporal functions returning integer values. These functions included `TO_DAYS()`, `PERIOD_DIFF()`, `PERIOD_ADD()`, `TO_SECONDS()`, `DAYOFMONTH()`, `DAYOFYEAR()`, `HOUR()`, and `MINUTE()`, among others. (Bug #32239578)
- Table subqueries of a natural join which retrieved only invisible columns were not handled correctly. (Bug #32235285)
- For debug builds, using `ALTER TABLE` to set a column to have a `DEFAULT` value of `TRUE` raised an assertion. (Bug #32235058)
- When an error was raised while evaluating a condition that was pushed down to the storage engine using index condition pushdown, the storage engine sometimes ignored the error and returned an error code indicating success, which could lead to assertion failures later.

  Now in such cases, we make sure that the executor detects that an error has been raised, and stop execution at that point. (Bug #32234773)
- For debug builds with binary logging disabled, `ALTER TABLE ... MODIFY COLUMN` with an invalid `DEFAULT` value raised an assertion. (Bug #32234194)
- Preparation of an aggregate function sometimes hit an assertion in debug builds when the function took a constant scalar subquery as argument and the scalar subquery raised an error. (Bug #32231698)
- For debug builds, improper character set handling for `NULLIF()` evaluated in aggregate context raised an assertion. (Bug #32231557)
- A `CREATE TABLE` statement with an unsupported `KEY_BLOCK_SIZE` value raised an assertion. (Bug #32227101)
- Upgrading a MySQL instance with a very large number of tables consumed an excessive amount of memory. Memory allocated to analyze data dictionary entities for possible upgrade was not released until all entities were processed. (Bug #32226180, Bug #101818)
- `ANALYZE TABLE` executed on a table concurrently with a long-running query on the same table caused subsequent queries on the table to wait for the long-running query to finish. This wait induced by `ANALYZE TABLE` is now eliminated, thus allowing the subsequent queries to execute with no wait. (Bug #32224917)
- Statements using a `LIKE` expression with an `ESCAPE` clause were not always handled correctly. (Bug #32213959)
- On ARM platforms, an assertion could be raised in utilities used during the build process. (Bug #32209415)
- `InnoDB` did not always handle some legal names for table partitions correctly. (Bug #32208630)
- With the `explicit_defaults_for_timestamp` variable disabled, Clone plugin installation failed with PFS table creation error. The variable setting caused `TIMESTAMP` columns in the Clone Performance Schema tables to be created incorrectly as `NOT NULL` columns. (Bug #32199611)

- `SHOW CREATE VIEW` produced invalid syntax for views created with a `ROLLUP` clause. This issue also affected `mysqldump`, which uses `SHOW CREATE VIEW`. (Bug #32197353, Bug #101740)
- A failed `CREATE TABLE` or `ALTER TABLE` with `SECONDARY_ENGINE_ATTRIBUTE` could leak memory. (Bug #32187507)
- Connection establishment failure could cause the server to count the number of open connections incorrectly. (Bug #32156518)
- Refactoring work done in MySQL 8.0.19 did not handle left joins correctly on columns using functions such as `IFNULL()`. (Bug #32141711)

  References: This issue is a regression of: Bug #30320525.
- The optimizer could choose to use a Skip Scan even for backward index scans for which it is inapplicable, resulting in unpredictable server behavior. (Bug #32127290)
- `SHOW CREATE USER` caused any pending transaction to commit. (Bug #32123671)
- Loadable function arguments containing window functions were evaluated at prepare time despite the fact that window functions are set up quite late in the prepare process. Now evaluation of these is delayed until execution time, similarly to how this is performed with respect to loadable function arguments containing subqueries. (Bug #32122078, Bug #32393265)
- Creating a table containing a column with a nonconstant default expression caused subsequent `ALTER TABLE` statements to fail. (Bug #32121425, Bug #101486)
- Updating a `BLOB`-like column with a value from a larger `BLOB`-like column could cause the updated column to have the wrong size, even zero. (Bug #32112403)
- Improper locking on an internal queue could cause `mysqlpump` to exit unexpectedly. (Bug #32067013)
- On MacOS, the preference pane now includes an option to load the configured `my.cnf` file during initialization. (Bug #32057159)
- The server did not always prepare correctly a statement using `GROUP_CONCAT()` on a `SELECT` with an `ORDER BY` clause. (Bug #32053547, Bug #31947466)
- The `sys` schema `create_synonym_db()` procedure fail to create `INFORMATION_SCHEMA` synonyms. (Bug #32050275, Bug #101258)
- Errors occurring in window functions were not always correctly propagated, which could lead to assertion failures in debug builds. (Bug #32028154)
- Calling `XA COMMIT` on a transaction started by another thread could result in Address Sanitizer warnings. (Bug #32025408)
- When the `mysql` client was used in batch mode, its parser could be confused by `USE` followed by `DROP DATABASE` when the `USE` database name was quoted. (Bug #32015466, Bug #101124)
- A change in MySQL 8.0.17 caused `comp_err` to become much slower. Normal performance has been restored. (Bug #32014733)

  References: This issue is a regression of: Bug #29781631, Bug #95377.
- Using `CAST()` on a `DATE` or `DATETIME` value in an `INSERT` statement raised a warning for `"0000-00-00"` and `"2000-02-31"`, but not for `"2000-01-00"` or `"2000-00-01"`. Now a warning is shown in each of these cases also. (Bug #32013612)

- The maximum length of the `MASTER_COMPRESSION_ALGORITHMS` value for the `CHANGE MASTER TO` statement was checked incorrectly. (Bug #32008597)
- When casting a dynamic parameter to `YEAR` (such as in `PREPARE s FROM "SELECT CAST(? AS YEAR)"`), type propagation was not preformed, causing execution of the prepared statement in which the parameter was used to fail. (Bug #32002844)
- Definitions of some system tables for a MySQL 5.7 instance upgraded to MySQL 8.0 differed from the definitions of the system tables in a new MySQL 8.0 installation. (Bug #31989290)
- Some `SHOW` statements using subqueries could result in unexpected server behavior. (Bug #31853180)
- A query using `GROUP BY column HAVING column_expression IS [NOT] NULL` returned correct results, but when `WITH ROLLUP` was added did not. (Bug #31848191)
- The `SHOW ENGINE PERFORMANCE SCHEMA STATUS` statement reported incorrect memory usage for the Performance Schema. (Bug #31795132, Bug #100624)
- When trying to generate an entity data model using Visual Studio 2019, some tables could not be imported with the entity framework wizard. This was due to a change in MySQL 8.0.21 made to orthogonal data type aggregation, which handles columns from `UNION` and from functions or operators such as `CASE` and `IF()`. This makes it possible to return a value of type `ENUM` or `SET`, which did not need to be handled previously in such cases. (Bug #31750645)

  References: See also: Bug #29698617.
- While optimizing the `ORDER BY` clause of a subquery there was a possibility of cleaning up a subquery tree referenced in the outer `SELECT`, which could lead to a premature exit. (Bug #31721430)
- A malformed name in the `mysql.func` system table could cause unexpected server behavior. (Bug #31674599)
- `LOAD DATA` performance degraded if many long rows were loaded. (Bug #31637142)
- Sessions could disable their own auditing. (Bug #31630954)
- Compiler options for using profile guided optimization with GCC were improved to include `-fprofile-partial-training` and `-fprofile-update=prefer-atomic` when appropriate. Thanks to Dmitriy Philimonov for the suggestion. (Bug #31450064, Bug #99781)
- Starting the server with the `performance_schema_max_thread_classes` and `performance_schema_max_thread_instances` system variables set to zero caused MySQL Enterprise Firewall to be unable to create new rules. (Bug #31335080, Bug #24947654, Bug #83519)
- A null pointer was incremented during recovery from the redo log, causing a runtime error in an Undefined Behavior Sanitizer (UBSAN) build. (Bug #31173032, Bug #32428131, Bug #32483976)
- With the `log_slow_extra` system variable enabled to add the `Errno` field to slow query log output, the error number was 0 even for failed statements. (Bug #30769965, Bug #98220)
- On debug builds, certain conversion operations using the `utf32` character set could cause unexpected server behavior. (Bug #30746908)
- `SELECT ... FOR UPDATE` from a nonexistent Performance Schema table produced `ER_TABLEACCESS_DENIED_ERROR` rather than `ER_NO_SUCH_TABLE`. (Bug #30701047, Bug #98068)
- Mishandling of stored program local variables could lead to unexpected server behavior. (Bug #30366310)

- The Performance Schema `metadata_locks` table could show incorrect `DURATION` values, such as when a metadata lock taken for a `TRANSACTION` duration was later modified to `EXPLICIT` duration by a `RENAME TABLE` operation. (Bug #30065213, Bug #96237)

- The `audit_log` plugin could fail to store its encryption password if it generated one at startup. (Bug #29559793)

- Uninstalling a plugin could affect subsequent execution of prepared statements. (Bug #29363867)

- Conversion of string-valued user-defined variables or function results to double (for example, using `CAST()`) did not emit a warning when truncation occurred. (Bug #27969934, Bug #21943299)

- When a view definition used `LIKE` with an `ESCAPE` clause, the contents of the `ESCAPE` clause were ignored, leading to wrong results. (Bug #26086751)

- It was possible to insert illegal ASCII values (outside 7-bit range) into character columns that used the `ascii` character set. This is now prohibited. (Bug #24847620)

- To enable use of spaces and other special characters within configuration values, `mysql_config_editor` now surrounds values it writes to the configuration file with double quote characters, and also escapes double quote characters used within values. (Bug #19953349, Bug #74691)

- When the aggregate iterator finds no rows, it calls on each item in its `SELECT` list to inform them of this (for example, so that `COUNT(*)` can set itself to zero, or `SUM(foo)` can set itself to `NULL`). After internal work done in MySQL 8.0.22, it could also inadvertently call hidden items. In some queries with doubly nested subqueries, one such hidden item could become its own parent subquery (and scalar subqueries in MySQL have special legacy handling of this call, for queries which are not `ONLY_FULL_GROUP_BY`), causing the entire subquery to return `NULL` when it should not have done so.

  This is fixed by making the call only on visible items, as in MySQL 8.0.21 and earlier. (Bug #102101, Bug #32335256)

- When interpreting the old-style plan to access paths, cache invalidators for `LATERAL` were delayed until all outer joins were completed, since outer joins could produce null-complemented rows that should also invalidate caches. Problems arose when an outer join contained a `LATERAL`, and that `LATERAL` referred only to tables from within the same outer join; in such cases the invalidator should be applied immediately and not delayed, lest we miss emitted rows, and the cache be incorrectly kept. In particular, this could happen when certain Information Schema tables were on the right side of an outer join, as these are now views defined using `LATERAL`.

  We fix this by delaying emission of the invalidator until we are inside the same (outer) join nest as the materialization to be invalidated, but no further. This also deals correctly with the case where rows from a table should invalidate two or more separate materializations, where some are within the join and some are higher up. (Bug #101460, Bug #32113029, Bug #32311147)

  References: This issue is a regression of: Bug #98238, Bug #30766181.

- An optimizer trace printed floating-point numbers with a maximum six characters, which meant that the precision could be very low for many values, given that sign, decimal point, and exponent could use many of these characters. This was especially problematic for large numbers, whose precision could

thus be as small as 1, and which could be rounded to values whose absolute value exceeded `DBL_MAX` and so could be rejected by JSON parsers.

Now such numbers are always printed in an optimizer trace with a precision of 6. (Bug #101457, Bug #32113020)

References: See also: Bug #94672, Bug #29493604.

- Filesort was used for a query having an `ORDER BY ... DESC` clause, even when an index on the descending column was available and used. This happened because an `ORDER BY` sub-clause was not removed due to matching a field in an equality predicate, even though it should have, so that the optimizer did not match the query with the descending index, leading to suboptimal performance. (Bug #101220, Bug #32038406)
- The debug server hit an assert when `optimizer_search_depth` was less than the number of `JOIN_TAB` structures used for a join. (Bug #100288, Bug #31655483)
- Following the `ALTER TABLE` operations `EXCHANGE PARTITION`, `IMPORT TABLESPACE`, and `IMPORT PARTITION TABLESPACE`, serialized digital information reflecting the previous role of the tablespace was left behind. Now in such cases, the old SDI is explicitly removed from both tablespaces involved in the exchange or import of a tablespace. (Bug #98501, Bug #30878065)
- Type resolution performed by the integer division operator (`DIV`) yielded a precision of one less than expected in the result.

  Our thanks to Kaiwang Chen for the contribution. (Bug #96459, Bug #30156563)

# Changes in MySQL 8.0.23 (2021-01-18)

This release adds support for macOS 11 (Big Sur).


- Account Management Notes
- C API Notes
- Compilation Notes
- Connection Management Notes
- Deprecation and Removal Notes
- Firewall Notes
- Optimizer Notes
- Performance Schema Notes
- Pluggable Authentication
- Security Notes
- Spatial Data Support
- SQL Syntax Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed

## Account Management Notes

- Granting the `RELOAD` privilege enables a user to perform a wide variety of operations. In some cases, it may be desirable for a user to be able to perform only some of these operations. To enable DBAs to avoid granting `RELOAD` and tailor user privileges more closely to the operations permitted, these new privileges of more limited scope are available:

  - `FLUSH_OPTIMIZER_COSTS`: Enables use of the `FLUSH OPTIMIZER_COSTS` statement.
  - `FLUSH_STATUS`: Enables use of the `FLUSH STATUS` statement.
  - `FLUSH_TABLES`: Enables use of the `FLUSH TABLES` statement.
  - `FLUSH_USER_RESOURCES`: Enables use of the `FLUSH USER_RESOURCES` statement.

  The new privileges apply only at the global level. For more information, see Privileges Provided by MySQL, and FLUSH Statement.

  The `mysql_refresh()` C API function performs operations similar to those of various `FLUSH` statements, but is unaffected by this change. It still requires the `RELOAD` privilege regardless of the operation for which it is invoked. (WL #14303)

## C API Notes

- For some applications, it may be useful to define metadata on a per-query basis. Examples include the URL of the page that produced a query, or extra processing information to be passed with a query for use by a plugin such as an audit plugin or query rewrite plugin. MySQL now supports this capability without the use of workarounds such as specially formatted comments included in query strings:

  - On the client side, the `mysql_bind_param()` C API function enables defining query attributes. These attributes apply to the next SQL statement sent to the server for execution. Additionally, the `mysql` and `mysqltest` clients have a `query_attributes` command that enables defining query attributes.
  - On the server side, a component service provides access to query attributes. A component named `query_attributes` uses this service to implement a `mysql_query_attribute_string()` loadable function that enables obtaining attribute values within SQL statements. The `query_attributes` component is optional but must be installed for the function to be available.

  For more information, see Query Attributes.

  Thanks to Facebook for suggesting the idea (and for contributing code, although it was not used). (Bug #27855905, Bug #28686334, WL #12542)

## Compilation Notes

- Thanks to Tzachi Zidenberg, who contributed a patch for compiling MySQL on aarch64 (ARM64). (Bug #31815236, Bug #100664)

## Connection Management Notes

- Selection of the account that matches incoming TCP client connections could be affected by account creation order. To make the matching algorithm more deterministic, matching the host name part of accounts now checks accounts specified using host IP addresses, in a specific order, before attempting to match accounts specified using host names. Host name matching remains unchanged. See Access Control, Stage 1: Connection Verification. (WL #14074)

## Deprecation and Removal Notes

- From MySQL 8.0.23, the statement `CHANGE MASTER TO` is deprecated. The alias `CHANGE REPLICATION SOURCE TO` should be used instead. The parameters for the statement also have aliases that replace the term `MASTER` with the term `SOURCE`. For example, `MASTER_HOST` and `MASTER_PORT` can now be entered as `SOURCE_HOST` and `SOURCE_PORT`. The `START REPLICA | SLAVE` statement's parameters `MASTER_LOG_POS` and `MASTER_LOG_FILE` now have aliases `SOURCE_LOG_POS` and SOURCE_LOG_FILE. The statements work in the same way as before, only the terminology used for each statement has changed. A deprecation warning is issued if the old versions are used.

  A new status variable, `Com_change_replication_source`, has been added as an alias for the `Com_change_master` status variable. Both the old and new version of the statement update both the old and new version of the status variable.

  The server rewrites all `CHANGE MASTER TO` statements as `CHANGE REPLICATION SOURCE TO` statements in the query log. The same is done for the statements `START SLAVE`, `STOP SLAVE`, `SHOW SLAVE STATUS`, `SHOW SLAVE HOSTS` and `RESET SLAVE`. The event name for the `CHANGE MASTER TO` statement is set to `statement/sql/change_replication_source` in the statement history table. (Bug #32145023, WL #14189)

- The `gen_blacklist()` user-defined function is deprecated. Use `gen_blocklist()` instead, which performs the same term-replacement operation. (WL #14176)

- The use of the system variables `master_info_repository` and `relay_log_info_repository` is now deprecated, and a warning message is issued if you attempt to set them or read their values. The system variables will be removed in a future MySQL version. These system variables were used to specify whether the replica's connection metadata repository and applier metadata repository were written to an `InnoDB` table in the mysql system database, or to a file in the data directory. The `FILE` setting was already deprecated in a previous release, and tables are the default for the replication metadata repositories in MySQL 8.0. (WL #13958)

- Flushing the host cache can be done using any of these methods:

  - Execute a `TRUNCATE TABLE` statement that truncates the Performance Schema `host_cache` table. This requires the `DROP` privilege for the table.

  - Execute a `FLUSH HOSTS` statement. This requires the `RELOAD` privilege.

  - Execute a `mysqladmin flush-hosts` command. This requires the `RELOAD` privilege.

  Although those methods are equivalent in effect, granting the `RELOAD` privilege enables a number of other operations in addition to host cache flushing, which is undesirable from a security standpoint. Granting the `DROP` privilege for the `host_cache` table is preferable because it has a more limited scope. Therefore, the `FLUSH HOSTS` statement is deprecated and will be removed in a future MySQL version. Instead, truncate the `host_cache` table.

  `mysqladmin flush-hosts` previously executed a `FLUSH HOSTS` statement. Now it attempts to truncate the `host_cache` table, falling back to `FLUSH HOSTS` only if the truncate operation fails. (WL #14329)

## Firewall Notes

- Administrators perform MySQL Enterprise Firewall management by registering profiles that specify sets of rules for permitted statements (allowlists). Previously, profiles could be associated only with individual accounts, so that, to apply a given allowlist to multiple account profiles, it was necessary to duplicate the

rule set for each profile. For easier administration and greater flexibility, the firewall now provides group profile capabilities:

- Named group profiles can be created. A group profile can include multiple accounts as members, and an account can be a member of multiple group profiles.
- Each group profile has its own allowlist. The profile allowlist applies to all member accounts, eliminating the need to duplicate it across multiple account profiles.

For more information, see MySQL Enterprise Firewall. (WL #11740)

## Optimizer Notes

- Switched the hash table used for hash joins from an unordered multimap to an unordered flat map implemented with a multimap adapter. This change yields the following improvements:
  - A faster hash table
  - Less memory usage due to less hash table overhead, less space used for alignment and key/value lengths, and better memory usage with many equal keys; this should also reduce the frequency at which it is necessary to spill to disk
  - Better memory control by approaching the allowed join buffer size more closely rather than being effectively limited to approximately 2/3 of `join_buffer_size`, and by making it possible to free old memory when the hash table grows

  (Bug #99933, Bug #31516149, WL #13459)

## Performance Schema Notes

- Performance Schema macros that previously expanded to dynamic calls now expand to static calls when possible to reduce processing overhead. (Bug #32028160)
- Performance overhead of timer code was reduced. This should be of most benefit to workloads with high concurrency using the Performance Schema. Thanks to Georgy Kirichenko for the contribution. (Bug #31960377, Bug #101018)

## Pluggable Authentication

- The MySQL Enterprise Edition SASL LDAP authentication plugin now supports `SCRAM-SHA-256` as an authentication method for MySQL clients and servers. `SCRAM-SHA-256` is similar to `SCRAM-SHA-1` but is more secure. Use of `SCRAM-SHA-256` requires an OpenLDAP server built using Cyrus SASL 2.1.27 or higher. See LDAP Authentication Methods. (WL #14180)

## Security Notes

- For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.1.1i. Issues fixed in the new OpenSSL version are described at https://www.openssl.org/news/cl111.txt and https://www.openssl.org/news/vulnerabilities.html. (Bug #32260610)

## Spatial Data Support

- The new `ST_HausdorffDistance()` and `ST_FrechetDistance()` functions return the discrete Fréchet and Hausdorff distances between two geometries, reflecting how similar the geometries are. See Spatial Relation Functions That Use Object Shapes. (WL #14128, WL #14129)

## SQL Syntax Notes

- MySQL now supports invisible columns, which normally are hidden to queries, but can be accessed if explicitly referenced. See Invisible Columns. (WL #10905)

## X Plugin Notes

- For X Protocol connections using the `MYSQL41` authentication method, if the nonce sent by the server was shorter than 20 bytes, the connection logic did not handle it correctly. (Bug #32036194)
- If a query that was building up a resultset was killed, X Plugin interpreted this as meaning the server session had been killed, and dropped the connection. The status of a query is now checked separately from the status of the server session. (Bug #31954296)
- A deadlock could occur if an X Protocol session attempted to display X Plugin status variables or settings at the same time as another X Protocol session was being released and reset. The situation is now handled appropriately. (Bug #31931873)
- If an X Protocol client with a connection to a server remains idle (not sending to the server) for longer than the relevant X Plugin timeout setting (read, write, or wait timeout), X Plugin closes the connection. In the case of a read timeout, the plugin returns a warning notice with the error code `ER_IO_READ_ERROR` to the client application.

  From MySQL 8.0.23, X Plugin now also sends a warning notice if a connection is actively closed due to a server shutdown, or by the connection being killed from another client session. In the case of a server shutdown, the warning notice is sent to all authenticated X Protocol clients with open connections, with the `ER_SERVER_SHUTDOWN` error code. In the case of a killed connection, the warning notice is sent to the relevant client with the `ER_SESSION_WAS_KILLED` error code, unless the connection was killed during SQL execution, in which case a fatal error is returned with the `ER_QUERY_INTERRUPTED` error code.

  Client applications can use the warning notices to display to users, or to analyze the reason for disconnection and decide whether to attempt reconnection to the same server, or to a different server. (WL #14166)
- For classic MySQL protocol, if an SQL query is using metadata locking or the sleep function, the connection to the server is checked periodically to verify that it is still alive. If not, the query can be stopped so that it does not continue to consume resources. Previously, X Protocol did not carry out these checks, and assumed that the connection was still alive. The check has now been added for X Protocol. (WL #14167)

## Functionality Added or Changed

- **InnoDB:** Performance was improved for the following operations:
  - Dropping a large tablespace on a MySQL instance with a large buffer pool (>32GBs).
  - Dropping a tablespace with a significant number of pages referenced from the adaptive hash index.
  - Truncating temporary tablespaces.

  The pages of dropped or truncated tablespaces and associated AHI entries are now removed from the buffer pool passively as pages are encountered during normal operations. Previously, dropping or truncating tablespaces initiated a full list scan to remove pages from the buffer pool immediately, which negatively impacted performance. (Bug #31008942, Bug #98869, WL #14100)

- **InnoDB:** The new `AUTOEXTEND_SIZE` option defines the amount by which `InnoDB` extends the size of a tablespace when it becomes full, making it possible to extend tablespace size in larger increments. Allocating space in larger increments helps to avoid fragmentation and facilitates ingestion of large amounts of data. The `AUTOEXTEND_SIZE` option is supported with the `CREATE TABLE`, `ALTER TABLE`, `CREATE TABLESPACE`, and `ALTER TABLESPACE` statements. For more information, see Tablespace AUTOEXTEND_SIZE Configuration.

  An `AUTOEXTEND_SIZE` size column was added to the `INFORMATION_SCHEMA.INNODB_TABLESPACES` table. (WL #13895)

- **InnoDB:** `InnoDB` now supports encryption of doublewrite file pages belonging to encrypted tablespaces. The pages are encrypted using the encryption key of the associated tablespace. For more information, see InnoDB Data-at-Rest Encryption. (WL #13775)

- **InnoDB:** `InnoDB` atomics code was revised to use C++ `std::atomic`. (WL #14235)

- **Group Replication:** The MySQL Server asynchronous connection failover mechanism now supports Group Replication topologies, by automatically monitoring changes to group membership and distinguishing between primary and secondary servers. When you add a group member to the source list and define it as part of a managed group, the asynchronous connection failover mechanism updates the source list to keep it in line with membership changes, adding and removing group members automatically as they join or leave. The new `asynchronous_connection_failover_add_managed()` and `asynchronous_connection_failover_delete_managed()` functions are used to add and remove managed sources.

  The connection is failed over to another group member in the following situations:

  - The currently connected source goes offline or leaves the group.
  - The currently connected source is no longer in the majority.
  - The currently connected source does not have the highest weighted priority in the group.

  For a managed group, a source's weight is assigned depending on whether it is a primary or a secondary server. So assuming that you set up the managed group to give a higher weight to a primary and a lower weight to a secondary, when the primary changes, the higher weight is assigned to the new primary, so the replica changes over the connection to it. This also applies to single (non- managed) servers, so the connection for a single server is also now failed over if another source server is available that has a higher weighted priority. (WL #14019)

- When invoked with the `--all-databases` option, `mysqldump` now dumps the `mysql` database first, so that when the dump file is reloaded, any accounts named in the `DEFINER` clause of other objects will already have been created. (Bug #32141046)

- Some overhead for disabled Performance Schema and `LOCK_ORDER` tool instrumentation was identified and eliminated. (Bug #32105698)

- For `BLOB` and `TEXT` columns that have a default value expression, the `INFORMATION_SCHEMA.COLUMNS` table and `SHOW COLUMNS` statement now display the expression. (Bug #31856459)

- For a multithreaded replica (where `slave_parallel_workers` is greater than 0), setting `slave_preserve_commit_order=1` ensures that transactions are executed and committed on the replica in the same order as they appear in the replica's relay log. Each executing worker thread waits until all previous transactions are committed before committing. If a worker thread fails to execute a transaction because a possible deadlock was detected, or because the transaction's execution

time exceeded a relevant wait timeout, it automatically retries the number of times specified by `slave_transaction_retries` before stopping with an error. Transactions with a non-temporary error are not retried.

The replication applier on a multithreaded replica has always handled data access deadlocks that were identified by the storage engines involved. However, some other types of lock were not detected by the replication applier, such as locks involving access control lists (ACLs) or metadata locking (for example, `FLUSH TABLES WITH READ LOCK` statements). This could lead to three-actor deadlocks with the commit order locking, which could not be resolved by the replication applier, and caused replication to hang indefinitely. From MySQL 8.0.23, deadlock handling on multithreaded replicas that preserve the commit order has been enhanced to mitigate these types of deadlocks. The deadlocks are not specifically resolved by the replication applier, but the applier is aware of them and initiates automatic retries for the transaction, rather than hanging. If the retries are exhausted, replication stops in a controlled manner so that the deadlock can be resolved manually. (Bug #107574, Bug #34291887, WL #13574)

- CRC calculations for binlog checksums are faster on ARM platforms. Thanks to Krunal Bauskar for the contribution. (Bug #99118, Bug #31101633, Bug #32163391)
- Replication channels can now be set to assign a GTID to replicated transactions that do not already have one, using the `ASSIGN_GTIDS_TO_ANONYMOUS_TRANSACTIONS` option of the `CHANGE REPLICATION SOURCE TO` statement. This feature enables replication from a source that does not use GTID-based replication, to a replica that does. For a multi-source replica, you can have a mix of channels that use `ASSIGN_GTIDS_TO_ANONYMOUS_TRANSACTIONS`, and channels that do not. The GTID can include the replica's own server UUID or a server UUID that you assign to identify transactions from different sources.

  Note that a replica set up with `ASSIGN_GTIDS_TO_ANONYMOUS_TRANSACTIONS` on any channel cannot be promoted to replace the replication source server in the event that a failover is required, and a backup taken from the replica cannot be used to restore the replication source server. The same restriction applies to replacing or restoring other replicas that use `ASSIGN_GTIDS_TO_ANONYMOUS_TRANSACTIONS` on any channel. The GTID set (gtid_executed) from a replica set up with `ASSIGN_GTIDS_TO_ANONYMOUS_TRANSACTIONS` is nonstandard and should not be transferred to another server, or compared with another server's `gtid_executed` set. (WL #12819)
- The new `temptable_max_mmap` variable defines the maximum amount of memory the TempTable storage engine is permitted to allocate from memory-mapped temporary files before it starts storing data to `InnoDB` internal temporary tables on disk. A setting of 0 disables allocation of memory from memory-mapped temporary files. For more information, see Internal Temporary Table Use in MySQL. (WL #14125)

## Bugs Fixed

- **InnoDB:** A `CREATE TABLE` operation that specified the `COMPRESSION` option was permitted with a warning on a system that does not support hole punching. The operation now fails with an error instead. (Bug #32174200)
- **InnoDB:** A MySQL DB system restart following an upgrade that was initiated while a data load operation was in progress raised an assertion failure. (Bug #32173596)
- **InnoDB:** An error message regarding the number of truncate operations on the same undo tablespace between checkpoints incorrectly indicated a limit of 64. The limit was raised from 64 to 50,000 in MySQL 8.0.22. (Bug #32151601, Bug #101601)
- **InnoDB:** `rw_lock_t` and `buf_block_t` source code structures were reduced in size. (Bug #32084500)

- **InnoDB:** An `InnoDB` transaction became inconsistent after creating a table using a storage engine other than `InnoDB` from a query expression that operated on `InnoDB` tables. (Bug #32079103)

- **InnoDB:** In some circumstances, such as when an existing gap lock inherits a lock from a deleted record, the number of locks that appear in the `INFORMATION_SCHEMA.INNODB_TRX` table could diverge from the actual number of record locks.

  Thanks to Fungo Wang from Alibaba for the patch. (Bug #32068538, Bug #101305)

- **InnoDB:** An off-by-one error in `Fil_system` sharding code was corrected, and the maximum number of shards (`MAX_SHARDS`) was changed to 69. (Bug #32052821, Bug #101260)

- **InnoDB:** The TempTable storage engine memory allocator allocated extra blocks of memory unnecessarily. (Bug #32018553)

- **InnoDB:** A `SELECT COUNT(*)` operation on a table containing uncommitted data performed poorly due to unnecessary I/O.

  Thanks to Brian Yue for the contribution. (Bug #31997733, Bug #100966)

- **InnoDB:** A race condition when shutting down the log writer raised an assertion failure. (Bug #31997362)

- **InnoDB:** Page cleaner threads were not utilized optimally in sync-flush mode, which could cause page flush operations to slow down or stall in some cases. Sync-flush mode occurs when `InnoDB` is close to running out of free space in the redo log, causing the page cleaner coordinator to initiate aggressive page flushing. (Bug #31994031)

- **InnoDB:** A high frequency of updates while undo log truncation was enabled caused purge to lag. The lag was due to the `innodb_purge_rseg_truncate_frequency` setting being changed temporarily from 128 to 1 when an undo tablespace was selected for truncation. The code that modified the setting has been removed. (Bug #31991688)

- **InnoDB:** Automated truncation of undo tablespaces caused a performance regression. To address this issue, undo tablespace files are now initialized at 16MB and extended by a minimum of 16MB. To handle aggressive growth, the file extension size is doubled if the previous file extension happened less than 0.1 seconds earlier. Doubling of the extension size can occur multiple times to a maximum of 256MB. If the previous file extension occurred more than 0.1 seconds earlier, the extension size is reduced by half, which can also occur multiple times, to a minimum of 16MB. Previously, the initial size of an undo tablespace depended on the `InnoDB` page size, and undo tablespaces were extended four extents at a time.

  If the `AUTOEXTEND_SIZE` option is defined for an undo tablespace, the undo tablespace is extended by the greater of the `AUTOEXTEND_SIZE` setting and the extension size determined by the logic described above.

  When an undo tablespace is truncated, it is normally recreated at 16MB in size, but if the current file extension size is larger than 16MB, and the previous file extension happened within the last second, the new undo tablespace is created at a quarter of the size defined by the `innodb_max_undo_log_size` variable.

  Stale undo tablespace pages are no longer removed at the next checkpoint. Instead, the pages are removed in the background by the `InnoDB` master thread. (Bug #31965404, Bug #32020900, Bug #101194)

- **InnoDB:** A `posix_fallocate()` failure while preallocating space for a temporary tablespace raised an error and caused an initialization failure. A warning is now issued instead, and `InnoDB` falls back to the non-`posix_fallocate()` method for preallocating space. (Bug #31965379)

- **InnoDB:** An invalid pointer caused a shutdown failure on a MySQL Server compiled with the `DISABLE_PSI_MEMORY` source configuration option enabled. (Bug #31963333)

- **InnoDB:** A long SX lock held by an internal function that calculates new statistics for a given index caused a failure. (Bug #31889883)

- **InnoDB:** The `INFORMATION_SCHEMA.INNODB_TABLESPACES` table reported a `FILE_SIZE` of 0 for some tables and schemas. When the associated tablespace was not in the memory cache, the tablespace name was used to determine the tablespace file name, which was not always a reliable method. The tablespace ID is now used instead. Using the tablespace name remains as a fallback method. (Bug #31841617)

- **InnoDB:** After dropping a `FULLTEXT` index and renaming the table to move it to a new schema, the `FULLTEXT` auxiliary tables were not renamed accordingly and remained in the old schema directory. (Bug #31773368, Bug #100570)

- **InnoDB:** After upgrading to MySQL 8.0, a failure occurred when attempting to perform a DML operation on a table that was previously defined with a full-text search index. (Bug #31749490)

- **InnoDB:** Importing a tablespace with a page-compressed table did not report a schema mismatch error for source and destination tables defined with a different `COMPRESSION` setting. The `COMPRESSION` setting of the exported table is now saved to the `.cfg` metadata file during the `FLUSH TABLES ... FOR EXPORT` operation, and that information is checked on import to ensure that both tables are defined with the same `COMPRESSION` setting. (Bug #31744694)

- **InnoDB:** Dummy keys used to check if the MySQL Keyring plugin is functioning were left behind in an inactive state, and the number of inactive dummy keys increased over time. The actual master key is now used instead, if present. If no master key is available, a dummy master key is generated. (Bug #31737924)

- **InnoDB:** Querying the `INFORMATION_SCHEMA.FILES` table after moving the `InnoDB` system tablespace outside of the data directory raised a warning indicating that the `innodb_system` filename is unknown. (Bug #31603047)

- **InnoDB:** In a replication scenario involving a replica with binary logging or `log_slave_updates` disabled, the server failed to start due to an excessive number of gaps in the `mysql.gtid_executed` table. The gaps occurred for workloads that included both `InnoDB` and non-`InnoDB` transactions. GTIDs for `InnoDB` transactions are flushed to the `mysql.gtid_executed` table by the GTID persister thread, which runs periodically, while GTIDs for non-`InnoDB` transactions are written to the to the `mysql.gtid_executed` table directly by replica server threads. The GTID persister thread fell behind as it cycled through merging entries and compressing the `mysql.gtid_executed` table. As a result, the size of the GTID flush list for `InnoDB` transactions grew over time along with the number of gaps in the `mysql.gtid_executed` table, eventually causing a server failure and subsequent startup failures. To address this issue, the GTID persister thread now writes GTIDs for both `InnoDB` and non-`InnoDB` transactions, and foreground commits are forced to wait if the GTID persister thread falls behind. Also, the `gtid_executed_compression_period` default setting was changed from 1000 to 0 to disabled explicit compression of the `mysql.gtid_executed` table by default.

  Thanks to Venkatesh Prasad for the contribution. (Bug #31599938, Bug #100118)

- **InnoDB:** Persisting GTID values for XA transactions affected XA transaction performance. Two GTID values are generated for XA transactions, one for the prepare stage and another for the commit stage. The first GTID value is written to the undo log and later overwritten by the second GTID value. Writing

of the second GTID value could only occur after flushing the first GTID value to the `gtid_executed` table. Space is now reserved in the undo log for both XA transaction GTID values. (Bug #31467953, Bug #99638)

- **InnoDB:** `InnoDB` source files were updated to address warnings produced when building Doxygen source code documentation. (Bug #31354760)
- **InnoDB:** The full-text search synchronization thread attempted to read a previously-freed word from the index cache. (Bug #31310404)
- **InnoDB:** A 20µs sleep in the `buf_wait_for_read()` function introduced with parallel read functionality in MySQL 8.0.17 took 1ms on Windows, causing an unexpected timeout when running certain tests. Also, AIO threads were found to have uneven amounts of waiting operating system IO requests. (Bug #31095274)

  References: This issue is a regression of: Bug #31123564.
- **InnoDB:** Cleanup in certain replicated XA transactions failed to reattach transaction object (`trx_t`), which raised an assertion failure. (Bug #31006095)
- **InnoDB:** The tablespace encryption type setting was not properly updated due to a failure during the resumption of an `ALTER TABLESPACE ENCRYPTION` operation following a server failure. (Bug #30883833, Bug #98537)
- **InnoDB:** An interrupted tablespace encryption operation did not update the `encrypt_type` table option information in the data dictionary when the operation resumed processing after the server was restarted. (Bug #30883833, Bug #98537, Bug #30888919, Bug #98564)
- **InnoDB:** Internal counter variables associated with thread sleep delay and threads entering an leaving `InnoDB` were revised to use C++ `std::atomic`. Built-in atomic operations were removed. Thanks to Yibo Cai from ARM for the contribution. (Bug #30567060, Bug #97704)
- **InnoDB:** A relaxed memory order was implemented for dictionary memory variable fetch-add (`dict_temp_file_num.fetch_add`) and store (`dict_temp_file_num.store`) operations.

  Thanks to Yibo Cai for the contribution. (Bug #30567054, Bug #97703)
- **InnoDB:** A background thread that resumed a tablespace encryption operation after the server started failed to take an metadata lock on the tablespace, which permitted concurrent DDL operations and led to a race condition with the startup thread. The startup thread now waits until the tablespace metadata lock is taken. (Bug #28531637)
- **InnoDB:** Calls to `numa_all_nodes_ptr` were replaced by the `numa_get_mems_allowed()` function. Thanks to Daniel Black for the contribution. (Bug #24693086, Bug #83044)
- **Partitioning:** `ALTER TABLE t1 EXCHANGE PARTITION ... WITH TABLE t2` led to an assert when `t1` was not a partitioned table. (Bug #100971, Bug #31941543)

  References: This issue is a regression of: Bug #29706669.
- **Replication:** The `network_namespace` parameter for the `asynchronous_connection_failover_add_source()` and `asynchronous_connection_failover_delete_source()` function is no longer used. These functions add and remove replication source servers from the source list for a replication channel for the asynchronous connection failover mechanism. The network namespace for a replication channel is managed using the `CHANGE REPLICATION SOURCE` statement, and has special requirements for Group Replication source servers, so it should no longer be specified in the functions. (Bug #32078189)

- **Replication:** When `mysqlbinlog`'s `--print-table-metadata` option was used, `mysqlbinlog` used a different method for assessing numeric fields to the method used by the server when writing to the binary log, resulting in incorrect metadata output relating to these fields. `mysqlbinlog` now uses the same method as the server. (Bug #31956206)

- **Replication:** When using network namespaces in a replication channel and the initial connection from the replica to the master was interrupted, subsequent connection attempts failed to use the correct namespace information. (Bug #31954087)

- **Replication:** If a group member was expelled and made an auto-rejoin attempt at a point when some tables on the instance were locked (for example while a backup was running), the attempt failed and no further attempts were made. This scenario is now handled correctly. (Bug #31460690)

- **Replication:** As the number of replicas replicating from a semisynchronous source server increased, locking contention could result in a performance degradation. The locking mechanisms used by the plugins have been changed to use shared locks where possible, avoid unnecessary lock acquisitions, and limit callbacks. The new behaviors can be implemented by enabling the following system variables:

  - `replication_sender_observe_commit_only=1` limits callbacks.

  - `replication_optimize_for_static_plugin_config=1` adds shared locks and avoids unnecessary lock acquisitions. This system variable must be disabled if you want to uninstall the plugin.

  Both system variables can be enabled before or after installing the semisynchronous replication plugin, and can be enabled while replication is running. Semisynchronous replication source servers can also get performance benefits from enabling these system variables, because they use the same locking mechanisms as the replicas. (Bug #30519928)

- **Replication:** On a multi-threaded replica where the commit order is preserved, worker threads must wait for all transactions that occur earlier in the relay log to commit before committing their own transactions. If a deadlock occurs because a thread waiting to commit a transaction later in the commit order has locked rows needed by a transaction earlier in the commit order, a deadlock detection algorithm signals the waiting thread to roll back its transaction. Previously, if transaction retries were not available, the worker thread that rolled back its transaction would exit immediately without signalling other worker threads in the commit order, which could stall replication. A worker thread in this situation now waits for its turn to call the rollback function, which means it signals the other threads correctly. (Bug #26883680, Bug #87796)

- **Replication:** GTIDs are only available on a server instance up to the number of non-negative values for a signed 64-bit integer (2 to the power of 63 minus 1). If you set the value of `gtid_purged` to a number that approaches this limit, subsequent commits can cause the server to run out of GTIDs and take the action specified by `binlog_error_action`. From MySQL 8.0.23, a warning message is issued when the server instance is approaching the limit. (Bug #26035544)

- **Group Replication:** When the system variable `transaction_write_set_extraction=XXHASH64` is set, which is the default in MySQL 8.0 and a requirement for Group Replication, the collection of writes for a transaction previously had no upper size limit. Now, for standard source to replica replication, the numeric limit on write sets specified by `binlog_transaction_dependency_history_size` is applied, after which the write set information is discarded but the transaction continues to execute. Because the write set information is then unavailable for the dependency calculation, the transaction is marked as non-concurrent, and is processed sequentially on the replica. For Group Replication, the process of extracting the writes from a transaction is required for conflict detection and certification on all group members, so the write set information cannot be discarded if the transaction is to complete. The byte limit set by `group_replication_transaction_size_limit` is applied instead of the numeric limit, and if the limit is exceeded, the transaction fails to execute. (Bug #32019842)

- **Group Replication:** If the Group Replication applier channel (`group_replication_applier`) was holding a lock on a table, for example because of a backup in progress, and the member was expelled from the group and tried to rejoin automatically, the auto-rejoin attempt was unsuccessful and did not retry. Now, Group Replication checks during startup and rejoin attempts whether the `group_replication_applier` channel is already running. If that is the case at startup, an error message is returned. If that is the case during an auto-rejoin attempt, that attempt fails, but further attempts are made as specified by the `group_replication_autorejoin_tries` system variable. (Bug #31648211)

- **Microsoft Windows:** On Windows, running the MySQL server as a service caused shared-memory connections to fail. (Bug #32009251)

- **JSON:** `JSON_ARRAYAGG()` did not always perform proper error handling. (Bug #31856260, Bug #32012559, Bug #32181438)

- **JSON:** When updating a `JSON` value using `JSON_SET()`, `JSON_REPLACE()`, or `JSON_REMOVE()`, the target column can sometimes be updated in-place. This happened only when the target table of the update operation was a base table, but when the target table was an updatable view, the update was always performed by writing the full `JSON` value.

  Now in such cases, an in-place update (that is, a partial update) is also performed when the target table is an updatable view. (Bug #25840784)

- **JSON:** Work done in MySQL 8.0.22 to cause prepared statements to be prepared only once introduced a regression in the handling of dynamic parameters to JSON functions. All `JSON` arguments were classified as data type `MYSQL_TYPE_JSON`, which overlooked the fact that JSON functions take two kinds of JSON parameters—JSON values and JSON documents—and this distinction cannot be made with the data type only. For Bug #31667405, this problem was solved for comparison operators and the `IN()` operator by making it possible to tag a JSON argument as being a scalar value, while letting arguments to other JSON functions be treated as JSON documents.

  The present fix restores for a number of JSON functions their treatment of certain arguments as JSON values, as listed here:

  The first argument to `MEMBER OF()`

  The third, fifth, seventh, and subsequent odd-numbered arguments to the functions `JSON_INSERT()`, `JSON_REPLACE()`, `JSON_SET()`, `JSON_ARRAY_APPEND()`, and `JSON_ARRAY_INSERT()`. (Bug #101284, Bug #32063203)

  References: See also: Bug #31667405.

- **JSON:** When `mysqld` was run with `--debug`, attempting to execute a query that made use of a multi-valued index raised an error. (Bug #99833, Bug #31474182)

- Use of the `thread_pool` plugin could result in Address Sanitizer warnings. (Bug #32213294)

- While pushing a condition down to a materialized derived table, and a condition is partially pushed down, the optimizer may, in some cases in which a query transformation has added new conditions to the `WHERE` condition, call the internal `fix_fields()` function for the condition that remains in the outer query block. A successful return from this function call was misinterpreted as an error, leading to the silent failure of the original statement. (Bug #32150145)

- Multiple calls to a stored procedure containing an `ALTER TABLE` statement that included an `ORDER BY` clause could cause a server exit. (Bug #32147402)

- Prepared statements involving stored programs could cause heap-use-after-free memory problems. (Bug #32131022, Bug #32045681, Bug #32051928)
- Queries on `INFORMATION_SCHEMA` tables that involved materialized derived tables could fail. (Bug #32127562, Bug #101504)
- A potential buffer overflow was fixed. Thanks to Sifang Zhao for pointing out the issue, and for suggesting a fix (although it was not used). (Bug #32113015, Bug #101448)
- Conversion of `FLOAT` values to values of type `INT` could generate Undefined Behavior Sanitizer warnings. (Bug #32099994, Bug #32100033)
- In multiple-row queries, the `LOAD_FILE()` function evaluated to the same value for every row. (Bug #32096341, Bug #101401)
- Generic Linux `tar` file distributions had too-restrictive file permissions after unpacking, requiring a manual `chmod` to correct. (Bug #32080900)
- For debug builds, prepared `SET` statements containing subqueries in stored procedures could raise an assertion. (Bug #32078387)

  References: See also: Bug #32100210.
- For prepared statements, `illegal mix of collations` errors could occur for legal collation mixes. (Bug #32077842, Bug #101346, Bug #32145078, Bug #101575)
- The functions `REGEXP_LIKE()`, `REGEXP_INSTR()`, and `REGEXP_REPLACE()` raise errors for malformed regular expression patterns, but could also return `NULL` for such cases, causing subsequent debug asserts. Now we ensure that these functions do not return `NULL` except in certain specified cases.

  The function `REGEXP_SUBSTR()` can always return `NULL`, so no such check is needed, and for this function we make sure that one is not performed. (Bug #32053093)
- Testing an aggregate function for `IS NULL` or `IS NOT NULL` in a `HAVING` condition using `WITH ROLLUP` led to wrong results. (Bug #32049313)
- When a new aggregate function was added to the current query block because an inner query block had an aggregate function requiring evaluation in the current one, the server did not add rollup wrappers to it as needed. (Bug #32034914)
- For debug builds, certain `CREATE TABLE` statements with `CHECK` constraints could raise an assertion. (Bug #32018406, Bug #101180)
- Incorrect BLOB field values were passed from `InnoDB` during a secondary engine load operation. (Bug #32014483)
- The LOCK_ORDER tool did not correctly represent `InnoDB` share exclusive locks. (Bug #31994052)
- The server did not handle properly an error raised when trying to use an aggregation function with an invalid column type as part of a hash join. (Bug #31989333)
- The length of the `WORD` column of the `INFORMATION_SCHEMA.KEYWORDS` table could change depending on table contents. (Bug #31982157)
- The Performance Schema `host_cache` table was empty and did not expose the contents of the host cache if the Performance Schema was disabled. The table now shows cache contents regardless of whether the Performance Schema is enabled. (Bug #31978763)
- A `HANDLER READ` statement sometimes hit an assert when a previous statement did not restore the original value of `THD::mark_used_columns` after use. (Bug #31977414)

- Importing a compressed table could cause an unexpected server exit if the table contained values that were very large when uncompressed. (Bug #31943021)
- Removed a memory leak that could occur when a subquery using a hash join and `LIMIT` was executed repeatedly. (Bug #31940549)
- A compilation failure on Ubuntu was corrected. (Bug #31930934, Bug #100938)
- Memory used for storing partial-revokes information could grow excessively for sessions that executed a large number of statements. (Bug #31919448)
- The server did not handle all cases of the `WHERE_CONDITION` optimization correctly. (Bug #31905199)
- `FLUSH TABLES WITH READ LOCK` could block other sessions from executing `SHOW TABLE STATUS`. (Bug #31894662)
- In some cases, `MIN()` and `MAX()` incorrectly returned `NULL` when used as window functions with temporal or `JSON` values as arguments. (Bug #31882291)
- `GRANT ... GRANT OPTION ... TO` and `GRANT ... TO .. WITH GRANT OPTION` sometimes were not correctly written to the server logs. (Bug #31869146, Bug #100793)
- For debug builds, `CREATE TABLE` using a partition list of more than 256 entries raised an assertion. (Bug #31867653)
- It was possible for queries in the file named by the `init_file` system variable to cause server startup failure. (Bug #31835782)
- When performing a hash join, the optimizer could register a false match between a negative integer value and a very large unsigned integer value. (Bug #31832001, Bug #31940639, Bug #100967)
- `SHOW VARIABLES` could report an incorrect value for the `partial_revokes` system variable. (Bug #31819558, Bug #100677)
- In the Performance Schema `user_defined_functions` table, the value of the `UDF_LIBRARY` column is supposed to be `NULL` for loadable functions registered via the service API. The value was incorrectly set to the empty string. (Bug #31791754)
- The server automatic upgrade procedure failed to upgrade older help tables that used the `latin1` character set. (Bug #31789964)
- Duplicate warnings could occur when executing an SQL statement that read the grant tables in serializable or repeatable-read transaction isolation level. (Bug #31769242)
- In certain queries with `DISTINCT` aggregates (which in general are solved by sorting before aggregation), the server used a temporary table instead of streaming due to the mistaken assumption that the logic for handling the temporary table performed deduplication. Now the server checks for the implied unique index instead, which is more robust and allows for the removal of unnecessary logic. (Bug #31762806)
- Certain combinations of `lower_case_table_names` values and schema names in Event Scheduler event definitions could cause the server to stall. (Bug #31733090)
- Calling one stored function from within another could produce a conflict in field resolution, resulting in a server exit. (Bug #31731334)
- Loadable functions defined without a `udf_init()` method could cause an unexpected server exit. (Bug #31701219)

- Setting the `secure_file_priv` system variable to `NULL` should disable its action, but instead caused the server to create a directory named `NULL`. (Bug #31700734, Bug #100384)

- `mysqlpump` could exit unexpectedly due to improper simultaneous accesses to shared structures. (Bug #31696241)

- Uninstalling a component and deregistering loadable functions installed by the component was not properly synchronized with whether the functions were currently in use. (Bug #31646698)

- Cleanup following execution of a prepared statement that performed a multi-table `UPDATE` or `DELETE` was not always done correctly, which meant that, following the first execution of such a prepared statement, the server reported a nonzero number of rows updated, even though no rows were actually changed. (Bug #31640267)

  References: See also: Bug #32100210.

- For the engines which support primary key extension, when the total key length exceeded `MAX_KEY_LENGTH` or the number of key parts exceeded `MAX_REF_PARTS`, key parts of primary keys which did not fit within these limits were not added to the secondary key, but key parts of primary keys were unconditionally marked as part of secondary keys.

  This led to a situation in which the secondary key was treated as a covering index, which meant sometimes the wrong access method was chosen.

  This is fixed by modifying the way in which key parts of primary keys are added to secondary keys so that those which do not fit within which do not fit within the limits mentioned previously mentioned are cleared. (Bug #31617858)

- When MySQL is configured with `-DWITH_ICU=system`, `CMake` now checks that the ICU library version is sufficiently recent. (Bug #31600044)

- When invoked with the `--binary-as-hex` option, `mysql` displayed `NULL` values as empty binary strings (`0x`).

  Selecting an undefined variable returned the empty binary string (`0x`) rather than `NULL`. (Bug #31549724, Bug #31638968, Bug #100251)

- Enabling `DISABLE_PSI_`*`xxx`* Performance Schema-related `CMake` options caused build failures. (Bug #31549724)

- Some queries returned different results depending on the value of `internal_tmp_mem_storage_engine`.

  The root cause of this issue related to the fact that, when buffering rows for window functions, if the size of the in-memory temporary table holding these buffered rows exceeds the limit specified, a new temporary table is created on disk; the frame buffer partition offset is set at the beginning of a new partition to the total number of rows that have been read so far, and is updated specifically for use when the temporary table is moved to disk (this being used to calculate the hints required to process window functions). The problem arose because the frame buffer partition offset was not updated for the specific case when a new partition started while creating the temporary table on disk, which caused the wrong rows to be read.

  This issue is fixed by making sure to update the frame buffer partition offset correctly whenever a new partition starts while a temporary table is moved to disk. (Bug #31546816)

- While buffering rows for window functions, if the size of the in-memory temporary table holding these buffered rows exceeds the limit specified by `temptable_max_ram`, a new temporary table is created on

disk. After the creation of the temporary table, hints used to process window functions need to be reset, since the temporary table is now moved to disk, making the existing hints unusable. When the creation of the temporary table on disk occurred when the first row in the frame buffer was being processed, the hints had not been initialized and trying to reset these uninitialized hints resulted in an unplanned server exit.

This issue is fixed by adding a check to verify whether frame buffer hints have been initialized, prior to resetting them. (Bug #31544404)

- The Performance Schema could produce incorrect results for joins on a `CHANNEL_NAME` column when the index for `CHANNEL_NAME` was disabled with `USE INDEX ()`. (Bug #31544023, Bug #99989)
- When removing unused window definitions, a subquery that was part of an `ORDER BY` was not removed. (Bug #31518806)
- In certain cases, the server did not handle multiply-nested subqueries correctly. (Bug #31472704)
- The recognized syntax for a `VALUES` statement includes an `ORDER BY` clause, but this clause was not resolved, so the execution engine could encounter invalid data. (Bug #31387510)
- The server attempted to access a non-existent temporary directory at startup, causing a failure. Checks were added to ensure that temporary directories exist, and that files are successfully created in the `tmpdir` directory. (Bug #31377118)
- While removing redundant sorting, a window's ordering was removed due to the fact that rows were expected to come in order because of the ordering of another window. When the other window was subsequently removed because it was unused, this resulted in unordered rows, which was not expected during evaluation.

  Now in such cases, removal of redundant sorts is not performed until after any unused windows have been removed. In addition, resolution of any rollups has been moved to the preparation phase. (Bug #31361393)
- Semisynchronous replication errors were incorrectly written to the error log with a subsystem tag of `Server`. They are now written with a tag of `Repl`, the same as for other replication errors. (Bug #31327337)
- A user could grant itself as a role to itself. (Bug #31222230)
- The server did not always correctly handle cases in which multiple `WHERE` conditions, one of which was always FALSE, referred to the same subquery. (Bug #31216115)
- With a `lower_case_table_names=2` setting, `InnoDB` background threads sometimes acquired table metadata locks using the wrong character case for the schema name part of a lock key, resulting in unprotected metadata and race conditions. The correct character case is now applied. Changes were also implemented to prevent metadata locks from being released before corresponding data dictionary objects, and to improve assertion code that checks lock protection when acquiring data dictionary objects. (Bug #31165802)
- If a `CR_UNKNOWN_ERROR` was to be sent to a client, an exception occurred. (Bug #31123643)
- Conversion of `DOUBLE` values to values of type `BIT`, `ENUM`, or `SET` could generate Undefined Behavior Sanitizer warnings. (Bug #31019130)
- Certain accounts could cause server startup failure if the `skip_name_resolve` system variable was enabled. (Bug #31018510)
- Client programs could unexpectedly exit if communication packets contained bad data. (Bug #30890850)

- A buffer overflow in the client library was fixed. (Bug #30885987)
- When creating a multi-valued or other functional index, a performance drop was seen when executing a query against the table on which the index was defined, even though the index itself was not actually used. This occurred because the hidden virtual column that backs such indexes was evaluated unnecessarily for each row in the query. (Bug #30838749)

  References: This issue is a regression of: Bug #28069731.
- `CMake` checks for `libcurl` dependencies were improved. (Bug #30268245)
- `mysql_config_editor` incorrectly treated `#` in password values as a comment character. (Bug #29861961, Bug #95597)
- In some cases, the optimizer attempted to compute the hash value for an empty string. Now a fixed value is always used instead. (Bug #22588319)
- The `INSERT()` and `RPAD()` functions did not correctly set the character set of the result. (Bug #22523946, Bug #79909, Bug #31887870, Bug #100841)
- Some corner cases for *`val1`* `BETWEEEN` *`val2`* `AND` *`val3`* were fixed, such as that `-1 BETWEEN 9223372036854775808 AND 1` returned true. (Bug #22515857, Bug #79878)
- For the Performance Schema `memory_summary_global_by_event_name` table, the low watermark columns could have negative values, and the high watermark columns had ever-increasing values even when the server memory usage did not increase. (Bug #22246001, Bug #79285)
- Several issues converting strings to numbers were fixed. (Bug #19186271, Bug #73248)
- Certain group by queries that performed correctly did not return the expected result when `WITH ROLLUP` was added. This was due to the fact that decimal information was not always correctly piped through rollup group items, causing functions returning decimal values such as `TRUNCATE()` to receive data of the wrong type. (Bug #101684, Bug #32179240)
- When creating fields for materializing temporary tables (that is, when needing to sort a join), the optimizer checks whether the item needs to be copied or is only a constant. This was not done correctly in one specific case; when performing an outer join against a view or derived table containing a constant, the item was not properly materialized into the table, which could yield spurious occurrences of `NULL` in the result. (Bug #101622, Bug #32162862)

  References: See also: Bug #31790217.
- When `REGEXP_REPLACE()` was used in an SQL statement, the internal function `Regexp_engine::Replace()` did not reset the error code value after handling a record, which could affect processing of the next record, which lead to issues.

  Our thanks to Hope Lee for the contribution. (Bug #101256, Bug #32050219)
- For a query having the following form, the column list sometimes assumed an inconsistent state after temporary tables were created, causing out-of-bounds indexing later:

```
SELECT * FROM (
    SELECT PI()
    FROM t1 AS table1, t1 AS table2
    ORDER BY PI(), table1.a
) AS d1;
```

  (Bug #101012, Bug #31955761, Bug #31978439)

  References: This issue is a regression of: Bug #31790217.

- When aggregating data that was already sorted (known as performing streaming aggregation, due to no temporary tables being used), it was not possible to determine when a group ended until processing the first row in the next group, by which time the group expressions to be output were often already overwritten.

  This is fixed by replacing the complex logic previously used with the much simpler method of saving a representative row for the group when encountering it the first time, so that its columns can easily be retrieved for the output row when needed. (Bug #100791, Bug #27272052, Bug #31073167, Bug #31790217, Bug #31868610)

- Subqueries making use of fulltext matching might not perform properly when `subquery_to_derived` was enabled, and could lead to an assert in debug builds. (Bug #100749, Bug #31851600)

- When an `ALTER TABLE ... CONVERT TO CHARACTER SET` statement is executed, the character set of every `CHAR`, `VARCHAR`, and `TEXT` column in the table is updated to the new `CHARACTER SET` value. This change was also applied to the hidden `CHAR` column used by an `ARRAY` column for a multi-valued index; since the character set of the hidden column must be one of `my_charset_utf8mb4_0900_bin` or `binary`, this led to an assert in debug builds of the server.

  This issue is resolved by no longer setting the character set of the hidden column to that of the table when executing the `ALTER TABLE` statement referenced previously; this is similar to what is done for `BLOB` columns in similar circumstances. (Bug #99403, Bug #31301101)

- In some cases, the server's internal string-conversion routines had problems handling floating-point values which used length specifiers and triggered use of scientific notation. (Bug #92537, Bug #101570, Bug #28691605, Bug #32144265)

  References: See also: Bug #88256, Bug #27041543.

# Changes in MySQL 8.0.22 (2020-10-19)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

- Account Management Notes
- Audit Log Notes
- C API Notes
- Compilation Notes
- Configuration Notes
- Deprecation and Removal Notes
- Keyring Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Pluggable Authentication
- SQL Syntax Notes
- X Plugin Notes

- Functionality Added or Changed
- Bugs Fixed

## Account Management Notes

- Lock handling for statements involving the grant tables was improved. (Bug #31291237, Bug #31576185, WL #14084)
- Modifying the `mysql.infoschema` and `mysql.sys` reserved accounts now requires the `SYSTEM_USER` privilege. (Bug #31255458)
- For the `CREATE USER`, `DROP USER`, and `RENAME USER` account-management statements, the server now performs additional security checks designed to prevent operations that (perhaps inadvertently) cause stored objects to become orphaned or that cause adoption of stored objects that are currently orphaned. Such operations now fail with an error. If you have the `SET_USER_ID` privilege, it overrides the checks and those operations produce a warning rather than an error; this enables administrators to perform the operations when they are deliberately intended. See Orphan Stored Objects. (WL #14073)

## Audit Log Notes

- For JSON-format log files, MySQL Enterprise Audit supports log-reading operations using the `audit_log_read()` user-defined function. Previously, specifying the position at which to begin reading was possible only by passing to `audit_log_read()` an argument containing a bookmark indicating the exact timestamp and event ID of a particular event. For greater flexibility, the argument now can be a start specifier that names any timestamp, to read starting from the first event that occurs on or after that timestamp. See Reading Audit Log Files. (WL #13936)

## C API Notes

- The MySQL client library now includes a `mysql_real_connect_dns_srv()` C API function that is similar to `mysql_real_connect()` but uses a DNS SRV record to determine the candidate hosts for establishing a connection to a MySQL server, rather than explicit host, port, and socket arguments.

  Applications that use the C API can call the new function directly. In addition, the `mysql` client program is modified to use DNS SRV capability; it now supports a `--dns-srv-name` option that takes precedence over `--host` and causes the connection to be based on a DNS SRV record. See mysql_real_connect_dns_srv().

  Connection establishment in other contexts is unaffected, including connections made by replicas, the `FEDERATED` storage engine, and client programs other than `mysql`. (WL #13905)

## Compilation Notes

- On Windows, Visual Studio 2019 Update 4 is now the minimum version for MySQL compilation. (Bug #31655401)
- The minimum version of the Boost library for server builds is now 1.73.0. (Bug #31309800)

## Configuration Notes

- The new `WITH_TCMALLOC` `CMake` option indicates whether to link with `-ltcmalloc`. If enabled, built-in `malloc()`, `calloc()`, `realloc()`, and `free()` routines are disabled. The default is `OFF`. `WITH_TCMALLOC` and `WITH_JEMALLOC` are mutually exclusive. (Bug #31785166)
- The new `COMPRESS_DEBUG_SECTIONS` `CMake` option indicates whether to compress the debug sections of binary executables (Linux only). Compressing executable debug sections saves space at

the cost of extra CPU time during the build process. The default is `OFF`. If this option is not set explicitly but the `COMPRESS_DEBUG_SECTIONS` environment variable is set, the option takes its value from that variable. (Bug #31498296)

- The `WITH_DEFAULT_FEATURE_SET` `CMake` option was removed. (Bug #31122507)
- On platforms that implement network namespace support (such as Linux), MySQL now enables configuring the network namespace for TCP/IP connections from client programs to the MySQL server or X Plugin:
  - On the server side, the `bind_address`, `admin_address`, and `mysqlx_bind_address` system variables have extended syntax for specifying the network namespace to use for a given IP address or host name on which to listen for incoming connections.
  - For client connections, the `mysql` client and the `mysqlxtest` test suite client support a `--network-namespace` option for specifying the network namespace.
  - For replication connections from replica servers to source servers, the `CHANGE MASTER TO` statement supports a `NETWORK_NAMESPACE` option for specifying the network namespace.

  For replication monitoring purposes, the Performance Schema `replication_connection_configuration` table, the replica server connection metadata repository (see Replication Metadata Repositories), and the `SHOW REPLICA | SLAVE STATUS` statement have a new column that displays the applicable network namespace for connections.

  For more information, including the host system prerequisites that must be satisfied to use this feature, see Network Namespace Support. (WL #12720)

## Deprecation and Removal Notes

- From MySQL 8.0.22, the `group_replication_ip_whitelist` system variable is deprecated, and the system variable `group_replication_ip_allowlist` has been added to replace it. The system variable works in the same way as before, only the terminology has changed.

  For both system variables, the default value is `AUTOMATIC`. If either one of the system variables has been set to a user-defined value and the other has not, the changed value is used. If both of the system variables have been set to a user-defined value, the value of `group_replication_ip_allowlist` is used. (WL #14175)
- From MySQL 8.0.22, the statements `START SLAVE`, `STOP SLAVE`, `SHOW SLAVE STATUS`, `SHOW SLAVE HOSTS` and `RESET SLAVE` are deprecated. The following aliases should be used instead:
  - Instead of `START SLAVE` use `START REPLICA`
  - Instead of `STOP SLAVE` use `STOP REPLICA`
  - Instead of `SHOW SLAVE STATUS` use `SHOW REPLICA STATUS`
  - Instead of `SHOW SLAVE HOSTS` use `SHOW REPLICAS`
  - Instead of `RESET SLAVE` use `RESET REPLICA`

  The statements work in the same way as before, only the terminology used for each statement and its output has changed.

  New status variables have been added as aliases for the related status variables. Both the old and new versions of the statements update both the old and new versions of these status variables:

- `Com_slave_start` is equivalent to `Com_replica_start`
- `Com_slave_stop` is equivalent to `Com_replica_stop`
- `Com_show_slave_status` is equivalent to `Com_show_replica_status`
- `Com_show_slave_hosts` is equivalent to `Com_show_replicas`

(WL #14171)

- The InnoDB memcached plugin is deprecated and support for it will be removed in a future MySQL version. (WL #14131)
- The `INFORMATION_SCHEMA.TABLESPACES` table is unused. It is now deprecated and will be removed in a future MySQL version. Other `INFORMATION_SCHEMA` tables may provide related information, as described in The INFORMATION_SCHEMA TABLESPACES Table. (WL #14064)

## Keyring Notes

- MySQL Enterprise Edition now includes a `keyring_oci` plugin that uses Oracle Cloud Infrastructure Vault as a back end for keyring storage. No key information is permanently stored in MySQL server local storage. All keys are stored in Oracle Cloud Infrastructure Vault, making this plugin well suited for Oracle Cloud Infrastructure MySQL customers for management of their MySQL Enterprise Edition keys. For more information, see The MySQL Keyring. (WL #9770)

## Optimizer Notes

- **Important Change:** A prepared statement is now prepared only once, when executing `PREPARE`, rather than once each time it is executed. In addition, a statement inside a stored procedure is also now prepared only once, when the stored procedure is first executed. This change enhances performance of such statements, since it avoids the added cost of repeated preparation and rollback of preparation structures, the latter being the source of several bugs.

  As part of this work, the manner in which dynamic parameters used in prepared statements are resolved is changed, with the resulting changes in prepared statement use cases listed here:

  - A parameter used in a prepared statement has its data type determined when the statement is prepared, and the type persists for each subsequent execution of the statement, unless the statement is reprepared (see PREPARE Statement, for information about when this may occur).
  - For a prepared statement of the form `SELECT expr1, expr2, ... FROM table ORDER BY ?`, passing an integer value *N* for the parameter no longer causes ordering of the results by the $N^{th}$ expression in the select list; the results are no longer ordered, as is expected with `ORDER BY constant`.
  - It was possible (although unsupported) to insert a floating point value into a `LIMIT` or `OFFSET` clause by means of a parameter (and possibly a user variable). This is now explicitly disallowed; such values must be integers.
  - The window functions `NTILE(NULL)`, `NTH_VALUE(expr, NULL)`, `LEAD(expr, nn)`, and `LAG(expr, nn)`, where *nn* is a negative number, are now disallowed, to comply with the SQL standard.
  - A user variable that is read by a prepared statement now has its type determined when the statement is prepared; the type persists for each subsequent execution of the statement.

- A user variable that is read by a statement within a stored procedure now has its type determined the first time the statement is executed; the type persists for all subsequent invocations of the containing stored procedure.
- For parameters for which no contextual information is available to determine the parameter type, the server assumes the parameter is a character string with the default character set, not a binary string. Parameters for which this is incorrect may be placed within a `CAST()` expression.

See PREPARE Statement, for the rules governing how the effectiue data types of parameters and user variables used within prepared statements are determined.

In addition, the rows (*`N`*) argument to the window functions `LAG()`, `LEAD()`, and `NTILE()` must now be an integer in the range `0` to $2^{63}$, inclusive, in any of the following forms:

- an unsigned integer constant literal
- a positional parameter marker (`?`)
- a user-defined variable
- a local variable in a stored routine

In addition, this argument is no longer permitted to be `NULL`. See the descriptions of the functions just referenced for more information. (Bug #48612, Bug #99601, Bug #100150, Bug #105166, Bug #11756670, Bug #23599127, Bug #31119132, Bug #31365678, Bug #31393719, Bug #31592822, Bug #31810577, Bug #33448735, WL #9384)

- The `filesort` algorithm now supports sorting a join on multiple tables, and not just a single table. (Bug #31310238, Bug #31559978, Bug #31563876)
- When using a `RIGHT JOIN`, some internal objects, were not converted to those suitable for use with a `LEFT JOIN` as intended. These included some lists of tables built at parse time, but which did not have their order reversed. This required maintaining code to handle instances in which a `LEFT JOIN` was originally a `RIGHT JOIN` as special cases, and was the source of several bugs. Now the server performs any necessary reversals at parse time, so that after parsing, a `RIGHT JOIN` is in fact, in all respects, a `LEFT JOIN`. (Bug #30887665, Bug #30964002, WL #6509)

  References: See also: Bug #12567331, Bug #21350125.

- Added support for periodic synchronization when writing to files with `SELECT INTO DUMPFILE` and `SELECT INTO OUTFILE` statements. This feature can be enabled by setting the `select_into_disk_sync` system variable to `ON`; the size of the write buffer cn be set using the server system variable `select_into_buffer_size`; the default buffer size is 131072 ($2^{17}$) bytes. An optional delay following synchronization to disk can also be set using the `select_into_disk_sync_delay` system variable; the default behaviour is not to allow any delay (that is, a delay time of 0 milliseconds).

  For more information, see the descriptions of the system variables referenced previously.

  Our thanks to Facebook for this contribution to MySQL 8.0. (Bug #30284861, WL #13926)

- MySQL now implements derived condition pushdown for eligible queries. What this means is that, for a query such as `SELECT * FROM (SELECT i, j FROM t1) AS dt WHERE i > constant`, it is now possible in many cases to push the outer `WHERE` condition down to the derived table, in this case resulting in `SELECT * FROM (SELECT i, j FROM t1 WHERE i > constant) AS dt`. Previously, if the derived table was materialized and not merged, MySQL materialized the entire table—in this case `t1`—then qualified the rows with the `WHERE` condition.

When the derived table cannot be merged into the outer query (for example, if the derived table uses aggregation), pushing the outer `WHERE` condition down to the derived table can reduce the number of rows that need to be processed, which should improve the query's performance.

An outer `WHERE` condition can be pushed down directly to a materialized derived table when the derived table uses no aggregate or window functions. In addition, when the derived table has a `GROUP BY` and uses no window functions, the outer `WHERE` condition can be pushed down to the derived table as a `HAVING` condition. If the derived table uses a window function and the outer `WHERE` references columns used in the window function's `PARTITION` clause, the `WHERE` condition can also be pushed down.

This optimization cannot be employed for a derived table that contains a `UNION` or `LIMIT` clause.

To enable derived condition pushdown, the `optimizer_switch` system variable's `derived_condition_pushdown` flag (added in this release) must be set to `on`. This is the default setting. If this optimization is disabled by the optimizer switch setting, you can enable it for a specific query using the `DERIVED_CONDITION_PUSHDOWN` optimizer hint (also added in this release). Use the `NO_DERIVED_CONDITION_PUSHDOWN` optimizer hint to disable the optimization for a given query.

For further information and examples, see Derived Condition Pushdown Optimization. (Bug #59870, Bug #88381, Bug #11766303, Bug #27590273, WL #8084)

## Packaging Notes

- For RPM and Debian packages, client-side plugins were moved to their own client-plugins package. (Bug #31584093)
- The `VERSION` file in MySQL source distributions is now named `MYSQL_VERSION` due to a naming conflict with Boost. (Bug #31466846)
- For platforms on which MySQL is built with `WITH_SYSTEMD=1`, packages no longer include legacy System V files: the `mysqld_multi.server` and `mysql.server` scripts, and the `mysql.server.1`, `mysqld_multi.1`, and `mysqld_safe.1` man pages. (Bug #31450888)

## Performance Schema Notes

- The `SHOW PROCESSLIST` statement provides process information by collecting thread data from all active threads. However, because the implementation iterates across active threads from within the thread manager while holding a global mutex, it has negative performance consequences, particularly on busy systems.

  An alternative `SHOW PROCESSLIST` implementation is now available based on the new Performance Schema `processlist` table. This implementation queries active thread data from the Performance Schema rather than the thread manager and does not require a mutex:

  - To enable the alternative implementation, enable the `performance_schema_show_processlist` system variable.
  - The alternative implementation of `SHOW PROCESSLIST` also applies to the `mysqladmin processlist` command.
  - The alternative implementation does not apply to the `INFORMATION_SCHEMA PROCESSLIST` table or the `COM_PROCESS_INFO` command of the MySQL client/server protocol.
  - To ensure that the default and alternative implementations yield the same information, certain configuration requirements must be met; see The processlist Table.

(WL #9090)

- An SQL interface to the most recent events written to the MySQL server error log is now available by means of queries on the new Performance Schema `error_log` table. This table has a fixed size, with old events automatically discarded as necessary to make room for new ones. The table is populated if error log configuration includes a log sink component that supports this capability (currently the traditional-format `log_sink_internal` and JSON-format `log_sink_json` sinks). Several new status variables provide information about `error_log` table operation. See The error_log Table. (WL #13681)

## Pluggable Authentication

- These changes were made for the LDAP authentication plugins:
  - For the SASL LDAP authentication plugin, the `SCRAM-SHA-1` authentication method is not supported on On SLES 12 and 15 and EL6 systems. The default method on those systems is now `GSSAPI`.
  - If the LDAP host is not set, the LDAP connection pool will not be initialized, which enables successful authentication plugin installation in cases when previously it would fail. (This might be the case when a site installs a plugin first, then configures it later.)
  - If an LDAP connection parameter is changed at runtime, the LDAP connection pool is reinitialized for the first subsequent authentication attempt.
  - If the LDAP server is restarted, existing connections in the connection pool become invalid. The LDAP authentication plugin detects this case and reinitializes the connection pool and (for the SASL LDAP plugin) the SASL challenge is resent.

  (Bug #31664270, Bug #31219323)

## SQL Syntax Notes

- The parser now supports parenthesized query expressions using this syntax:

```
( query_expression )
  [order_by_clause]
  [limit_clause]
  [into_clause]
```

  Other variations are possible; see Parenthesized Query Expressions (Bug #30592703)

- It is now possible to cast values of other types to `YEAR`, using either the `CAST()` function or the `CONVERT()` function. These functions now support `YEAR` values of one or two digits in the range 0-99, and four-digit values in the range 1901-2155. Integer 0 is converted to Year 0; a string consisting of one or more zeroes (following possible truncation) is converted to the year 2000. Casting adds 2000 to values in the range 1-69 inclusive, and 1900 to values in the range 70-99 inclusive.

  Strings beginning with one, two, or four digits followed by at least one non-digit character (and possibly other digit or non-digit characters) are truncated prior to conversion to `YEAR`; in such cases, the server emits a truncation warning. Floating-point values are rounded prior to conversion; `CAST(1944.5 AS YEAR)` returns 1945 due to rounding, and `CAST("1944.5" AS YEAR)` returns 1944 (with a warning) due to truncation.

  `DATE`, `DATETIME`, and `TIMESTAMP` are cast to the `YEAR` portion of the value. A `TIME` value is cast to the current year. Not specifying the value to be cast as a `TIME` value may yield a different result from what is expected; `CAST("13:47" AS YEAR)` returns 2013 due to truncation of the string value, and `CAST(TIME "13:47" AS YEAR)` returns 2020 as of the year of this release.

Casting of GEOMETRY values to YEAR is not supported. A cast of an incompatible type or an out-of-range or illegal value returns NULL.

YEAR can also be used as the return type for the JSON_VALUE() function. This function supports four-digit years only, and otherwise follows the same rules as apply to CAST() and CONVERT() when performing casts to YEAR.

For more information, see the description of the CONVERT() function. (WL #14015)

- When selecting a TIMESTAMP column value, it is now possible to convert it from the system time zone to a UTC DATETIME when retrieving it, using the AT TIME ZONE operator which is implemented for the CAST() function in this release.

  The syntax is CAST(*value* AT TIME ZONE *specifier* AS DATETIME[(*precision*)]), where the *value* is a TIMESTAMP, and the *specifier* is one of [INTERVAL] '+00:00' or 'UTC'. (INTERVAL is optional with the first form of the specifier, and cannot be used with 'UTC'.) The *precision* of the DATETIME value returned by the cast can optionally be specified up to 6 decimal places.

  Values that were inserted into the table using a timezone offset are also supported.

  AT TIME ZONE cannot be used with CONVERT(), or in any other context other than as part of a CAST() function call. The ARRAY keyword and creation of multi-valued indexes are also not supported when using AT TIME ZONE.

  A brief example is shown here:

```
mysql> SELECT @@system_time_zone;
+--------------------+
| @@system_time_zone |
+--------------------+
| EDT                |
+--------------------+
1 row in set (0.00 sec)

mysql> CREATE TABLE ex (ts TIMESTAMP);
Query OK, 0 rows affected (0.81 sec)

mysql> INSERT INTO ex VALUES
    >     ROW(CURRENT_TIMESTAMP),
    >     ROW('2020-07-31 21:44:30-08:00');
Query OK, 2 rows affected (0.09 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> TABLE ex;
+---------------------+
| ts                  |
+---------------------+
| 2020-07-28 21:39:31 |
| 2020-08-01 01:44:30 |
+---------------------+
2 rows in set (0.00 sec)

mysql> SELECT ts, CAST(ts AT TIME ZONE 'UTC' AS DATETIME) AS ut FROM ex;
+---------------------+---------------------+
| ts                  | ut                  |
+---------------------+---------------------+
| 2020-07-28 21:39:31 | 2020-07-29 01:39:31 |
| 2020-08-01 01:44:30 | 2020-08-01 05:44:30 |
+---------------------+---------------------+
```

```
2 rows in set (0.00 sec)
```

For more information and examples, see the description of the `CAST()` function in the *MySQL Manual*. (WL #12535)

## X Plugin Notes

- In specific conditions, terminating an X Protocol connection could cause MySQL Server to stop unexpectedly. (Bug #31671503)

## Functionality Added or Changed

- `LOCK TABLES` privilege checking for views was improved. (Bug #31304432, WL #14092)
- You can use MySQL Server's new asynchronous connection failover mechanism to automatically establish an asynchronous (source to replica) replication connection to a new source after the existing connection from a replica to its source fails. The connection fails over if the replication I/O thread stops due to the source stopping or due to a network failure. The asynchronous connection failover mechanism can be used to keep a replica synchronized with multiple MySQL servers or groups of servers that share data. To activate asynchronous connection failover for a replication channel set `SOURCE_CONNECTION_AUTO_FAILOVER=1` on the `CHANGE MASTER TO` statement for the channel, and set up a source list for the channel using the `asynchronous_connection_failover_add_source` and `asynchronous_connection_failover_delete_source` functions. (WL #12649)
- The new `innodb_extend_and_initialize` variable permits configuring how `InnoDB` allocates space to file-per-table and general tablespaces on Linux. By default, when an operation requires additional space in a tablespace, `InnoDB` allocates pages to the tablespace and physically writes NULLs to those pages. This behavior affects performance if new pages are allocated frequently. As of MySQL 8.0.22, you can disable `innodb_extend_and_initialize` on Linux systems to avoid physically writing NULLs to newly allocated tablespace pages. When `innodb_extend_and_initialize` is disabled, space is allocated using `posix_fallocate()` calls, which reserve space without physically writing NULLs.

  A `posix_fallocate()` operation is not atomic, which makes it possible for a failure to occur between allocating space to a tablespace file and updating the file metadata. Such a failure can leave newly allocated pages in an uninitialized state, resulting in a failure when `InnoDB` attempts to access those pages. To prevent this scenario, `InnoDB` writes a redo log record before allocating a new tablespace page. If a page allocation operation is interrupted, the operation is replayed from the redo log record during recovery. (WL #13782)
- To permit concurrent DML and DDL operations on MySQL grant tables, read operations that previously acquired row locks on MySQL grant tables are now executed as non-locking reads. The operations that are now performed as non-locking reads on MySQL grant tables include:
  - `SELECT` statements and other read-only statements that read data from grant tables through join lists and subqueries, including `SELECT ... FOR SHARE` statements, using any transaction isolation level.
  - DML operations that read data from grant tables (through join lists or subqueries) but do not modify them, using any transaction isolation level.

  Statements that no longer acquire row locks when reading data from grant tables report a warning if executed while using statement-based replication.

  When using -`binlog_format=mixed`, DML operations that read data from grant tables are now written to the binary log as row events to make the operations safe for mixed-mode replication.

`SELECT ... FOR SHARE` statements that read data from grant tables now report a warning. With the `FOR SHARE` clause, read locks are not supported on grant tables.

DML operations that read data from grant tables and are executed using the `SERIALIZABLE` isolation level now report a warning. Read locks that would normally be acquired when using the `SERIALIZABLE` isolation level are not supported on grant tables. (WL #14087)

- The `ALTER DATABASE` statement now supports a `READ ONLY` option that controls whether to permit modification of a database and objects within it. This option is useful for database migration because a database for which `READ ONLY` is enabled can be migrated to another MySQL instance without concern that the database might be changed during the operation. See ALTER DATABASE Statement.

  A new `INFORMATION_SCHEMA` table named `SCHEMATA_EXTENSIONS` displays database options. Currently, it displays `READ ONLY=1` for read-only databases. See The INFORMATION_SCHEMA SCHEMATA_EXTENSIONS Table. (WL #13369)

## Bugs Fixed

- **InnoDB:** Code related to transaction support for histogram sampling was removed, including related assertion code that caused test failures. Transaction support is not required for histogram sampling. (Bug #31787736)
- **InnoDB:** Encryption information was not set for redo log archive log writer thread write operations. (Bug #31690196)
- **InnoDB:** The `TTASEventMutex::exit` function was optimized for ARM64. Thanks to Krunal Bauskar for the contribution. (Bug #31589019, Bug #100132)
- **InnoDB:** `InnoDB` failed to compile with the `DISABLE_PSI_RWLOCK` CMake option enabled. (Bug #31578289)
- **InnoDB:** The transaction isolation level, which is set to `READ UNCOMMITTED` for histogram sampling to avoid unnecessary lookups of old record versions, was not reset after the sampling operation completed. (Bug #31564407)
- **InnoDB:** A query that updated the clustered index of an internal temporary table returned an incorrect result. The modified pages of the clustered index were not added to the flush list resulting in lost changes when the modified pages were evicted from the buffer pool. (Bug #31560679)

  References: This issue is a regression of: Bug #29207450.
- **InnoDB:** A build dependency on the Boost library defined for the TempTable storage engine was removed. (Bug #31505048)
- **InnoDB:** A workaround was implemented to handle a Clang compiler issue in 32-bit builds that causes the `ATOMIC_LLONG_LOCK_FREE` value to be defined as “sometimes lock-free” while `__atomic_always_lock_free` returns true for the same type on the same platform. (Bug #31504609)
- **InnoDB:** A `REDUNDANT` row format table created in an earlier version of MySQL, where the row format was not defined explicitly, permitted the addition of an index that exceeded the `REDUNDANT` row format index column size limit. (Bug #31479542, Bug #99791)
- **InnoDB:** A DML operation on a column defined with a multi-valued index caused a failure. (Bug #31479282)
- **InnoDB:** A failure occurred during master key rotation. An undo tablespace in-memory object was freed prematurely. (Bug #31467626)

- **InnoDB:** Unused physical read ahead code was removed from the parallel read interface. (Bug #31429385)
- **InnoDB:** A master key rotation operation failed to skip an undo tablespace that was already truncated, which lead to an assertion failure when shutting down the server. (Bug #31400195)
- **InnoDB:** After importing a tablespace for a page-compressed table, pages were no longer compressed, and `INFORMATION_SCHEMA.INNODB_TABLESPACES` metadata incorrectly indicated that pages were compressed. The table's compression information was unavailable during the import operation. (Bug #31396947)
- **InnoDB:** A rollback and update operation after performing an instant DDL operation raised an assertion. (Bug #31391126)
- **InnoDB:** The log system (`log_sys`) sharded read-write lock caused a performance regression in CPU-bound workloads. (Bug #31389135)
- **InnoDB:** Compiling with the `UNIV_ENCRYPT_DEBUG` option enabled caused compilation errors. (Bug #31369540)
- **InnoDB:** DDL operations on a partitioned table could cause a failure. `TABLE_SHARE` and table instance objects were opened for all partitions unnecessarily. (Bug #31365127)
- **InnoDB:** After changing a `VARCHAR` column collation from `utf8mb4` to `utf8mb4_bin` in an in-place `ALTER TABLE` operation and adding an index on the same column, a case-sensitive query on the `VARCHAR` column returned an incorrect result. The `VARCHAR` column collation was changed in the data dictionary but not in the in-memory table object. Consequently, the index created on the `VARCHAR` column used stale column information causing comparisons to use the previously defined collation. (Bug #31361838)
- **InnoDB:** An `ALTER TABLE ... IMPORT TABLESPACE` operation on a large encrypted and compressed table failed with a `Page decompress failed after reading from disk` error. The decryption operation did not use the encryption block size used during encryption. Also, the encryption process did not consider compressed length, while the decryption process decrypts data by compressed length only. (Bug #31313533)
- **InnoDB:** A failure occurred during a concurrent update operation. The failure was due to an invalid previous record value. (Bug #31205266, Bug #99286)
- **InnoDB:** Upgrade from MySQL 5.7 to MySQL 8.0 failed on an instance with a table created in a general tablespace and defined with a `FULLTEXT` index. The correct data dictionary space ID for table could not determined. (Bug #31154128, Bug #99211)
- **InnoDB:** The function used to process the `SHOW ENGINE INNODB MUTEX` statement was insufficiently isolated from other threads adding new mutexes concurrently. (Bug #31105262)
- **InnoDB:** Failure to call a buffer pool page I/O completion routine resulted in orphan buffer pool I/O write pages. (Bug #31073853)
- **InnoDB:** Numerous system temporary table pages at the tail of the buffer pool flush list caused a performance degradation. The `flush_list_mutex` was held while the flush list scan traversed over system temporary table pages. The flush list scan now excludes system temporary table pages. (Bug #31060470, Bug #98974)
- **InnoDB:** The buffer control block structure (`buf_block_t`) was freed while reducing the size of the buffer pool, causing an assertion failure. The fix for this bug also backports important aspects of the fix for Bug #20735882 / Bug #76343, and replaces the internal

`buf_block_is_uncompressed()` function with the `buf_pointer_is_block_field_instance()` function. The `buf_block_is_uncompressed()` function returned false in too many cases, affecting OLTP query throughput. (Bug #31036301, Bug #31389823)

- **InnoDB:** Parallel read threads failed to respond to an explicit transaction interruption. (Bug #31016076)
- **InnoDB:** In session started with `START TRANSACTION WITH CONSISTENT SNAPSHOT`, a range query returned a truncated result. The end range flag was not reset at the beginning of the index read resulting in an aborted read and missing rows. (Bug #30950714, Bug #98642)

  References: This issue is a regression of: Bug #23481444.
- **InnoDB:** A full-text phrase search raised an assertion failure. Thanks to TXSQL (Tencent MySQL) for the contribution. (Bug #30933728, Bug #31228694)

  References: This issue is a regression of: Bug #22709692.
- **InnoDB:** A failure occurred while attempting to initialize the system tablespace on a raw disk partition. Additionally, a `INPLACE` DDL operation on the raw-disk partition tablespace failed with an error instead of switching to the `COPY` algorithm. (Bug #30867065, Bug #98091)
- **InnoDB:** LOB purge code (`lob::purge()`) did not properly handle latches taken during B-tree mini-transaction (`btr_mtr`) commit and restore operations, which could lead to conflicts between B-tree and LOB mini-transactions. (Bug #30620011)
- **InnoDB:** A long running statistics calculation operation on a large table blocked other operations requiring access to the table's statistics, causing those operations to fail. A new statistics calculation mutex was introduced, which permits concurrent access table statistics. Thanks to Kamil Holubicki for the contribution. (Bug #30607708)
- **InnoDB:** Two connections attempted to use the same transaction handler object resulting in a stalled query. (Bug #30594501)
- **InnoDB:** Shutting down the server with `innodb_fast_shutdown` setting greater than 0 raised an assertion failure. The assertion was caused by the presence of recovered transactions that were not yet rolled back. Assertion code was revised to ignore recovered transactions during a fast shutdown. Messages are now written to the error log when recovered transactions that are not rolled back are left behind by a fast shutdown. Slow shutdown now waits for recovered transactions to be rolled back. Various other shutdown logic improvements were implemented. (Bug #30226841)
- **InnoDB:** Dedicated log writer threads, introduced in MySQL 8.0.11, caused a CPU-bound performance regression on low-concurrency systems. To address this issue, the new `innodb_log_writer_threads` variable permits disabling dedicated log writer threads so that redo log records are written from the log buffer to the system buffers and flushed from the system buffers to the redo log files by each user thread, which is the behavior prior to the introduction of dedicated log writer threads. Other redo logging optimizations were implemented, including the removal of an unnecessary log closer thread that wasted CPU time, and optimizations to remedy too-aggressive checkpoint activity and excessive flush calls. The issues addressed by this fix also manifested in a `LOAD DATA` performance regression. (Bug #30088404, Bug #30003849)
- **InnoDB:** Restarting the server with an incorrect `lower_case_table_names` setting after a failure caused a hang condition. At startup, `InnoDB` waited for a transaction to roll back, but the rollback thread was not initiated due to a startup validation failure caused by the incorrect `lower_case_table_names` setting. (Bug #29833945)
- **Packaging; Group Replication:** MySQL Server Docker images did not expose the Group Replication recommended port (33061). (Bug #31627536)

- **Packaging:** Added a `client-plugins` RPM that was separated from the `client` package. It contains shared plugins for MySQL client applications. (Bug #35162346)

- **Replication:** You can now set the value of the `gtid_purged` system variable in a stored procedure, which was not previously permitted. You cannot set `gtid_purged` in a stored function. (Bug #31571427)

- **Replication:** When a replication source server shuts down and restarts, its `MEMORY` tables become empty. To replicate this effect to replicas, the first time that the source uses a given `MEMORY` table after startup, it logs an event that notifies replicas that the table must be emptied by writing a statement to the binary log to that effect. Previously, this was a `DELETE` statement, but it is now a `TRUNCATE TABLE` statement. A replica server also writes this statement to its own binary log when it shuts down and restarts. The statement is always logged in statement format, even if the binary logging format is set to `ROW`, and it is written even if `read_only` or `super_read_only` mode is set on the server. (Bug #29848785, Bug #95496)

- **Replication:** When the system variable `session_track_gtids` was set to `OWN_GTID` on a multithreaded replica, the replica's performance would degrade over time and begin to lag behind the master. The cause was the buildup of the GTIDs recorded by the replica's worker threads at each transaction commit, which increased the time taken by the worker threads to insert new ones. Session state tracking is now disabled for worker threads on a multithreaded replica. Thanks to Facebook for the contribution. (Bug #29049207, Bug #92964)

- **Replication:** When using row-based replication, the replica was allowed to use an invisible index when searching for rows to synchronize. (Bug #96148, Bug #30072179)

- **Group Replication:** X Plugin could stop unexpectedly if a Group Replication notification was issued after a new X Protocol connection was made but before the session was created. The dispatcher thread that handles Group Replication notifications now checks that the session pointer is valid. (Bug #31742798)

- **Group Replication:** Group Replication handling of memory allocation issues when adding transaction write sets has been improved. (Bug #31586243)

- **Group Replication:** While a remote cloning procedure was taking place on a joining member during distributed recovery, Group Replication considered the pre-cloning `gtid_executed` value on the joining member when identifying the common set of transactions that had been applied on all members. This meant that garbage collection for applied transactions from the group's set of certification information (shown as the `count_transactions_rows_validating` field in the Performance Schema table `replication_group_member_stats`) did not take place during the remote cloning procedure. If the remote cloning procedure took a long time, the certification information could therefore get too large to transmit to the joining member when it restarted after the remote cloning procedure, in which case an error was raised and the member was not able to join the group.

  To avoid this issue, Group Replication now considers only group members with `ONLINE` status when identifying the common set of transactions that have been applied on all members. When a joining member enters `ONLINE` state after distributed recovery, its certification information is updated with the certification information from the donor at the time when the member joined, and garbage collection takes place for this on future rounds.

  As a workaround for this issue in earlier releases, after the remote cloning operation completes, wait two minutes to allow a round of garbage collection to take place to reduce the size of the group's certification information. Then issue the following statement on the joining member, so that it stops trying to apply the previous set of certification information:

  ```
  RESET SLAVE FOR CHANNEL group_replication_recovery;
  ```

(Bug #31446381, Bug #99778)

- **Group Replication:** It was possible for a group member that left the group due to a communication error to reconnect between auto-rejoin attempts while the auto-rejoin procedure was still ongoing, which left Group Replication unable to function on the member. Group Replication error management and member status handling has now been corrected to prevent this situation. (Bug #31401797)

- **Microsoft Windows:** On Windows, build targets could fail if the build was on a file system root, such as `R:/`. (Bug #31315467)

- **JSON:** `JSON_OBJECT()` did not always perform proper checking for `NULL` values. (Bug #31393934)

- The new `WITH_SYSTEMD_DEBUG` `CMake` option, if enabled, produces additional systemd debugging information, for platforms on which systemd is used to run MySQL. The default is `OFF`. (Bug #31788834)

- For RPM and Debian packages, client-side plugins were moved from the server package to the client package in MySQL 8.0.21. This could cause failures relating to LDAP authentication plugins when upgrading from 5.7 packages to 8.0 packages. Packaging adjustments were made to avoid this problem. (Bug #31782612)

  References: This issue is a regression of: Bug #31123564, Bug #31336340.

- The timestamp written for the `ts` key by the `log_sink_json` JSON-format error log sink did not have the same value as other timestamps in the same log message. (Bug #31749103)

- Kerberos authentication for the SASL LDAP authentication plugin incorrectly handled failure to acquire a ticket-granting ticket. (Bug #31727195)

- For some third-party libraries, enabling link-time optimization caused build failures. (Bug #31701553, Bug #100410)

- Printing an excessively long diagnostic message could cause the server to exit unexpectedly. (Bug #31686926)

- A page-compressed table was cloned as an uncompressed table. The associated tablespace object, which includes a compression flag, was not initialized prior to the cloning operation. (Bug #31677990, Bug #100243)

- Certain cases of successful LDAP authentication could cause the server to hang. (Bug #31661437)

- During transformation of a grouped query into a derived table, when the `WHERE` clause and the `HAVING` clause became part of the derived table, the condition count was not updated for the derived table. This resulted in reduced memory allocation while creating keys for `ref` access. (Bug #31661309)

- When a value was compared using `LIKE` with a table column not defined as one of the MySQL string types, the server sometimes did not raise the expected error. (Bug #31659015)

- The `acquire_related()` service function returned the default service in some cases when it should have returned an error. (Bug #31655906)

- In bootstrapping mode, certain multiple-statement transactions could cause unexpected server behavior. (Bug #31650096)

- A remote cloning operation checked for the availability of a plugin on the recipient that was removed from the donor instance previously. References to the uninstalled plugin had not been released. Error reporting issues related to plugin mismatches and availability were also addressed. (Bug #31639732, Bug #100244)

- In debug builds, the server attempted to evaluate subqueries while creating a view. (Bug #31590301)

  References: This issue is a regression of: Bug #25466100.

- A condition using `RAND()` was not pushed down even in cases where it was safe to do so, that is when no windowing function or `GROUP BY` is in use. (Bug #31587575)

- While pushing conditions down to a derived table, a constant condition such as `WHERE FALSE` or `WHERE TRUE` was pushed down to the first table in the derived table, which is not necessary as the condition has nothing to do with the derived table. MySQL now avoids pushing constant conditions down to derived tables in such cases.

  In addition used tables are now updated for the condition that needs to be pushed down to the derived table, following code inspection revealing that this was not done after replacing the columns in the condition with the derived table expressions. (Bug #31587493)

- A query using `WHERE column > (... IN (SELECT ...))` could sometimes trigger an assertion in the range optimizer. (Bug #31586906)

  References: This issue is a regression of: Bug #30473261.

- It was possible for `ANALYZE TABLE` to fail with `Duplicate key error` if a row was inserted in the interval between the check for the existence of the row and the execution of the insert, and the statistics table was updated concurrently. `ANALYZE TABLE` now ignores the error in this situation. (Bug #31582758)

- The range optimizer does not use the correct lock type after cloning the handler needed to perform merged scans, and instead used a read lock unconditionally. This resulted in various different side effects for different scenarios.

  For example, a `SELECT` with `FOR UPDATE` requests a write lock, but after cloning the handler for an index merge scan, the range optimizer requested a read lock which resulted in a mismatch. Similarly, for data dictionary tables, the lock type was set to `LOCK_NONE` due to the special handling required for such tables.

  To prevent this problem from occurring, we now ensure that the original lock type of the handler is always used in the cloned handler as well. (Bug #31582383)

- In some cases, a query using an `ANY` subquery gave an incorrect result when the `subquery_to_derived` optimizer switch was enabled. (Bug #31566339)

- When `FALSE AND condition` was simplified as `FALSE`, temporary table resources allocated for the `condition` were not always released afterwards. (Bug #31565009)

- A value equal to `ULLONG_MAX` could be inserted into a `BIT(64)` column, but not retrieved. (Bug #31564742, Bug #100053)

- While removing an unused window definition, a subquery that was part of an `ORDER BY` was not removed. The optimizer then tried to optimize the subquery without locking the tables. Now, when removing an unused window definition, the server cleans up any subqueries present as part of the definition. (Bug #31518806)

  References: This issue is a regression of: Bug #27062031.

- A coding problem introduced in MySQL 8.0.20 could cause client applications to exit unexpectedly during shutdown. (Bug #31515752)

  References: This issue is a regression of: Bug #27045306.

- Added a missing error code translation from ICU `U_REGEX_NUMBER_TOO_BIG` to MySQL `ER_REGEX_NUMBER_TOO_BIG`. (Bug #31514995)
- Merging during `filesort` operations could fail to remove duplicates for queries that used `DISTINCT`. (Bug #31498664, Bug #99900)
- MySQL's internal `DYNAMIC_STRING` class formerly allocated memory in a linear fashion, that is, by a predetermined number of bytes. The class has been revised such that it now allocates memory exponentially, which should make operations such as repeated string appends more efficient. (Bug #31491799)
- `LOCK_mutex` mishandling could result in a memory leak. (Bug #31491146)
- A newly added collation was not added and could cause an unexpected exit on shutdown. (Bug #31470422)
- On Windows, file name reuse by the `GetTempFileName()` function could cause an assertion to be raised. (Bug #31468590)
- A `LATERAL` subquery was incorrectly converted into an antijoin. (Bug #31465717)
- `NATURAL JOIN` evaluation could inadvertently match hidden virtual columns created by functional indexes. (Bug #31463511, Bug #99807)
- Sort keys for string hash join keys using more than 1024 bytes were not handled correctly by the server. (Bug #31437753)
- The server attempted to delete from a view whose definition included `HAVING` when the `HAVING` clause was constant and evaluated as true even though a view with `HAVING` as part of its definition should not be updatable. (Bug #31429865)
- Privilege requirements were checked incorrectly for the `INFORMATION_SCHEMA.USER_ATTRIBUTES` table. (Bug #31427410)
- When the internal function `replace_index_subquery()` failed, the server still attempted to create iterators for the affected subquery. Now the function raises a clear error instead. (Bug #31427072)
- A query using `WHERE NOT EXISTS (SELECT` *`const`* `FROM` *`table`* `WHERE` *`column`*`=FROM_UNIXTIME(`*`value`*`)` was not handled correctly. (Bug #31425664)
- In some cases, `key_hint` handling was improperly applied to derived and internal temporary tables. (Bug #31424455)
- Re-execution of prepared `INSERT` statements could fail for inserts through a view. (Bug #31417951)
- `JSON` scalar evaluation could enter an infinite loop. (Bug #31406724)
- The `user_attributes` column in `mysql.user` table rows could be affected incorrectly by partial revokes. (Bug #31405985)
- Improper window function initialization could cause a server exit. (Bug #31389573, Bug #31437834)
- Sensitive LDAP authentication plugin system variables now display as asterisks when retrieved in SQL statements. (Bug #31388444, Bug #31391864)
- `mysql-test-run.pl` tests under no-threads connection handling failed with ASAN builds due to improper resource group initialization. This has been fixed. Thanks to Xiaoyu Wang, Tencent Technology for the contribution. (Bug #31378900, Bug #99609)

- Using the `authentication_ldap_simple` authentication plugin with SSL could cause a segmentation fault during shutdown. (Bug #31364927)
- Killing a query could raise spurious assertions in the hash join iterator. (Bug #31361354)
- In some cases, an outer reference that was not `LATERAL` was not marked as read-only as expected. (Bug #31359965)
- A failure occurred when upgrading from MySQL 5.7 to MySQL 8.0 due to invalid references to orphaned events (events for which a database no longer exists). The server now fails with an appropriate error messages when orphaned events are encountered during upgrade. Error messages for orphaned stored routines were also revised. (Bug #31335554)
- Enabling the `create_admin_listener_thread` system variable could cause a server exit during startup. (Bug #31335279)
- After `ALTER TABLE` to add an expression default to a column, the first insert inserted a value as if the expression had been evaluated at alter time and not insert time. (Bug #31330789, Bug #99513)
- The LDAP authentication plugins did not properly compare the user-supplied authentication method against the permitted methods. (Bug #31320532)
- Certain views could cause a following `USE` statement to result in an unexpected server exit. (Bug #31311312)
- When a filesort sorted a buffer and `LIMIT` was active, it first sorted all rows and then discarded those that did not fit within the limit, which required sorting many rows that were certain to be discarded later. Now the optimizer sorts only the rows actually needed. Internal testing shows that this change can speed up the sort phase for a simple string sorting benchmark (as measured by `EXPLAIN ANALYZE`) by up to 15%. (Bug #31303537)
- A dynamic range scan runs the range optimizer for each row fetched from the first table in a join to determine whether a range scan can be picked for the second table using the value available from that row. If the row contains no usable indexes, a table scan may be chosen instead. For the query giving rise to this issue, a table scan is chosen once, followed by a range scan on a non-covering index, and the dynamic range iterator has two read sets which are used for both these cases. One of these, used for the table scan, includes the base columns of generated columns required for processing the query; the other read set does not include the base columns in the read set used for range scans. This is because, for covering indexes, the read set should not include base columns to avoid adding unneeded columns by hash join or batched key access. The issue arose because the second read set was also used for a non-covering index, which resulted in an assert.

  To prevent this from happening, when initializing a table read set in the dynamic range iterator, we now make sure that it includes the base columns when the range optimizer picks a non-covering index. (Bug #31280526)

  References: This issue is a regression of: Bug #30417361.
- It was possible to insert an out-of-range value for a `TIMESTAMP` if it included a timezone offset. (Bug #31239157)
- The `keyring_hashicorp` keyring plugin did not limit the size of keys for key operations. (Bug #31205715)
- Configuring with `-DWITH_ZSTD=system` failed for older versions of the `zstd` library. `CMake` now checks the `zstd` version and requires at least 1.0.0 for compilation, 1.2.0 to run compression checks. (Bug #31174920, Bug #99241)

- In some cases, a `SELECT` that obtained status variable information from Performance Schema tables and that included a sort by a column containing temporal values was not handled correctly. (Bug #31168097)

- In some cases, `ROUND()` and `TRUNCATE()` did not return the data type of their first arguments as expected. This fix insures that return types from these functions follow these rules, where the first argument is of the type shown:

  - For any integer type, the return type is `BIGINT`.
  - For any floating-point type or any non-numeric type, the return type is `DOUBLE`.
  - For `DECIMAL`, the return type is also `DECIMAL`.
  - The type attributes for the return value are also copied from the first argument, except in the case of `DECIMAL`, when the second argument is a constant value.
  - When the desired number of decimal places is less than the scale of the argument, the scale and the precision of the result are adjusted accordingly. In addition, for the `ROUND()` function, the precision is extended by one place to accomodate rounding that increases the number of significant digits. If the second argument is negative, the return type is adjusted such that its scale is 0, with a corresponding precision.

  For more information, see the description of the `ROUND()` function. (Bug #31128028)

- A `SELECT ... FOR SHARE` statement now only requires the `SELECT` privilege. Previously, the `SELECT` privilege was required with at least one of the `DELETE`, `LOCK TABLES`, or `UPDATE` privileges. (Bug #31096384, Bug #99101)

- A semijoin strategy was chosen for the join of a correlated subquery having a `LIMIT` clause and requiring a row other than the first, which caused the `LIMIT` clause to be ignored and invalid rows to be returned. Now, when `LIMIT` used with this type of join specifies a row other than the first row, or more than one row, the semijoin strategy is no longer employed. (Bug #31096309)

- After the fix for Bug #81009, privilege checks for truncating Performance Schema tables were too restrictive when `read_only` or `super_read_only` were enabled, causing truncation to fail even for users with appropriate table privileges. (Bug #31080309, Bug #99072)

  References: This issue is a regression of: Bug #81009.

- `ORDER BY` did not work as expected for queries with `ROLLUP` in which window functions were also used. (Bug #31073133)

- Some `INSERT` statements were not handled correctly. (Bug #31072198)

- Date interval calculations checked for overflow but not underflow. Now they check for both. (Bug #31054071)

- If an XA prepared transaction rollback XID was incorrectly formatted, the transaction remained in recovered state for `XA COMMIT` and `XA ROLLBACK` statements (or raised an assertion for debug builds) rather that reporting an error. (Bug #31030205)

- Database-level privileges inherited through a role were not handled properly for database names that contained wildcard characters. (Bug #31013538, Bug #98876)

- When the `--local` option was given, `mysqlimport` mishandled the `MYSQL_OPT_LOAD_DATA_LOCAL_DIR` option for `mysql_options()` so that it had no effect. (Bug #31001550)

- Certain prepared statements could cause an unexpected server exit. (Bug #30943963)
- `OPTIMIZE TABLE` for `MyISAM` tables could cause table size to increase and query performance to decrease. `REPAIR TABLE` for `MyISAM` tables could cause the `Table is already up to date` status produced by a previous `OPTIMIZE TABLE` to be lost. (Bug #30869674, Bug #98511, Bug #29755517)
- `mysqlpump` object validation included objects in excluded databases. (Bug #30819012)
- Inserting a `TIMESTAMP` value having a timezone offset which also had a zero for the month, day, or both, led to an assert. Such a value should be and is now rejected, regardless of the `sql_mode` setting. (Bug #30786762)

  References: See also: Bug #31239157.
- Privileges granted using roles could be mishandled at the column-privilege level. (Bug #30660403, Bug #97971)
- Comparison of a `TIME` value with `NULL` in some cases raised an assertion. (Bug #30324587)

  References: This issue is a regression of: Bug #25949639.
- LDAP authentication plugins enforced CA verification incorrectly, which could result in use of an incorrect CA. (Bug #30220357)
- `ORDER BY` queries were not executed correctly when `sort_buffer_size` and `max_sort_length` were set to values which caused the internal limit on the maximum number of keys allowed per sort buffer to be set to 0. (Bug #30175483)
- A large number of nested arguments in full-text search query caused an error. (Bug #29929684)
- A potential misreporting of memory use by the Performance Schema has been corrected. (Bug #29912403)
- When `explicit_defaults_for_timestamp` was disabled and a `NULL` was inserted into a generated column declared as `TIMESTAMP NOT NULL`, the server would attempt to convert the inserted value to `CURRENT_TIMESTAMP`. Such an insertion is now rejected with `ER_BAD_NULL_ERROR`. (Bug #29449518)
- The `SET_VAR` hint did not accept a floating point value specified as a system variable setting. (Bug #29349748)
- Previously, when `NULL` was used as the format argument to `STR_TO_DATE()`, irrelevant warnings were printed. Now, when `NULL` is passed to it, the function returns `NULL`. (Bug #27265863)
- In some cases, incorrect use of `IS NULL` generated multiple warnings about invalid arguments. (Bug #27264652)
- Resolving an `ORDER BY` column that referred to a `SELECT` list column from a derived table was not performed correctly when executing certain prepared statements. (Bug #26808862)
- When using `EXPLAIN` on a multi-table `UPDATE` statement in which a generated column was referenced in a condition, the output always showed the table containing this column as being updated, whether the table was actually updated or not. (Bug #22671310)
- An assertion could be raised when the SQL layer passed incorrect information to `InnoDB` about the type of operation to be performed on a temporary table. (Bug #22503696)

- This construct works for base tables to insert a row using all default values but failed for views:

  ```
  INSERT INTO name () VALUES ();
  ```

  (Bug #15988466, Bug #67863)

- In some cases, the server issued an error when an invisible index was used in an index hint even when the `use_invisible_indexes` optimizer switch was not set to `OFF`. (Bug #100024, Bug #31550839)

- Regular expression functions such as `REGEXP_LIKE()` yielded inconsistent results with binary string arguments. These functions now reject binary strings with an error. (Bug #98950, Bug #98951, Bug #31031886, Bug #31031888)

- When range values specified in a predicate are not compatible with the data type of the column with which the values are compared, the range optimizer rounds off the range values and assigns certain flags so that it does not exclude rows that qualify for the range because of rounding. In the specific query that triggered the reported issue, a column named `id` of type `INT` was tested using `id NOT IN (-0.1, 0.1)`, and the values being tested are rounded to integers, with the predicate thus being treated as `NOT IN (0,0)`. The optimizer then treats this as the intervals `id < 0` and `0 < id < 0`, but in this case it also set a flag to a value that indicated that reads should begin following rows containing 0 for the value to be compared. Now in such cases, the flag is set in such a way that the values which have been rounded are treated correctly. (Bug #98826, Bug #30988735)

  References: This issue is a regression of: Bug #80244, Bug #22661012.

- For a view based on a join having an updatable part and one that was not, the error message generated when attempting to update a column of this view that was not updatable referenced the source table or view instead of the view actually named in the offending `UPDATE` statement. (Bug #80655, Bug #22891840)

# Changes in MySQL 8.0.21 (2020-07-13)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

In the documentation for MySQL 8.0.21, we have started changing the term “master” to “source”, the term “slave” to “replica”, the term “whitelist” to “allowlist”, and the term “blacklist” to “blocklist”. There are currently no changes to the product's syntax, so these terms are still present in the documentation where the current code requires their use. See the blog post MySQL Terminology Updates for more information.

- Account Management Notes
- C API Notes
- Compilation Notes
- Configuration Notes
- Connection Management Notes
- Deprecation and Removal Notes
- JSON Notes
- Optimizer Notes
- Packaging Notes

- Pluggable Authentication
- Security Notes
- Test Suite Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Account Management Notes

- You can now set per-user comments and attributes when creating or updating MySQL user accounts. A user comment consists of arbitrary text passed as the argument to a `COMMENT` clause used with a `CREATE USER` or `ALTER USER` statement. A user attribute consists of data in the form of a JSON object passed as the argument to an `ATTRIBUTE` clause used with either of these two statements. The attribute can contain any valid key-value pairs in JSON object notation.

  For example, the first of the following two statements creates a user account `bill@localhost` with the comment text `This is Bill's user account`. The second statement adds a user attribute to this account, using the key `email`, with the value `bill@example.com`.

  ```
  CREATE USER 'bill'@'localhost' COMMENT 'This is Bill\'s user account';

  ALTER USER 'mary'@'localhost'
      ATTRIBUTE '{"email":"bill@example.com"}';
  ```

  Only one of `COMMENT` or `ATTRIBUTE` can be used in the same `CREATE USER` or `ALTER USER` statement.

  User comments and user attributes are stored together internally as a JSON object, with the comment text as the value of an element with the key `comment`. You can information retrieve user comments and user attributes from the `ATTRIBUTE` column of the `INFORMATION_SCHEMA.USER_ATTRIBUTES` table; since this data is in JSON format, you can work with it using MySQL's JSON function and operators (see JSON Functions). Changes to an existing user attribute are merged with its current value, as you had used `JSON_MERGE_PATCH()`; new key-value pairs are appended to the attribute, and new values for existing keys overwrite their previous values.

  To remove a given key-value pair from a user attribute, use `ALTER USER` *`user`* `ATTRIBUTE '{"`*`key`*`":null}'`.

  For more information and examples, see CREATE USER Statement, ALTER USER Statement, and The INFORMATION_SCHEMA USER_ATTRIBUTES Table. (WL #13562)

  References: See also: Bug #31067575.

## C API Notes

- Per OpenSSL recommendation, `x509_check_host()` and `X509_check_ip_asc()` calls in the C client library were replaced, respectively, with `X509_VERIFY_PARAM_set1_host()` and `X509_VERIFY_PARAM_set1_ip_asc()` calls. (Bug #29684791)

- The MySQL C API now supports compression for asynchronous functions. This means that the `MYSQL_OPT_COMPRESSION_ALGORITHMS` and `MYSQL_OPT_ZSTD_COMPRESSION_LEVEL` options for the `mysql_options()` function now affect asynchronous operations, not just synchronous operations. See mysql_options().

Our thanks to Facebook for the contribution. (Bug #96802, Bug #30284871, WL #13510)

## Compilation Notes

- The minimum version of the Boost library for server builds is now 1.72.0. (Bug #30963985)

## Configuration Notes

- `tcmalloc` is no longer a permitted value for the `mysqld_safe --malloc-lib` option. (Bug #31372027)

## Connection Management Notes

- MySQL Server supports a “main” network interface for ordinary client connections, and optionally an administrative network interface for administrative client connections. Previously, the main and administrative interfaces used the same TLS configuration, such as the certificate and key files for encrypted connections. It is now possible to configure TLS material separately for the administrative interface:

  - There are new configuration parameters that apply specifically to the administrative interface.
  - The `ALTER INSTANCE RELOAD TLS` statement is extended with a `FOR CHANNEL` clause that enables specifying the channel (interface) for which to reload the TLS context.
  - The new Performance Schema `tls_channel_status` table exposes TLS context properties for the main and administrative interfaces.
  - For backward compatibility, the administrative interface uses the same TLS context as the main interface unless some nondefault TLS parameter value is configured for the administrative interface.

  For more information, see Administrative Interface Support for Encrypted Connections, ALTER INSTANCE Statement, and The tls_channel_status Table. (WL #13850)

## Deprecation and Removal Notes

- **Partitioning:** Columns with index prefixes are not supported as part of a table's partitioning key; previously such columns were simply omitted by the server when referenced in creating, altering, or upgrading a table that was partitioned by key, with no indication that this omission had taken place, except when the proposed partitioning function used only columns with prefixes, in which case the statement failed with an error message that did not identify the actual source of the problem. This behavior is now deprecated, and subject to removal in a future release in which using any such columns in the proposed partitioning key will cause the `CREATE TABLE` or `ALTER TABLE` statement in which they occur to be rejected.

  When one or more columns using index prefixes are specified as part of the partitioning key, a warning is now generated for each such column. In addition, when a `CREATE TABLE` or `ALTER TABLE` statement is rejected because all columns specified in the proposed partitioning key employ index prefixes, the error message returned now makes clear the reason the statement did not succeed. This includes cases in which the columns proposed the partitioning function are defined implicitly as those in the table's primary key by employing an empty `PARTITION BY KEY()` clause.

  For more information and examples, see Column index prefixes not supported for key partitioning, and KEY Partitioning. (Bug #29941932, Bug #29941959, Bug #31100205, WL #13588)

  References: See also: Bug #29942014.

## JSON Notes

- Added the `JSON_VALUE()` function, which simplifies creating indexes on `JSON` columns. A call to `JSON_VALUE(json_doc, path RETURNING type)` is equivalent to calling `CAST( JSON_UNQUOTE( JSON_EXTRACT(json_doc, path) ) AS type)`, where *`json_doc`* is a JSON document, *`path`* is a JSON path expression pointing to a single value within the document, and *`type`* is a data type compatible with *`CAST()`*. `RETURNING type` is optional; if no return type is specified, `JSON_VALUE()` returns `VARCHAR(512)`.

  `JSON_VALUE()` also supports `ON EMPTY` and `ON ERROR` clauses similar to those used with `JSON_TABLE()`.

  You can create indexes on a `JSON` column using `JSON_VALUE()` as shown here:

  ```
  CREATE TABLE inventory(
      items JSON,
      INDEX i1 ( (JSON_VALUE(items, '$.name' RETURNING CHAR(50))) ),
      INDEX i2 ( (JSON_VALUE(items, '$.price' RETURNING DECIMAL(5,2))) ),
      INDEX i3 ( (JSON_VALUE(items, '$.quantity' RETURNING UNSIGNED)) )
  );
  ```

  Assuming the `items` column contains values such as `'{"name": "hat", "price": "22.95", "quantity": "17"}'`, you can issue queries, such as the following, that are able to use these indexes:

  ```
  SELECT items->"$.price" FROM inventory
      WHERE JSON_VALUE(items, '$.name' RETURNING CHAR(50)) = "hat";

  SELECT * FROM inventory
      WHERE JSON_VALUE(items, '$.price' RETURNING DECIMAL(5,2)) <= 100.01;

  SELECT items->"$.name" AS item, items->"$.price" AS amount
      FROM inventory
      WHERE JSON_VALUE(items, '$.quantity' RETURNING UNSIGNED) > 500;
  ```

  For more information and examples, see the description of the `JSON_VALUE()` function. (WL #12228)

## Optimizer Notes

- MySQL attempts to use an ordered index for any `ORDER BY` or `GROUP BY` query that has a `LIMIT` clause, overriding any other choices made by the optimizer, whenever it determines that this would result in faster execution. Because the algorithm for making this determination makes certain assumptions about data distribution and other conditions, it may not always be completely correct, and it is possible in some cases that choosing a different optimization for such queries can provide better performance. To handle such occurrences, it is now possible to disable this optimization by setting the `optimizer_switch` system variable's `prefer_ordering_index` flag to `off`.

  For more information about this flag and examples of its use, see Switchable Optimizations, and LIMIT Query Optimization.

  Our thanks to Jeremy Cole for the contribution. (Bug #97001, Bug #30348211, WL #13929)

  References: See also: Bug #31686878.

- A single-table `UPDATE` or `DELETE` statement that uses a subquery having a `[NOT] IN` or `[NOT] EXISTS` predicate can now in many cases make use of a semijoin transformation or subquery materialization. This can be done when the statement does not use `LIMIT` or `ORDER BY`, and when semijoin or subquery materialization is allowed by any optimizer hints used in the subquery, or by the value of the `optimizer_switch` server system variable.

You can see when the semijoin optimization or subquery materialization is used for an eligible single-table `DELETE` or `UPDATE` due to the presence of a `join_optimization` object in the optimizer trace. You can also see that the conversion is performed by checking the output of `EXPLAIN FORMAT=TREE`; if the optimization is not performed, this shows `<not executable by iterator executor>`, while a multi-table statement reports a full plan.

As part of this work, semi-consistent reads are now supported by multi-table `UPDATE` of `InnoDB` tables, when the transaction isolation level is weaker than `REPEATABLE READ`. (Bug #35794, Bug #96423, Bug #11748293, Bug #30139244, WL #6507)

- Added the `optimizer_switch` flag `subquery_to_derived`. When this flag is set to `on`, the optimizer transforms eligible scalar subqueries into left outer joins (and in some cases, inner joins) on derived tables. This optimization can be applied to a subquery which meets the following conditions:

  - It uses one or more aggregate functions but no `GROUP BY`.
  - It is part of a `SELECT`, `WHERE`, `JOIN`, or `HAVING` clause.
  - It is not a correlated subquery.
  - It does not make use of any nondeterminstic functions.

  `ANY` and `ALL` subqueries which can be rewritten to use `MIN()` or `MAX()` are also not affected.

  With `subquery_to_derived=on`, the optimization can also be applied to a table subquery which is the argument to `IN`, `NOT IN`, `EXISTS`, or `NOT EXISTS`, and which does not contain a `GROUP BY` clause.

  The `subquery_to_derived` flag is set to `off` by default, since it generally does not improve performance, and its intended use for the most part is for testing purposes.

  For more information, see Switchable Optimizations, for more information and examples. See also Optimizing Derived Tables, View References, and Common Table Expressions with Merging or Materialization, and LIMIT Query Optimization. (WL #13851)

- Building on work done in MySQL 8.0.18, the server now performs injection of casts into queries to avoid mismatches when comparing string data types with those of numeric or temporal types; as when comparing numeric and temporal types, the optimizer now adds casting operations in the item tree inside expressions and conditions in which the data type of the argument and the expected data type do not match. This makes queries in which string types are compared with numeric or temporal types equivalent to queries which are compliant with the SQL standard, while maintaining backwards compatibility with previous releases of MySQL. Such casts are now performed whenever string values are compared to numeric or temporal values using any of the standard numeric comparison operators (`=`, `>=`, `>`, `<`, `<=`, `<>`/`!=`, and `<=>`).

  Such implicit casts are now performed between a string type (`CHAR`, `VARCHAR`, `BINARY`, `VARBINARY`, `BLOB`, `TEXT`, `ENUM`, or `SET`) and a numeric type (`SMALLINT`, `TINYINT`, `MEDIUMINT`, `INT`/`INTEGER`, `BIGINT`; `DECIMAL`/`NUMERIC`; `FLOAT`, `DOUBLE`, `REAL`; and `BIT`) by casting the string value to `DOUBLE`; if the numeric value is not already of type `DOUBLE`, `FLOAT`, or `REAL`, it is also cast to `DOUBLE`. A `YEAR` value is also cast to `DOUBLE` when compared with a string value (as is the string value). For such comparisons between string types and `TIMESTAMP` or `DATETIME` values, the arguments are cast as `DATETIME`; when a string type is compared with a `DATE` value, the string is cast to `DATE`.

  For example, a query such as `SELECT * FROM t1 JOIN t2 ON t1.char_col = t2.int_col` is rewritten and executed as `SELECT * FROM t1 JOIN t2 ON CAST(t1.char_col AS DOUBLE) = CAST(t2.int_col AS DOUBLE)`, and `SELECT * FROM t1 JOIN t2 ON t1.varchar_col =`

`t2.timestamp_col` is transformed to `SELECT * FROM t1 JOIN t2 ON CAST(t1.varchar_col AS DATETIME) = CAST(t2.timestamp_col AS DATETIME)` prior to execution.

You can see when casts are injected into a given query by viewing the output of `EXPLAIN ANALYZE`, `EXPLAIN FORMAT=JSON`, or `EXPLAIN FORMAT=TREE`. `EXPLAIN [FORMAT=TRADITIONAL]` can also be used, but in this case it is necessary, following execution of the `EXPLAIN` statement, to issue `SHOW WARNINGS` to view the rewritten query.

This change is not expected to cause any difference in query results or performance. (WL #13456)

## Packaging Notes

- For RPM and Debian packages, client-side plugins were moved from the server package to the client package. Additionally, debug versions of client-side plugins were moved to the test package. (Bug #31123564, Bug #31336340)
- MSI packages for Windows no longer include the legacy server data component. (Bug #31060177)
- The bundled Protobuf library was upgraded from version 3.6.1 to version 3.11. (Bug #31000511, Bug #98852)
- The `libevent` library bundled with MySQL was upgraded to version 2.1.11. In addition, for the `WITH_LIBEVENT` `CMake` option, the following two changes were made:

  1. `yes` is no longer permitted as a synonym for `system`. Use `system` instead.
  2. If `system` is specified but no system `libevent` is found, the bundled version is no longer used in place of the missing system library, and an error occurs instead.

  (Bug #30926742)
- The ICU (International Components for Unicode) library bundled with MySQL has been upgraded to version 65.1.

## Pluggable Authentication

- The MySQL Enterprise Edition `authentication_ldap_sasl` plugin that implements SASL LDAP authentication supports multiple authentication methods, but depending on host system configuration, they might not all be available. The new `Authentication_ldap_sasl_supported_methods` status variable provides discoverability for the supported methods. Its value is a string consisting of supported method names separated by spaces. Example: `"SCRAM-SHA1 GSSAPI"` (WL #13876)

## Security Notes

- **Incompatible Change:** Access to the `INFORMATION_SCHEMA.FILES` table now requires the `PROCESS` privilege.

  This change affects users of the `mysqldump` command, which accesses tablespace information in the `FILES` table, and thus now requires the `PROCESS` privilege as well. Users who do not need to dump tablespace information can work around this requirement by invoking `mysqldump` with the `--no-tablespaces` option. (Bug #30350829)
- For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.1.1g. Issues fixed in the new OpenSSL version are described at https://www.openssl.org/news/cl111.txt and https://www.openssl.org/news/vulnerabilities.html. (Bug #31296697)

- Previously, `LOCAL` data loading capability for the `LOAD DATA` statement could be controlled on the client side only by enabling it for all files accessible to the client, or by disabling it altogether. The new `MYSQL_OPT_LOAD_DATA_LOCAL_DIR` option for the `mysql_options()` C API function enables clients to restrict `LOCAL` data loading to files located in a designated directory. See Security Considerations for LOAD DATA LOCAL. (WL #13168)

## Test Suite Notes

- `mysql-test-run.pl` no longer accepts unique prefixes of command options. Complete option names must be given. (Bug #31390127)
- MySQL tests were updated to use googletest 1.10.0. (Bug #31364750)
- The `innodb.innodb_mysql` test case was updated to avoid nondeterminism of output row order. Thanks to Facebook for the contribution. (Bug #30810572, Bug #98377)
- `mysql-test-run.pl` now supports an `--mtr-port-exclude` option for specifying the range of ports to exclude when searching for available port ranges to use. The `MTR_PORT_EXCLUDE` environment variable may also be set to achieve the same effect. Thanks to Facebook for the contribution. (Bug #30809607)
- In addition to aborting on receipt of **CTRL**+**C** (SIGINT), `mysql-test-run.pl` now also displays a list of test cases that failed up to that point. (Bug #30407014)

## X Plugin Notes

- Where a dollar sign ($) was used to reference an entire document, X Plugin handled the reference differently depending on the context in which it was used. This has now been standardized. (Bug #31374713)
- With certain settings for the global SQL mode, X Plugin's authentication process failed to accept a correct user password. The authentication process now operates independently from the global SQL mode's setting to ensure consistency. (Bug #31086109)

## Functionality Added or Changed

- **Important Change; Group Replication:** By default, a replication source server writes a checksum for each event in the binary log, as specified by the system variable `binlog_checksum`, which defaults to the setting `CRC32`. Previously, Group Replication did not support the presence of checksums in the binary log, so `binlog_checksum` had to be set to `NONE` when configuring a server instance that would become a group member. This requirement is now removed, and the default can be used. The setting for `binlog_checksum` does not have to be the same for all members of a group.

  Note that Group Replication does not use checksums to verify incoming events on the `group_replication_applier` channel, because events are written to that relay log from multiple sources and before they are actually written to the originating server's binary log, which is when a checksum is generated. Checksums are used to verify the integrity of events on the `group_replication_recovery` channel and on any other replication channels on group members. (WL #9038)
- **Performance:** Improved the implementation of the `UNHEX()` function by introducing a lookup table for mapping a hexadecimal digit string to its binary representation. This change speeds up execution of the function by a factor of 8 or more in testing. (Bug #31173103)
- **InnoDB:** Redo logging can now be enabled and disabled using `ALTER INSTANCE {ENABLE|DISABLE} INNODB REDO_LOG` syntax. This functionality is intended for loading data into a new MySQL instance. Disabling redo logging helps speed up data loading by avoiding redo log writes.

The new `INNODB_REDO_LOG_ENABLE` privilege permits enabling and disabling redo logging.

The new `Innodb_redo_log_enabled` status variable permits monitoring redo logging status.

See Disabling Redo Logging. (WL #13795)

- **InnoDB:** Truncating an undo tablespace on a busy system could affect performance due to associated flushing operations that remove old undo tablespace pages from the buffer pool and flush the initial pages of the new undo tablespace to disk. To address this issue, the flushing operations were removed.

  Old undo tablespace pages are now released passively as they become least recently used, or are removed at the next full checkpoint. The initial pages of the new undo tablespace are now redo logged instead of flushed to disk during the truncate operation, which also improves durability of the undo tablespace truncate operation.

  To prevent potential issues caused by an excessive number of undo tablespace truncate operations, truncate operations on the same undo tablespace between checkpoints are now limited to 64. If the limit is exceeded, an undo tablespace can still be made inactive, but it is not truncated until after the next checkpoint.

  `INNODB_METRICS` counters associated with defunct undo truncate flushing operations were removed. Removed counters include: `undo_truncate_sweep_count`, `undo_truncate_sweep_usec`, `undo_truncate_flush_count`, and `undo_truncate_flush_usec`.

  See Undo Tablespaces. (WL #11819)

- **InnoDB:** At startup, `InnoDB` validates the paths of known tablespace files against tablespace file paths stored in the data dictionary in case tablespace files have been moved to a different location. The new `innodb_validate_tablespace_paths` variable permits disabling tablespace path validation. This feature is intended for environments where tablespaces files are not moved. Disabling tablespace path validation improves startup time on systems with a large number of tablespace files.

  For more information, see Disabling Tablespace Path Validation. (WL #14008)

- **InnoDB:** Table and table partition data files created outside of the data directory using the `DATA DIRECTORY` clause are now restricted to directories known to `InnoDB`. This change permits database administrators to control where tablespace data files are created and ensures that the data files can be found during recovery.

  General and file-per-table tablespaces data files (`.ibd` files) can no longer be created in the undo tablespace directory (`innodb_undo_directory`) unless that directly is known to `InnoDB`.

  Known directories are those defined by the `datadir`, `innodb_data_home_dir`, and `innodb_directories` variables.

  Truncating an `InnoDB` table that resides in a file-per-table tablespace drops the existing tablespace and creates a new one. As of MySQL 8.0.21, `InnoDB` creates the new tablespace in the default location and writes a warning to the error log if the tablespace was created with an earlier version and the current tablespace directory is unknown. To have `TRUNCATE TABLE` create the tablespace in its current location, add the directory to the `innodb_directories` setting before running `TRUNCATE TABLE`. (WL #13065)

- **InnoDB:** To improve concurrency for operations that require access to lock queues for table and row resources, the lock system mutex (`lock_sys->mutex`) was replaced by sharded latches, and lock queues were grouped into table and page *lock queue shards*, with each shard protected by a dedicated mutex. Previously, the single lock system mutex protected all lock queues, which was a point

of contention on high-concurrency systems. The new sharded implementation permits more granular access to lock queues.

The lock system mutex (`lock_sys->mutex`) was replaced by the following sharded latches:

- A global latch (`lock_sys->latches.global_latch`) consisting of 64 read-write lock objects (`rw_lock_t`). Access to an individual lock queue requires a shared global latch and a latch on the lock queue shard. Operations that require access to all lock queues take an exclusive global latch, which latches all table and page lock queue shards.
- Table shard latches (`lock_sys->latches.table_shards.mutexes`), implemented as an array of 512 mutexes, with each mutex dedicated to one of 512 table lock queue shards.
- Page shard latches (`lock_sys->latches.page_shards.mutexes`), implemented as an array of 512 mutexes, with each mutex dedicated to one of 512 page lock queue shards.

The Performance Schema `wait/synch/mutex/innodb/lock_mutex` instrument for monitoring the single lock system mutex was replaced by instruments for monitoring the new global, table shard, and page shard latches:

- `wait/synch/sxlock/innodb/lock_sys_global_rw_lock`
- `wait/synch/mutex/innodb/lock_sys_table_mutex`
- `wait/synch/mutex/innodb/lock_sys_page_mutex`

(WL #10314)

- **Group Replication:** Group Replication group members can now advertise a list of IP addresses that joining members can use to make connections to them for state transfer during distributed recovery. Previously, the existing member's standard SQL client connection was used for this purpose as well as for client traffic. Advertising distributed recovery endpoints instead gives you improved control of distributed recovery traffic (comprising remote cloning operations and state transfer from the binary log) in your network infrastructure. The list of distributed recovery endpoints for a member is specified using the new `group_replication_advertise_recovery_endpoints` system variable, and the same SSL requirements are applied that would be in place if the SQL client connection was used for distributed recovery. (WL #13767)

- **Group Replication:** You can now specify user credentials for distributed recovery on the `START GROUP_REPLICATION` statement using the `USER`, `PASSWORD`, and `DEFAULT_AUTH` options. These credentials are used for distributed recovery on the `group_replication_recovery` channel. When you specify user credentials on `START GROUP_REPLICATION`, the credentials are saved in memory only, and are removed by a `STOP GROUP_REPLICATION` statement or server shutdown. These credentials can replace user credentials set using a `CHANGE MASTER TO` statement, which are stored in the replication metadata repositories, and can therefore help to secure the Group Replication servers against unauthorized access.

  The new method of providing user credentials is not compatible with starting Group Replication automatically on server start. If user credentials have previously been set using a `CHANGE MASTER TO` statement, credentials that you specify on `START GROUP_REPLICATION` take precedence over these. However, the credentials from the replication metadata repositories are used if `START GROUP_REPLICATION` is specified without user credentials, which happens on automatic starts if the `group_replication_start_on_boot` system variable is set to `ON` (including after a remote cloning operation for distributed recovery). To gain the security benefits of specifying user credentials on `START GROUP_REPLICATION`, ensure that `group_replication_start_on_boot` is set to `OFF` (the default

is `ON`), and use a `CHANGE MASTER TO` statement to clear any user credentials previously set for the `group_replication_recovery` channel. (WL #13768)

- **Group Replication:** The minimum setting for the maximum size for the XCom message cache in Group Replication, specified by the `group_replication_message_cache_size` system variable, has been reduced from approximately 1 GB to 134217728 bytes, or approximately 128 MB. Note that this size limit applies only to the data stored in the cache, and the cache structures require an additional 50 MB of memory. The same cache size limit should be set on all group members. The default XCom message cache size of 1 GB, which was formerly also the minimum setting, is unchanged.

  The smaller message cache size is provided to enable deployment on a host that has a restricted amount of available memory and good network connectivity. Having a very low `group_replication_message_cache_size` setting is not recommended if the host is on an unstable network, because a smaller message cache makes it harder for group members to reconnect after a transient loss of connectivity. If some messages that were exchanged during a member's temporary absence have been deleted from the other members' XCom message caches because their maximum size limit was reached, the member cannot reconnect using the message cache. It must leave the group and rejoin in order to retrieve the transactions through distributed recovery, which is a slower process than using the message cache, although the member still can rejoin in this way without operator intervention.

  Note that from MySQL 8.0.21, by default an expel timeout of 5 seconds is added before a member is expelled from the group (specified by the `group_replication_member_expel_timeout` system variable). With this default setting the XCom message cache therefore now needs to store the messages exchanged by the group in a 10-second period (the expel timeout plus the initial 5-second detection period), rather than in a 5-second period as previously (the initial 5-second detection period only). (WL #13979)

- **Group Replication:** `group_replication_member_expel_timeout` specifies the period of time in seconds that a Group Replication group member waits after creating a suspicion, before expelling from the group the member suspected of having failed. The initial 5-second detection period before a suspicion is created does not count as part of this time.

  Previously, the waiting period specified by `group_replication_member_expel_timeout` defaulted to 0, meaning that a suspected member was liable for expulsion immediately after the 5-second detection period ended. Following user feedback, the waiting period now defaults to 5 seconds, giving a member that loses touch with the group 10 seconds in total to reconnect itself to the group. If the member does reconnect in this time, it can recover missed messages from the XCom message cache and return to `ONLINE` state automatically, rather than being expelled from the group and needing the auto-rejoin procedure or manual operator intervention to rejoin.

  If you previously tuned the size of the XCom message cache with reference to the expected volume of messages in the previous default time before a member was expelled (the 5-second detection period only), increase your `group_replication_message_cache_size` setting to account for the new expel timeout, which doubles the default time to 10 seconds. With the new default expel timeout you might start to see warning messages from GCS on active group members, stating that a message that is likely to be needed for recovery by a member that is currently unreachable has been removed from the message cache. This message shows that a member has had a need to use the message cache to reconnect, and that the cache size might not be sufficient to support the current waiting period before a member is expelled. (WL #13773)

- **Group Replication:** The Group Replication auto-rejoin feature is now activated by default. The `group_replication_autorejoin_tries` system variable, which is available from MySQL 8.0.16, makes a member that has been expelled or reached its unreachable majority timeout try to rejoin the group automatically. This system variable, which originally defaulted to 0 so auto-rejoin was not

activated, now defaults to 3, meaning that a member makes three attempts to rejoin the group in the event of its expulsion or unreachable majority timeout. Between each attempt the member waits for 5 minutes. If the specified number of tries is exhausted without the member rejoining or being stopped, the member proceeds to the action specified by the `group_replication_exit_state_action` system variable.

The auto-rejoin feature minimizes the need for manual intervention to bring a member back into the group, especially where transient network issues are fairly common. During and between auto-rejoin attempts, a member remains in super read only mode and does not accept writes. However, reads can still be made on the member, with an increasing likelihood of stale reads over time. If you want to intervene to take the member offline, the member can be stopped manually at any time by using a `STOP GROUP_REPLICATION` statement or shutting down the server. If you cannot tolerate the possibility of stale reads for any period of time, set the `group_replication_autorejoin_tries` system variable to 0, in which case operator intervention is required whenever a member is expelled from the group or reaches its unreachable majority timeout. (WL #13706)

- Previously, the `--disabled-storage-engines` option did not ignore spaces around storage engines listed in the option value. Spaces around engine names are now ignored. (Bug #31373361, Bug #99632)
- The new `HANDLE_FATAL_SIGNALS CMake` option enables configuring whether Address Sanitizer and Undefined Behavior Sanitizer builds use the sanitizer runtime library to handle fatal signals rather than a MySQL-internal function. The option default is `ON` for non-sanitizer builds, `OFF` for sanitizer builds. If the option is `OFF`, the default action is used for SIGBUS, SIGILL and SIGSEGV, rather than the internal function. (Bug #31068443)
- Using a column that is repeated twice or more in `GROUP BY` (through an alias), combined with `ROLLUP`, had behavior differing from MySQL 5.7. Example:

```
SELECT a, b AS a, COUNT(*) FROM t1 GROUP BY a, b WITH ROLLUP;
```

  Behavior of such queries has been changed to better match MySQL 5.7. They should be avoided, however, because behavior may change again in the future or such queries may become illegal. (Bug #30921780, Bug #98663)
- `comp_err` provides better error messages for certain input file issues. Thanks to Facebook for the contribution. (Bug #30810629, Bug #98390)
- MySQL Server Docker containers now support server restart within a client session (which happens, for example, when the RESTART statement is executed by a client or during the configuration of an InnoDB Cluster instance). To enable this important feature, containers should be started with the `docker run` option `--restart` set to the value `on-failure`. See Starting a MySQL Server Instance for details. (Bug #30750730)
- `EXPLAIN ANALYZE` now supports the `FORMAT` option. Currently, `TREE` is the only supported format. (Bug #30315224)
- `ALTER INSTANCE ROTATE INNODB MASTER KEY` is no longer permitted when `read_only` or `super_read_only` are enabled. (Bug #30274240)
- On storage engines that support atomic DDL, the `CREATE TABLE ... SELECT` statement is now logged as one transaction in the binary log when row-based replication is in use. Previously, it was logged as two transactions, one to create the table, and the other to insert data. With this change, `CREATE TABLE ... SELECT` statements are now safe for row-based replication and permitted for use with GTID-based replication. For more information, see Atomic Data Definition Statement Support. (Bug #11756034, Bug #47899, WL #13355)
- `LOAD XML` now supports `CDATA` sections in the XML file to be imported. (Bug #98199, Bug #30753708)

- X Plugin's `mysqlx_bind_address` system variable now accepts multiple IP addresses like MySQL Server's `bind_address` system variable does, enabling X Plugin to listen for TCP/IP connections on multiple network sockets.

  An important difference in behavior is that for MySQL Server, any error in the list of addresses prevents the server from starting, but X Plugin (which is not a mandatory plugin) does not do this. With X Plugin, if one of the listed addresses cannot be parsed or if X Plugin cannot bind to it, the address is skipped, an error message is logged, and X Plugin attempts to bind to each of the remaining addresses. X Plugin's `Mysqlx_address` status variable displays only those addresses from the list for which the bind succeeded. If none of the listed addresses results in a successful bind, X Plugin logs an error message stating that X Protocol cannot be used. (WL #12715)

- `ENGINE_ATTRIBUTE` and `SECONDARY_ENGINE_ATTRIBUTE` options were added to `CREATE TABLE`, `ALTER TABLE`, and `CREATE INDEX` syntax. The `ENGINE_ATTRIBUTE` option was also added to `CREATE TABLESPACE` and `ALTER TABLESPACE` syntax. The new options, which permit defining storage engine attributes for tables, columns, indexes, and tablespaces, are reserved for future use.

  The following `INFORMATION_SCHEMA` tables were added for querying storage engine attributes for tables, columns, indexes, and tablespaces. Values are stored in the data dictionary. The tables are reserved for future use.

  - `INFORMATION_SCHEMA.TABLES_EXTENSIONS`
  - `INFORMATION_SCHEMA.COLUMNS_EXTENSIONS`
  - `INFORMATION_SCHEMA.TABLE_CONSTRAINTS_EXTENSIONS`
  - `INFORMATION_SCHEMA.TABLESPACES_EXTENSIONS`

  (WL #13341)

- The default logging level for MySQL Server omits informational log messages, which previously included some significant lifecycle events for Group Replication that were non-error situations, such as a group membership change. Messages about significant events for a replication group have now been reclassified as system messages, so they always appear in the server error log regardless of the server logging level. Operators can therefore review a complete history of the server's membership in a replication group. In addition, socket bind errors on the group communication layer have been reclassified from information to error messages. (WL #13769)

## Bugs Fixed

- **InnoDB:** A `GROUP BY` operation on a `JSON` array column caused failures in an UBSan build of MySQL due to incorrect type casting. (Bug #31451475)

- **InnoDB:** Several `InnoDB` error log messages were defined without symbolic values. (Bug #31401028)

- **InnoDB:** The file segment for a single page write was not released after a data file write failure associated with a doublewrite flush and sync operation. (Bug #31370227)

- **InnoDB:** Code that was accessed when undo tablespace truncation used the same space ID before and after a truncate operation was removed. That scenario no longer occurs. The truncated undo tablespace is replaced by a new undo tablespace datafile with a different space ID. (Bug #31354435)

- **InnoDB:** The range of reserved space IDs for undo tablespaces was increased from 512 per undo tablespace to 400000. (Bug #31340834)

- **InnoDB:** An error that occurred while inserting a log into the `ddl_log` table was not returned making it appear as though the operation was successful, and a transaction was not registered while performing a tablespace encryption operation. (Bug #31236217)
- **InnoDB:** The `lob::purge()` function did not free LOBs correctly for an undo log record type (`TRX_UNDO_UPD_DEL_REC`) that is generated when an insert operation modifies a delete-marked record. (Bug #31222046, Bug #99339)
- **InnoDB:** A shutdown error occurred following an attempt to rebuild a discarded partition. (Bug #31215415)
- **InnoDB:** The internal `get_real_path()` function, responsible for retrieving directory or a file path, was modified to strip trailing separators before determining if a path is a file or directory. Additionally, if a path does not exist or cannot be identified as a file or subdirectory, the function now assumes the path is a file if the basename has a three letter suffix. (Bug #31215160)
- **InnoDB:** Tablespace-related error messages were revised. (Bug #31205520, Bug #31205441)
- **InnoDB:** To avoid potential compilation issues, `__attribute__((const))` and `__attribute__((pure))` attributes were removed from internal `InnoDB` functions. (Bug #31153123)
- **InnoDB:** The parallel read thread limit was not observed when spawning read threads for histogram sampling, causing an assertion failure. (Bug #31151218)
- **InnoDB:** The transaction read view was not checked when sampling records for generation of histogram statistics. (Bug #31151077)
- **InnoDB:** An I/O completion routine was not able acquire an LRU list mutex due to a latch held by another thread. (Bug #31128739)
- **InnoDB:** An attachable transaction thread requested an `InnoDB` ticket that was already reserved by the main thread, causing a deadlock. Additionally, the server failed to respond to `KILL` statements in this deadlock scenario. (Bug #31090777)
- **InnoDB:** The `INNODB_METRICS` table `AVG_COUNT_RESET` value for a counter defined as a module owner reported NULL. The `METRIC_AVG_VALUE_RESET` field was incorrectly marked as NULL. Thanks to Fungo Wang for the contribution. (Bug #31084706, Bug #98990)
- **InnoDB:** At startup, following an unexpected stoppage during an undo tablespace truncate operation, some rollback segment header pages were found to be corrupted. Encryption of rollback segment header pages was initiated while the header pages were being written, resulting in some header pages not being encrypted, as expected. (Bug #31045160)
- **InnoDB:** Various aspects of the lock system (`lock_sys`) code were refactored, and issues with `lock_sys` `lock_rec_block_validate()` and `lock_test_prdt_pacge_lock()` functions were fixed. The `lock_rec_block_validate()` function called another function repeatedly, which could result in locks not being validated under certain circumstances. The implementation also had a potential quadratic time complexity. The `lock_test_prdt_page_lock()` function did not iterate over all locks as intended. (Bug #31001732)
- **InnoDB:** Use of memory-mapped files after exceeding the `temptable_max_ram` threshold caused a performance degradation. (Bug #30952983, Bug #98739)
- **InnoDB:** In debug mode, a `DROP TABLE` operation on a table with an incorrectly defined `COMPRESSION` clause caused a failure. `InnoDB` did not return an error to the caller for proper handling. (Bug #30899683, Bug #98593)
- **InnoDB:** Purge thread activity was excessive when the history list length approached zero, wasting CPU resource and causing mutex contention. (Bug #30875956)

- **InnoDB:** A regression introduced in MySQL 8.0.18 affected `INFORMATION_SCHEMA.INNODB_COLUMNS` query performance. Schema and table data dictionary objects were fetched repeatedly to retrieve partition column information. (Bug #30837086, Bug #98449)

  References: This issue is a regression of: Bug #93033, Bug #28869903.

- **InnoDB:** An `ALTER TABLE ... IMPORT TABLESPACE` operation with a `.cfg` file failed with an "Incorrect key file for table" error. The `row_import::m_flags` member was not initialized. (Bug #30830441)

- **InnoDB:** A `DROP TABLE` operation performed after discarding a partition did not remove the associated data files, and `DROP DATABASE` failed with an error indicating that the database directory could not be removed. Upgrade from MySQL 5.7 to MySQL 8.0 also failed if a partitioned table with a discarded partition was present. The `DISCARD` attribute was applied to the table object instead of the partition object in the data dictionary, which made it appear that all partitions were discarded. (Bug #30818917)

- **InnoDB:** The server failed intermittently with an "ibuf cursor restoration fails" error. (Bug #30770380, Bug #91033)

- **InnoDB:** An `ALTER TABLE` operation that copied data from one table to another returned an "Out of range value for column" error. The counter that tracks the number of `AUTO_INCREMENT` rows required for a multi-row insert operation was not always set back to zero after a bulk insert operation. (Bug #30765952, Bug #98211)

- **InnoDB:** The internal TempTable `records_in_range()` handler function contained a `DBUG_ABORT()` call that caused assertion failures in debug builds, and empty result sets in regular builds for some queries. (Bug #30716037)

- **InnoDB:** The `btr_cur_pessimistic_update()` function failed to handle a cursor position change caused by a `lob::purge()` call. (Bug #30712878)

- **InnoDB:** A type conversion failure during a `DELETE IGNORE` operation caused an assertion failure. A `JSON` value was not converted to the expected value. (Bug #30664660)

- **InnoDB:** A purge operation encountered a null LOB reference, causing an assertion failure. (Bug #30658887)

- **InnoDB:** Chunk size was not calculated correctly when deallocating memory from the `TempTable` storage engine, causing a regression in `SELECT DISTINCT` query performance. (Bug #30562964)

- **InnoDB:** A segmentation fault occurred in the TempTable storage engine while using the thread pool plugin. TempTable thread-local variables were not compatible with the use of different threads for statements issued by a single client connection. Use of thread local variables also lead to excessive memory consumption due to the memory used by thread-local variables remaining allocated for the life of the thread. To address these issues, thread-local variables were replaced by a caching mechanism. (Bug #30050452, Bug #31116036, Bug #99136)

- **InnoDB:** A fatal "page still fixed or dirty" error occurred during shutdown. (Bug #29759555, Bug #95285)

  References: This issue is a regression of: Bug #29207450.

- **Partitioning:** A query against a partitioned table, which used an `ORDER BY`, returned unordered results under the following conditions:

  - The table had a composite index with a prefix on one of the columns.

  - The query's `WHERE` clause contained an equality condition on the prefixed column.

- The column with the prefix was the leftmost column in the index.
- The column used in the `ORDER BY` was the rightmost column in the index.
- The index was used for handling the `ORDER BY`.

Our thanks to Quanan Han for the suggestion. (Bug #84070, Bug #25207522)

- **Replication:** If a group's consistency level (set by the `group_replication_consistency` system variable) was set to `BEFORE` or `BEFORE_AND_AFTER`, it was possible for a deadlock to occur in the event of a primary failover. The primary failover is now registered differently to avoid this situation. (Bug #31175066, Bug #98643)

- **Replication:** When a replication source server shuts down and restarts, its `MEMORY` tables become empty. To replicate this effect to replicas, the first time that the source uses a given `MEMORY` table after startup, it notifies replicas that the table must be emptied by writing a `DELETE` statement for that table to the binary log. Previously, the generated `DELETE` statement was written to the binary log statement cache for the current session, which could result in it being logged together with other statements under the same GTID, or logged without `BEGIN` and `COMMIT` statements. Also, in some situations, the generated `DELETE` statement could consume the GTID intended for the transaction that triggered it. The generated `DELETE` statement is now logged with accompanying `BEGIN` and `COMMIT` statements, and the resulting transaction is flushed to the binary log immediately after it is written to the statement cache, so that it always receives its own GTID and is kept separate from other transactions. (Bug #30527929, Bug #25681518, Bug #77729)

- **Replication:** Following a patch in MySQL 8.0.14, if a function call contained operations on temporary tables, it could be written to the binary log in statement format when `binlog_format = MIXED` was set. This led to `CREATE TEMPORARY TABLE` statements being incorrectly written to the binary log if they contained a function call. Following further analysis, operations on temporary tables in stored functions and triggers are now marked as unsafe for binary logging in statement format, as they have a strong chance of causing issues with replication. When `binlog_format = MIXED` is set, these operations are now logged in row format. (Bug #30395151, Bug #30320009)

- **Replication:** A fix made in MySQL 8.0.14 and MySQL 5.7.25 for a deadlock scenario involving the system variables `binlog_transaction_dependency_tracking` and `binlog_transaction_dependency_history_size` had the side effect of leaving the writeset history used for transaction dependency tracking unprotected from concurrent update. The writeset history and tracking mode are now locked correctly whenever they are accessed. (Bug #29719364, Bug #95181)

  References: See also: Bug #28511326, Bug #91941.

- **Replication:** If a `CHANGE MASTER TO` statement was issued with `MASTER_USER` specified as empty (`MASTER_USER=''`), the statement succeeded and cleared any previously specified user name in the replication metadata repositories. However, if information was subsequently read from the repositories, for example during an automatic restart of a Group Replication channel, a default user name could be substituted for the channel. This issue has now been fixed, so from MySQL 8.0.21, it is a valid approach to set an empty `MASTER_USER` user name if you always provide user credentials using the `START SLAVE` statement or `START GROUP_REPLICATION` statement that starts the replication channel. This approach means that the replication channel always needs operator intervention to restart, but the user credentials are not recorded in the replication metadata repositories.

  The documentation for the `CHANGE MASTER TO` statement has also been corrected to clarify that it is possible to specify `MASTER_USER=''`, and the resulting error occurs only if you attempt to start the replication channel with the empty credentials. (Bug #27357189)

- **Group Replication:** A global value that is set for the `group_replication_consistency` system variable, which controls all user connections, is applied on Group Replication's internal connections to MySQL Server modules using the SQL API, which are handled in a similar way to user connections. This could sometimes lead to Group Replication reporting the use of `group_replication_consistency` as an error, for example when checking the clone plugin status during distributed recovery. Such internal connections using the SQL API now use consistency level `EVENTUAL`, which matches the behavior before `group_replication_consistency` was available, and does not cause an error message. (Bug #31303354, Bug #99345)

- **Group Replication:** On Windows, Group Replication's use of the Windows API function SleepConditionVariableCS to wait for new write events caused noticeably high CPU usage by this function after Group Replication had been running for two days or more, which could be corrected by restarting the MySQL server instance, but then increased again over time as before. This was caused by the use of two clock functions to calculate the timeout after which the SleepConditionVariableCS function was called, which drifted relative to each other over time, making the timeout progressively shorter and the calls to the function more frequent. The issue has been corrected on Windows by using the current time from a single clock to calculate the timeout. (Bug #31117930)

- **Group Replication:** If Group Replication was stopped while distributed recovery was in progress, memory issues could result from an attempt to access the record of the member that was selected as the donor. This record is now kept locally with the distributed recovery state. (Bug #31069563)

- **Group Replication:** When distributed recovery for Group Replication involves a remote cloning operation, the flag set on the server to indicate this remains set until the server instance is restarted. Previously, if Group Replication was stopped and restarted on the server, that flag caused Group Replication to purge the relay log files for the `group_replication_applier` channel, as is required on starting after a remote cloning operation to ensure that there is no mismatch with the cloned data tables. If there were any unapplied transactions in the purged relay log files, the member could not subsequently be used to bootstrap a group, although it could successfully join a group by retrieving the transactions from another member. Group Replication now ignores the flag on its second or subsequent starts, and only purges the relay log files the first time it is started after a remote cloning operation. (Bug #31059680)

- **Group Replication:** To avoid the possibility of data inconsistency, Group Replication blocks a new incarnation of the same server (with the same address but a new identifier) from joining the group while its old incarnation is still listed as a member. Previously, the Group Replication Group Communication System (GCS) treated the connection to the old incarnation of a server as active while it was attempting to send messages to the server, and only recognized that the connection was inactive when the socket returned an error, which might take a significant amount of time. During that period, the new incarnation of the server was unable to join the group because the existing members did not connect to it, as they were still waiting on the connection to the old incarnation. Now, GCS only treats a connection to a server as active while messages can be sent to it successfully. If the socket is no longer writeable, the server connection is treated as inactive and is proactively closed. The connection close triggers the group member to attempt reconnection to that server address, upon which a connection is established to the new incarnation of the server, enabling the new incarnation to join the group. (Bug #30770577)

- **Group Replication:** Group Replication did not broadcast a notification when switching from single-primary mode to multi-primary mode. The change is now notified for use in routing. (Bug #30738896)

- **Group Replication:** Setting the `group_replication_force_members` system variable to force a specified membership for a group could fail if another member had already requested the expulsion of the member that was driving the `group_replication_force_members` operation. The operation to implement the configuration specified by the `group_replication_force_members` system variable forced any pending group reconfigurations to take place first. If one of those successfully expelled the member where the system variable had been set, because the expel timeout that was set on the member had expired, the operation timed out and failed to complete. To avoid this situation,

Group Replication now proceeds directly to implementing the new configuration specified by the `group_replication_force_members` system variable, and ignores any other pending group reconfigurations. (Bug #29820966)

- **Group Replication:** Group Replication tracking of connections to other group members only took into account the incoming connections, not the outgoing connections. This meant if the outgoing connection from member A to member B was broken, for example by a firewall configuration issue, but the incoming connection from member B to member A was intact, member A would display member B's status as `ONLINE`, although member A's messages were not reaching member B. Member B would display member A's status as `UNREACHABLE`. Now, if a group member starts to receive pings from another group member to which it has an active connection (in this case, if member A received pings from member B), this is treated as an indicator of an issue with the connection. If sufficient pings are received, the connection is shut down by the recipient of the pings (in this case, member A), so that the status of the connection is consistent for both members. (Bug #25660161, Bug #84796)

- **JSON:** When the expression and path passed to `JSON_TABLE()` yielded a JSON null, the function raised an error instead of returning SQL `NULL` as required. (Bug #31345503)

- **JSON:** In MySQL 5.7, and in MySQL 8.0 prior to 8.0.17, the server attempted to convert JSON boolean values to their SQL counterparts when testing them directly with `IS TRUE`, as shown here:

```
mysql> CREATE TABLE test (id INT, col JSON);

mysql> INSERT INTO test VALUES (1, '{"val":true}'), (2, '{"val":false}');

mysql> SELECT id, col, col->"$.val" FROM test WHERE col->"$.val" IS TRUE;
+------+--------------+--------------+
| id   | col          | col->"$.val" |
+------+--------------+--------------+
|    1 | {"val": true} | true         |
+------+--------------+--------------+
```

As the result of work done in MySQL 8.0.17 to ensure that all predicates in SQL conditions are complete (that is, a condition of the form `WHERE value` is rewritten as `WHERE value <> 0`), and that a `NOT IN` or `NOT EXISTS` condition in a `WHERE` or `ON` clause is converted to an antijoin, evaluation of a JSON value in an SQL boolean context performs an implicit comparison against JSON integer 0. This means that the query shown previously returns the following result in MySQL 8.0.17 and later:

```
mysql> SELECT id, col, col->"$.val" FROM test WHERE col->"$.val" IS TRUE;
+------+---------------+--------------+
| id   | col           | col->"$.val" |
+------+---------------+--------------+
|    1 | {"val": true}  | true         |
|    2 | {"val": false} | false        |
+------+---------------+--------------+
```

In such cases, the server also now provides a warning: `Evaluating a JSON value in SQL boolean context does an implicit comparison against JSON integer 0; if this is not what you want, consider converting JSON to a SQL numeric type with JSON_VALUE RETURNING`. Thus, the query can now be rewritten using `JSON_VALUE()` as shown here:

```
mysql> SELECT id, col, col->"$.val" FROM test
    ->     WHERE JSON_VALUE(col, "$.val" RETURNING UNSIGNED) IS TRUE;
+------+--------------+--------------+
| id   | col          | col->"$.val" |
+------+--------------+--------------+
|    1 | {"val": true} | true         |
+------+--------------+--------------+
```

(Bug #31168181)

- **JSON:** A `GROUP BY` query against a table having a multi-valued index was not always handled correctly by the server. (Bug #31152942)
- If `log_error_services` was persisted, in some cases it could take effect at the wrong time during startup. (Bug #31464539)
- `SHOW CREATE USER` after certain manual grant table modifications could cause a server exit. (Bug #31462844)
- Some in-memory updates of partial revokes could produce incorrect privileges. (Bug #31430086)
- If `log_error_verbosity` was set using `SET PERSIST`, it did not take effect early enough during server startup to affect `InnoDB` initialization. (Bug #31410674)
- The parser incorrectly raised an assertion before rejecting subqueries in generated column expressions. (Bug #31396191)
- This release makes the following two micro-optimizations for degenerate hash joins (that is, those with no join conditions):

  1. For a degenerate hash antijoin or semijoin, add `LIMIT 1` when building the hash table, since having more rows than this cannot change the result.
  2. For a degenerate hash antijoin with a nonempty hash table, avoid scanning the outer side.

  Together, these changes handle a performance regression whereby a rewrite to a hash antijoin caused a `NOT EXISTS` query which was not rewritten to be executed by the optimizer and be replaced with “zero rows found”. To handle the case in which a nested loop is used instead, a non-correlated subquery inside `NOT EXISTS` is no longer transformed to an antijoin.

  This fix also applies to subqueries using *`constant`* `NOT IN (`*`non_correlated_subquery`*`)`. (Bug #31376809)
- Configuring with `-DWITH_EDITLINE=system` resulted in compilation failures for older library versions. (Bug #31366715)
- The upgrade of the bundled `libedit` library in the previous MySQL distribution caused a problem for builds using that library such that **CTRL**+**C** (SIGINT) in the `mysql` client required a following Enter to take effect in some circumstances. (Bug #31360025)
- Columns declared with both `AUTO_INCREMENT` and `DEFAULT` value expressions are a nonpermitted combination, but `ALTER TABLE` failed to produce an error for `SET DEFAULT (`*`expr`*`)` operations on `AUTO_INCREMENT` columns. (Bug #31331454)
- It was possible to set the `protocol_compression_algorithms` system variable to the empty string. This is no longer permitted. (Bug #31326231)
- A lookup function used internally in the MySQL server returns integer -1 when the argument is ambiguous; this resulted in undefined behavior when this value was converted to an unsigned value prior to use as an argument in subsequent calculations. Now when the function returns -1, this is handled as an error and the value is not used further. (Bug #31326120)
- Negation of a signed value in certain cases led to undefined behavior; to prevent this from occurring, the value to be negated is now treated as unsigned. (Bug #31325602)
- The `WEIGHT_STRING()` function did not always return the correct result for an integer argument. (Bug #31321257)

  References: This issue is a regression of: Bug #30776132.

- Assigning `CONCAT('')` or `CONCAT_WS('')` to a variable set the variable to `NULL`, not the empty string. (Bug #31320716, Bug #99485, Bug #31413167, Bug #99722)
- Corrected problems where under some circumstances privilege restrictions could be ignored. (Bug #31306814, Bug #31315692)
- Certain `SELECT` statement privileges to lock rows were not checked properly and could block other users incorrectly. (Bug #31293065)
- When performing a filesort, an internal function could sometimes return `NULL` on failure, even if the subselect being sorted was not nullable. (Bug #31281602)
- Statement rewriting for the binary log was inefficient on Windows. (Bug #31260698)

  References: This issue is a regression of: Bug #30654405.
- An inconsistency in representing anonymous users in memory could cause issues while performing privilege-granting operations. (Bug #31246179)
- If the administrative connection interface was enabled, a race condition could lead to problems accepting Unix socket file connections on the main connection interface. (Bug #31241872)
- When a role was granted with `WITH ADMIN OPTION`, the grantee was able to manage the role only after activating it. (Bug #31237368)
- Invalid rows in the `default_roles` or `role_edges` system tables could cause server misbehavior. (Bug #31217385)
- Component deinitialization failure at runtime could lead to repeated messages written to the error log at shutdown. (Bug #31217037)
- The prohibition on granting roles to anonymous users was incompletely enforced. (Bug #31215017)
- A privilege-escalation issue was corrected. (Bug #31210226)
- The `keyring_hashicorp` keyring plugin did not perform sufficient validity checking on the values of its configuration parameters. (Bug #31205363)
- The `keyring_hashicorp` keyring plugin did not permit binary log encryption to be enabled (by setting the `binlog_encryption` system variable). (Bug #31204841)
- The `keyring_hashicorp` keyring plugin did not permit an encryption password to be set by the `audit_log` plugin. (Bug #31197670)
- Some queries using `REGEXP_SUBSTR()` with an `ORDER BY` clause were not handled correctly by the server. (Bug #31184858)
- Some instances where pointer arithmetic was applied to `nullptr` were corrected. (Bug #31172750)
- If the available file descriptors were exhausted, `mysql_real_connect()` caused the client to exit. (Bug #31151052)
- Using the `killall` command to initiate a `mysqld` shutdown resulted in no message being logged to indicate the start of shutdown. This has been corrected. (Bug #31121907)
- Calling `mysql_real_connect_nonblocking()` with an invalid host could cause the client to exit upon calling `mysql_close()`. (Bug #31104389, Bug #99112)
- For Debian packages, Python 2 dependencies that could cause installation failures were removed. (Bug #31099324)

- A potential memory leak in `lf_hash_insert()` was fixed. (Bug #31090258, Bug #99078)
- Within the LDAP SASL authentication plugins, multiple calls to `sasl_client_done()` could cause undefined behavior in some cases. (Bug #31088206)
- With the thread pool plugin enabled, high concurrency conditions could cause loss of client context resulting in a server exit. (Bug #31085322)
- For result sets processed using `mysql_use_result()`, `mysql_fetch_row_nonblocking()` did not increment the number of rows, so that after all the rows were fetched, `mysql_num_rows()` returned an incorrect number of rows. (Bug #31082201, Bug #99073)
- Removed an unneeded optimization for `EXISTS()` that was never actually evaluated. (Bug #31069510)
- For a server started with the `--skip-grant-tables` option, enabling the `partial_revokes` system variable caused a server exit. (Bug #31066069, Bug #31202963)
- Queries that used a recursive common table expression with an outer reference could return incorrect results. (Bug #31066001, Bug #99025)
- The parser could fail for multibyte character sets with a minimum character length greater than 1 byte. (Bug #31063981)
- In some cases, the `LEAST()` function could return `NULL` for non-nullable input. (Bug #31054254)

  References: This issue is a regression of: Bug #25123839.
- `mysql_real_connect_nonblocking()` blocked if the `MYSQL_OPT_CONNECT_TIMEOUT` option was set. (Bug #31049390, Bug #98980)
- The last call to the `mysql_fetch_row_nonblocking()` C API function to return the null row was setting an error when it should not have. (Bug #31048553, Bug #98947)
- On Windows, the default connection type uses a named pipe. The nonblocking C API, which is intended for TCP/SSL connections, did not take that into account and caused the client to exit. It now produces an error message indicating the problem. (Bug #31047717)
- X Plugin connections that failed to authenticate due to nonexistence of the user modified the global `audit_log_filter_id` system variable. (Bug #31025461)
- `LOAD DATA` did not ignore hidden generated columns when parsing input file rows. (Bug #31024266, Bug #98925)
- Pinbox exhaustion in the metadata locking subsystem could produce a misleading error message. (Bug #31019269, Bug #98911)
- `CREATE TABLE ... SELECT` failed if it included a functional index. (Bug #31017765, Bug #98896)
- For X Protocol connections, checking the global session mutex was improved to eliminate a minor performance degradation as the number of threads increased. (Bug #31000043)
- In certain cases, executing a query containing multiple subqueries could lead to an unplanned shutdown of the server. (Bug #30975826)
- `SHOW CREATE TRIGGER` failed if `FLUSH TABLES WITH READ LOCK` was in effect. (Bug #30964944)
- Excessive access checks were performed on certain of the data dictionary tables that underlie `INFORMATION_SCHEMA` views, resulting in slow `SHOW COLUMNS` performance. These checks were reduced to improve performance.

In addition, several SHOW statements implemented as INFORMATION_SCHEMA queries were found to benefit from enabling the derived_merge flag for the optimizer_switch system variable. Such queries now internally enable that flag temporarily for better performance, regardless of the flag session value. Affected queries are:

```
SHOW SCHEMAS
SHOW TABLES
SHOW TABLE STATUS
SHOW COLUMNS
SHOW KEYS
SHOW EVENTS
SHOW TRIGGERS
SHOW PROCEDURE STATUS
SHOW FUNCTION STATUS
SHOW CHARACTER SET
SHOW COLLATION
```

(Bug #30962261, Bug #98750, Bug #30921214)

- Two otherwise identical queries executed separately returned one row when using a case-sensitive collation and two rows with a case-insensitive collation. When the same two predicates were combined in a single query using AND, two rows were returned when only one row should have been. (Bug #30961924)
- ALTER TABLE on a SET column that had a display width greater than 255 was not done in place, even if otherwise possible. (Bug #30943642, Bug #98523)
- The server checked whether a number in yottabytes was too large to print by comparing the value as a double to ULLONG_MAX, which cannot be represented as a double. This caused the double value immediately above ULLONG_MAX yottabytes to be printed as 0Y, the erroneous conversion being reported by Clang 10. (Bug #30927590)
- Resource group SQL statements such as CREATE RESOURCE GROUP did not work over connections that use X Protocol. (Bug #30900411)
- SHOW GRANTS could display function privileges as procedure privileges. (Bug #30896461, Bug #98570)
- The audit_log plugin mishandled connection events when multiple clients connected simultaneously. (Bug #30893593)
- The LOCK_ORDER tool reported a syntax error for empty dependency graphs. Empty graphs are now permitted.

  The LOCK_ORDER tool could exhibit unexpected behavior due to mishandling thread list maintenance. (Bug #30889192)
- Upgrades from MySQL 5.7 did not grant the REPLICATION_APPLIER privilege to root. (Bug #30783149)
- The gen_range() user-defined function could mishandle its arguments, causing a server exit. (Bug #30763294)
- During UPDATE processing, conversion of an internal in-memory table to InnoDB could result in a key-length error. (Bug #30674616)
- Attempts to grant dynamic privileges (which are always global) at the procedure or function level did not produce an error. (Bug #30628160)

- Table value constructors ignored the `LIMIT` clause. The clause is now taken into account. For example: `VALUES ROW(1), ROW(2), ROW(3) LIMIT 2` outputs 1 and 2. (Bug #30602659)
- It is possible to define a column named `*` (single asterisk character), but `SELECT `*`` was treated identically to `SELECT *`, making it impossible to select only this column in a query; in other words, the asterisk character was expanded to a list of all table columns even when it was surrounded by backticks. (Bug #30528450)
- The `FROM_DAYS()` function could produce results that were out of range (with the year > 9999). (Bug #30455845, Bug #97340)
- For debug builds, altering the `mysql.func` table to `MyISAM` (not a recommended operation in any case) caused a server exit. Now this operation is prohibited. (Bug #30248138, Bug #96692)
- Queries on the `INFORMATION_SCHEMA KEY_COLUMN_USAGE` and `TABLE_CONSTRAINTS` views could be slow due to `UNION` use in their definitions. These were rewritten to move the `UNION` into a `LATERAL` table to enable the optimizer to better use indexes. (Bug #30216864, Bug #30766181, Bug #98238)
- In certain cases, a `LIMIT` clause incorrectly caused the optimizer to estimate that zero rows needed to be read from a table. (Bug #30204811)

  References: This issue is a regression of: Bug #29487181.
- An internal packet-length function returned values of the wrong integer type. (Bug #30139031)
- Calculations by `mysqldump` for the length of `INSERT` statements did not take into account the `_binary` character set introducer used for `VARBINARY` strings. (Bug #29998457, Bug #96053)
- The messages printed to the error log during upgrade of partitioned tables defined with prefix keys did not provide sufficient details. Detailed warnings that indicate the schema, table, column, and prefix length are now printed. (Bug #29942014)

  References: See also: Bug #31100205.
- `mysql_store_result()` could fail to detect invalid data packets. (Bug #29921423)
- An assertion was raised if creating a child table in a foreign key relation caused an engine substitution. (Bug #29899151, Bug #95743)
- `mysqltest` and `mysql-test-run.pl` no longer support the `--sleep` command-line option. `mysqltest` no longer supports the `real_sleep` command. (Bug #29770237)
- The server permitted connections for hosts with names longer than the maximum permitted length (255 characters). (Bug #29704941)
- In a multiple-table `UPDATE` that updated the key of the first table, if a temporary table strategy was used, duplicate entries could be written to the temporary table, followed by occurrence of a `Can't find record` error. (Bug #28716103)
- The server sometimes mistakenly removed a subquery with a `GROUP BY` when optimizing a query, even in some cases when this subquery was used by an outer select. This could occur when the subquery also used an aggregate function. (Bug #28240054)
- Coercibility of the `NAME_CONST()` function was assessed incorrectly. (Bug #26319675)
- When reading rows from a storage engine, errors other than “no more records” could be ignored, leading to problems later. (Bug #20162055)

- When a multi-table update used a temporary table, this was not shown in the output of `EXPLAIN FORMAT=TREE`, even though such use could have an impact on the performance of the `UPDATE` statement for which this was done. (Bug #17978975)
- When performing a filesort for removing duplicates, such as when executing `SELECT DISTINCT`, it may be necessary to perform another sort afterwards to satisfy an `ORDER BY`. In cases where such an `ORDER BY` had been pushed down into the first table of a join, as opposed to the join as a whole, this final sort was not actually performed. (Bug #99687, Bug #31397840)
- Refactoring work done in MySQL 8.0.20 caused single-row buffering for `GROUP BY` of non-nullable columns not to function correctly, not taking into account that such a column could be the inner table for an outer join, and thus would have a `NULL` flag that would need to be copied. In a `GROUP BY` without a temporary table, this would cause the `NULL` flag to come from the next output row instead of the previous one, and the data returned to be inconsistent. (Bug #99398, Bug #31252625)

  References: This issue is a regression of: Bug #30460528.
- A logical error in the constant folding code for the case in which a constant of type `DECIMAL` or `FLOAT` was the left-hand operand and an integer column value was the right-hand operand yielded an incorrect result. (Bug #99145, Bug #31110614)
- A query whose predicate compared `0` with `-0` where at least one of these was a floating-point value returned incorrect results. (Bug #99122, Bug #31102789)
- Reimplemented rollups without using slices. This fixes the following known issues:
  - A repeating column in a `GROUP BY ... WITH ROLLUP` yielded the wrong result; that is, a `GROUP BY` of the form `GROUP BY a, b, a WITH ROLLUP` erroneously produced `NULL` for some of the column names in the result.
  - A `GROUP BY ... WITH ROLLUP` that did not require a temporary table to print the result also produced an erroneous `NULL` in place of at least one of the expected column names in the output.

  (Bug #98768, Bug #99141, Bug #26227613, Bug #29134467, Bug #30967158, Bug #30969045, Bug #31110494)
- `SELECT DISTINCT( HEX( WEIGHT_STRING(varchar_column) ) )` returned a truncated result. (Bug #98592, Bug #30898753)
- Problems with error handling in queries with `MAX()`, `MIN()`, or both, combined with a `GROUP BY` clause, meant that such a query continued executing until it went through all possible iterations even when an error should have caused it to terminate immediately. (Bug #98242, Bug #30769515)
- After refactoring the type propagation code for `LEAST()`, `GREATEST()`, and other functions, as well as `UNION`, an adjustment of the result type for data types like `ENUM` also replaced the calculated integer data type with a type that could not accommodate both signed and unsigned values. (Bug #95148, Bug #29698617)

  References: This issue is a regression of: Bug #83895, Bug #25123839.

# Changes in MySQL 8.0.20 (2020-04-27)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

- Account Management Notes
- Compilation Notes

- Deprecation and Removal Notes
- JSON Notes
- Logging Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Pluggable Authentication
- SQL Syntax Notes
- Test Suite Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Account Management Notes

- Previously, for a user to access definitions of routines the user did not define, the user was required to have the global `SELECT` privilege, which is very broad. The new `SHOW_ROUTINE` privilege may be granted instead as a privilege with a more restricted scope that permits access to routine definitions. (That is, an administrator can rescind global `SELECT` from users that do not otherwise require it and grant `SHOW_ROUTINE` instead.) This enables an account to back up stored routines without requiring a broad privilege.

  `SHOW_ROUTINE` provides access to:

  - The contents of the `INFORMATION_SCHEMA.ROUTINES` table.
  - The `SHOW CREATE FUNCTION` and `SHOW CREATE PROCEDURE` statements.
  - The `SHOW FUNCTION CODE` and `SHOW PROCEDURE CODE` statements.
  - The `SHOW FUNCTION STATUS` and `SHOW PROCEDURE STATUS` statements.

  For upgrades from an older version of MySQL, any user who has the global `SELECT` privilege is granted `SHOW_ROUTINE`, if there is not already some user who has `SHOW_ROUTINE`. (WL #9049)

## Compilation Notes

- **Solaris:** Clang and GCC now can be used for compiling MySQL on Solaris, although both are experimental and cannot currently be used for production code. (Bug #30562248)
- On EL7 and EL8, `CMake` configuration was adjusted to look for GCC 9 before GCC 8. Because `libmysqlclient` ships with MySQL distributions, client applications built against `libmysqlclient` on those platforms are affected and may need to be recompiled. (Bug #30722756)
- On Windows, the `CMake` compiler-version check for Visual Studio was updated to indicate that Visual Studio 2019 is the currently supported version. (The version check can be bypassed by running `CMake` with `-DFORCE_UNSUPPORTED_COMPILER=1`.) (Bug #30688403)

## Deprecation and Removal Notes

- **JSON:** Previously, it was possible to specify `ON EMPTY` and `ON ERROR` clauses in either order when invoking the `JSON_TABLE()` function. This runs counter to the SQL standard, which stipulates that when `ON EMPTY` is specified, it must always come before any `ON ERROR` clause. For this reason, specifying `ON ERROR` before `ON EMPTY` is now deprecated, and trying to do so causes the server to issue a warning. Support for the nonstandard syntax will be removed in a future version of MySQL. (WL #13512)

- The `max_length_for_sort_data` system variable is now deprecated due to optimizer changes that make it obsolete and of no effect. (WL #13600)

  References: See also: Bug #30473261.

- The use of `VALUES()` to access new row values in `INSERT ... ON DUPLICATE KEY UPDATE` statements is now deprecated, and is subject to removal in a future MySQL release. Instead, you should use aliases for the new row and its columns as implemented in MySQL 8.0.19 and later.

  For example, the statement shown here uses `VALUES()` to access new row values:

  ```
  INSERT INTO t1 (a,b,c) VALUES (1,2,3),(4,5,6)
    ON DUPLICATE KEY UPDATE c=VALUES(a)+VALUES(b);
  ```

  Henceforth, you should instead use a statement similar to the following, which uses an alias for the new row:

  ```
  INSERT INTO t1 (a,b,c) VALUES (1,2,3),(4,5,6) AS new
    ON DUPLICATE KEY UPDATE c = new.a+new.b;
  ```

  Alternatively, you can employ aliases for both the new row and each of its columns, as shown here:

  ```
  INSERT INTO t1 (a,b,c) VALUES (1,2,3),(4,5,6) AS new(m,n,p)
    ON DUPLICATE KEY UPDATE c = m+n;
  ```

  For more information and examples, see INSERT ... ON DUPLICATE KEY UPDATE Statement. (WL #13325)

## JSON Notes

- The `rapidjson` library included with MySQL has been upgraded to the GitHub snapshot of 16 January 2020. A fix for a compiler error encountered when building from the snapshot on Mac OS X has been added. (Bug #30898701)

## Logging Notes

- Sending a `SIGHUP` signal to the server no longer causes it to write a status report to the error log. Other actions performed by the server in response to `SIGHUP` continue to be done. See Unix Signal Handling in MySQL.

  Similarly, `mysqladmin debug` no longer causes the status report to be written. (Bug #30578923)

- The `log_sink_json` JSON-format error log sink now includes a `ts` (timestamp) in log messages. The value is an integer indicating milliseconds since the epoch (`'1970-01-01 00:00:00'` UTC). See Error Log Output Format. (WL #13786)

## Optimizer Notes

- Hash joins are now used any time a nested block loop would be employed. This means that hash joins can be used for the following types of queries:

- Inner non-equi-joins
- Semijoins
- Antijoins
- Left outer joins
- Right outer joins

This builds on work done for MySQL 8.0.18, and removes a limitation in the implementation such that a hash join could be used only with a query having at least one equi-join condition. In addition, both inner and outer joins (including semijoins and antijoins) can now employ batched key access (BKA), which allocates join buffer memory incrementally so that individual queries need not use up large amounts of resources that they do not actually require for resolution. For more information, see Batched Key Access Joins.

This fix completes the task of replacing the executor used in previous versions of MySQL with the iterator executor, including replacement of the old index subquery engines that governed queries of the form `WHERE value IN (SELECT column FROM table WHERE condition)` for those `IN` queries which have not been converted into semijoins, as well as queries materialized into the same form, which depended on internals from the old executor.

For more information and examples, see Hash Join Optimization. (Bug #30528604, Bug #30473261, Bug #30912972, WL #13377, WL #13476)

- This release implements several new index-level optimizer hints, which function much like existing index hints that employ SQL keywords such as `FORCE INDEX` and `IGNORE INDEX`. These are intended to replace the equivalent index hints, which will be deprecated in a future MySQL release (and eventually removed). The new hints are listed here, along with a brief description of each:

  - `JOIN_INDEX`: Forces MySQL to use the specified index or indexes for any available access method, such as `ref`, `range`, `index_merge`, and so on. This is equivalent to the `FORCE INDEX FOR JOIN` index hint.

    `NO_JOIN_INDEX`: Causes the server to ignore the specified index or indexes for any access method. The equivalent index hint is `IGNORE INDEX FOR JOIN`.

  - `GROUP_INDEX`: Makes the server use the specified index or indexes for index scans for `GROUP BY` operations. Equivalent to `FORCE INDEX FOR GROUP BY`.

    `NO_GROUP_INDEX`: Forces MySQL to ignore the specified index or indexes for index scans for `GROUP BY` operations. The equivalent index hint is `IGNORE INDEX FOR GROUP BY`.

  - `ORDER_INDEX`: Causes MySQL to use the specified index or indexes for sorting rows. It is equivalent to `FORCE INDEX FOR ORDER BY`.

    `NO_ORDER_INDEX`: Keeps the server from using the specified index or indexes for performing row sorts. Equivalent to `IGNORE INDEX FOR ORDER BY`.

  - `INDEX`: Acts as the combination of `JOIN_INDEX`, `GROUP_INDEX`, and `ORDER_INDEX`, forcing the server to use the specified index or indexes for any and all scopes. Equivalent to `FORCE INDEX`.

    `NO_INDEX`: Acts as the combination of `NO_JOIN_INDEX`, `NO_GROUP_INDEX`, and `NO_ORDER_INDEX`; that is, it forces MySQL to ignore the specified index or indexes for any and all scopes. It is equivalent to the index hint `IGNORE INDEX`.

Consider the following query using index hints on a table having the indicated columns and index:

```
SELECT a,b FROM t1 USE INDEX FOR ORDER BY (i_ab) ORDER BY a;
```

Using the index-level optimizer hints introduced in this release, this query can be rewritten as shown here:

```
SELECT /*+ ORDER_INDEX(t1 i_ab) */ a,b FROM t1 ORDER BY a;
```

The new index-level optimizer hints can be used with `SELECT`, `UPDATE`, and `DELETE` statements. (This is unlike index hints using `FORCE INDEX` or `IGNORE INDEX`, which can be used only with `SELECT` and `UPDATE`.) Thus, statements like the following are possible:

```
UPDATE /*+ INDEX(t1 i_ab) */ t1 SET d = 1
    WHERE a = 1 AND b = 2 AND c = 3;

DELETE /*+ INDEX(t1 i_a,i_c) */ FROM t1
    WHERE a = 1 AND b = 2 AND c = 3;
```

Multiple hints can be specified within the same comment, like this:

```
DELETE /*+ INDEX(t1 i_a) JOIN_INDEX(t1 i_c) */
    FROM t1 WHERE a = 1 AND b = 2 AND c = 3;
```

Index-level optimizer hints can be used concurrently with other optimizer hints. When you do so, the index-level hints apply first; the effects of any other optimizer hints are limited to the set of indexes permitted by the index-level hints.

Index-level hints can also be used when creating views, as shown here:

```
CREATE VIEW v1 AS
    SELECT /*+ NO_INDEX(t1 i_a,i_b) */ a FROM t1
    WHERE b IN
        (SELECT /*+ NO_INDEX(t1 i_ab,i_b) */ a FROM t1 WHERE a > 3)
    ORDER BY a;
```

If these index-level optimizer hints are used in the same statement as index hints, the index hints are ignored.

The new index-level optimizer hints are equivalent to `FORCE INDEX` rather than `USE INDEX`; in other words, using one or more of the index-level optimizer hints means that a table scan is used only if there is no way to use one of the named indexes to find rows in the table. To cause MySQL to use the same index or set of indexes as with a given instance of `USE INDEX`, you can use `NO_INDEX`, `NO_JOIN_INDEX`, `NO_GROUP_INDEX`, `NO_ORDER_INDEX`, or some combination of these.

For more information and examples, see Index-Level Optimizer Hints. (WL #13538)

## Packaging Notes

- Binary packages that include `curl` rather than linking to the system `curl` library have been upgraded to use `curl` 7.69.0. (Bug #30866333)
- For RPM packages, the `comp_err` utility has been moved to the `-test` subpackage and marked as a test component. (Bug #30716034)
- The bundled `libedit` library was upgraded to version 3.1. (Bug #28939380, Bug #20770875, Bug #22930525, Bug #22332089, Bug #27433491, Bug #27285445, WL #13534)

- The bundled LZ4 library was upgraded to version 1.9.2. This fixes certain issues such as Bug #30369643 producing a `mysqlpump` runtime error. (WL #13690)

  References: See also: Bug #30369643.

## Performance Schema Notes

- The Performance Schema collected session-related statistics for errors that can occur only globally and not per session. This is no longer done, reducing memory overhead for error instrumentation. Additionally, rows for global errors are no longer included in error summaries reported per thread, account, user, or host. (Bug #30311574)

## Pluggable Authentication

- An LDAP server can be configured to delegate LDAP searches to another LDAP server, a functionality known as LDAP referral. However, enabling LDAP referral can cause searches to fail with LDAP operation errors under certain conditions. To enable the MySQL Enterprise Edition LDAP authentication plugins to avoid referral errors, the new `authentication_ldap_simple_referral` and `authentication_ldap_sasl_referral` system variables are available. These variables enable each plugin to control whether the LDAP server should use referral during MySQL authentication. See LDAP Search Referral. (WL #12888)

- The MySQL Enterprise Edition SASL LDAP authentication plugin now supports GSSAPI/Kerberos as an authentication method for MySQL clients and servers on Linux. This is useful in Linux environments where applications access LDAP using Microsoft Active Directory, which has Kerberos enabled by default. See LDAP Authentication Methods.

  This feature is available for all RPM and DEB packages for Linux, but not for the TAR archive packages. (WL #12888)

## SQL Syntax Notes

- Previously, the `INTO` clause for `SELECT` statements could appear at either of two positions:

  - Before `FROM`:

    ```
    SELECT * INTO OUTFILE 'file_name' FROM table_name;
    ```

  - Before a trailing locking clause:

    ```
    SELECT * FROM table_name INTO OUTFILE 'file_name' FOR UPDATE;
    ```

  `INTO` now can appear in a third position, at the end of `SELECT` statements:

  ```
  SELECT * FROM table_name FOR UPDATE INTO OUTFILE 'file_name';
  ```

  Placing `INTO` at the end is the preferred position. The position before a locking clause is now deprecated and support for it will be removed in a future MySQL version. In other words, `INTO` after `FROM` but not at the end of the `SELECT` produces a warning.

  Additionally, some changes have been made for `UNION` with respect to `INTO`. These `UNION` variants containing `INTO` are syntactically correct and produce the same result:

  ```
  ... UNION SELECT * FROM table_name INTO OUTFILE 'file_name';
  ... UNION (SELECT * FROM table_name) INTO OUTFILE 'file_name';
  ... UNION SELECT * INTO OUTFILE 'file_name' FROM table_name;
  ... UNION (SELECT * INTO OUTFILE 'file_name' FROM table_name);
  ```

However, the last two variants are confusing, as if they collect information from the named table rather than the entire query expression (the `UNION`). Those two `UNION` variants containing `INTO` now are deprecated and support for them will be removed in a future MySQL version. Thus:

- In the trailing query block of a query expression, use of `INTO` before `FROM` produces a warning.
- In a parenthesized trailing block of a query expression, use of `INTO` (regardless of its position relative to `FROM`) produces a warning.

The deprecations apply to all `INTO` forms: `INTO OUTFILE`, `INTO DUMPFILE`, and `INTO var_list`. (WL #13559)

## Test Suite Notes

- The `perfschema.idx_compare_replication_applier_status` test case was updated to store the old value of number of transaction retries and compare it with the new value of number of transaction retries. Thanks to Facebook for the contribution. (Bug #30810627, Bug #98389)

## X Plugin Notes

- If the MySQL Server instance's client connections limit, as specified by the `max_connections` server system variable, was reached while X Plugin was starting up, X Plugin was unable to create a session to get the server configuration, so failed to start. X Plugin now creates an administrative session (using the `mysql_admin_session` service) during startup, which is not subject to the client connections limit. (Bug #30894981)
- When an X Protocol session could not be initialized because there were too many X Protocol connections already, the error code 5011 `Could not open session` was returned. The more relevant error code 1040 `Too many connections` is now returned in this situation. (Bug #30753637)
- An issue with validating JSON references caused an error when creating a collection with a validation schema. (Bug #30733330)
- During shutdown of a MySQL Server instance with X Protocol connections to clients, a race condition in X Plugin could cause invalid client connections to be accepted for processing. Because invalid clients were ignored for client timeout verification during shutdown, these clients blocked shutdown until the timeout set by the `mysqlx_wait_timeout` system variable was reached, which defaults to 8 hours. To prevent this issue, client timeout verification now includes clients that are in an invalid state. (Bug #30702685)
- When connecting to a MySQL 8.0 server, X Plugin set a different collation for the session to that used by the `mysql` client, which could cause issues with queries that depended on the collation. X Plugin now uses the `utf8mb4_0900_ai_ci` collation, which is the default for the `utf8mb4` characterset. (Bug #30516849)
- The worker threads for X Protocol connections were identified as system threads on creation, and assigned to the `SYS_default` resource group. This identification meant they could not be assigned to user resource groups for resource management purposes. They are now identified as user threads and assigned to the `USR_default` resource group. Note that X Protocol does not currently support `CREATE`, `ALTER`, `DROP`, and `SET RESOURCE GROUP` statements, but these statements can operate on X Protocol connection threads using classic MySQL protocol connections. (Bug #30059288)
- X Plugin can now access the MySQL system variables as soon as initialization starts, so the plugin install thread can set up the required connectivity itself rather than starting a separate thread. (Bug #29127302)

## Functionality Added or Changed

- **Important Change:** Previously, including any column of a blob type larger than `TINYBLOB` or `BLOB` as the payload in an ordering operation caused the server to revert to sorting row IDs only, rather than complete rows; this resulted in a second pass to fetch the rows themselves from disk after the sort was completed. Since `JSON` and `GEOMETRY` columns are implemented internally as `LONGBLOB`, this caused the same behavior with these types of columns even though they are almost always much shorter than the 4GB maximum for `LONGBLOB` (or even the 16 MB maximum for `MEDIUMBLOB`). The server now converts columns of these types into packed addons in such cases, just as it does `TINYBLOB` and `BLOB` columns, which in testing showed a significant performance increase. *The handling of `MEDIUMBLOB` and `LONGBLOB` columns in this regard remains unchanged*.

  One effect of this enhancement is that it is now possible for `Out of memory` errors to occur when trying to sort rows containing very large (multi-megabtye) `JSON` or `GEOMETRY` column values if the sort buffers are of insufficient size; this can be compensated for in the usual fashion by increasing the value of the `sort_buffer_size` system variable. (Bug #30400985, Bug #30804356)

- **InnoDB:** The Contention-Aware Transaction Scheduling (CATS) algorithm, which prioritizes transactions that are waiting for locks, was improved. Transaction scheduling weight computation is now performed a separate thread entirely, which improves computation performance and accuracy.

  The First In First Out (FIFO) algorithm, which had also been used for transaction scheduling, was removed. The FIFO algorithm was rendered redundant by CATS algorithm enhancements. Transaction scheduling previously performed by the FIFO algorithm is now performed by the CATS algorithm.

  A `TRX_SCHEDULE_WEIGHT` column was added to the `INFORMATION_SCHEMA.INNODB_TRX` table, which permits querying transaction scheduling weights assigned by the CATS algorithm.

  The following `INNODB_METRICS` counters were added for monitoring code-level transaction scheduling events:

  - `lock_rec_release_attempts`

    The number of attempts to release record locks.

  - `lock_rec_grant_attempts`

    The number of attempts to grant record locks.

  - `lock_schedule_refreshes`

    The number of times the wait-for graph was analyzed to update transaction schedule weights.

  (WL #13486)

- **InnoDB:** The storage area for the doublewrite buffer was moved from the system tablespace to doublewrite files. Moving the doublewrite buffer storage area out of the system tablespace reduces write latency, increases throughput, and provides flexibility with respect to placement of doublewrite buffer pages. The following system variables were introduced for advanced doublewrite buffer configuration:

  - `innodb_doublewrite_dir`

    Defines the doublewrite buffer file directory.

  - `innodb_doublewrite_files`

    Defines the number of doublewrite files.

- `innodb_doublewrite_pages`

  Defines the maximum number of doublewrite pages per thread for a batch write.

- `innodb_doublewrite_batch_size`

  Defines the number of doublewrite pages to write in a batch.

For more information, see Doublewrite Buffer. (WL #5655)

- `EXPLAIN ANALYZE` can now be stopped during execution using `KILL QUERY` or **CTRL-C**. (Bug #30787515)
- `EXPLAIN FORMAT=TREE` now displays `inversion` information for windowing functions. (Bug #30770631)
- `EXPLAIN FORMAT=TREE` output has been improved to provide more information about evaluated window functions, and to match that supplied for regular aggregates. (Bug #30573446, Bug #30582782)
- Configuring with the `-DWITH_LTO=1` `CMake` option now works on macOS. (Bug #30125902)
- You can now enable binary log transaction compression on a MySQL server instance. When binary log transaction compression is enabled, transaction payloads are compressed using the zstd algorithm, and then written to the server's binary log file as a single event (a `Transaction_payload_event`). Compressed transaction payloads remain in a compressed state while they are sent in the replication stream to replicas, other Group Replication group members, or clients such as `mysqlbinlog`. They are not decompressed by receiver threads, and are written to the relay log still in their compressed state. Binary log transaction compression therefore saves storage space both on the originator of the transaction and on the recipient (and for their backups), and saves network bandwidth when the transactions are sent between server instances.

  You can enable binary log transaction compression on a MySQL server instance using the `binlog_transaction_compression` system variable, which defaults to `OFF`. You can also use the `binlog_transaction_compression_level_zstd` system variable to set the level for the zstd algorithm that is used for compression. This value determines the compression effort, from 1 (the lowest effort) to 22 (the highest effort). (WL #3549)

- A new option for the `CHANGE MASTER TO` statement, `REQUIRE_TABLE_PRIMARY_KEY_CHECK`, enables a replication slave to select its own policy for primary key checks. When the option is set to `ON` for a replication channel, the slave always uses the value `ON` for the `sql_require_primary_key` system variable in replication operations, requiring a primary key. When the option is set to `OFF`, the slave always uses the value `OFF` for the `sql_require_primary_key` system variable in replication operations, so that a primary key is never required, even if the master required one. When the `REQUIRE_TABLE_PRIMARY_KEY_CHECK` option is set to `STREAM`, which is the default, the slave uses whatever value is replicated from the master for each transaction.

  - For multisource replication, setting `REQUIRE_TABLE_PRIMARY_KEY_CHECK` to `ON` or `OFF` enables a slave to normalize behavior across the replication channels for different masters, and keep a consistent setting for the `sql_require_primary_key` system variable. Using `ON` safeguards against the accidental loss of primary keys when multiple masters update the same set of tables. Using `OFF` allows masters that can manipulate primary keys to work alongside masters that cannot.

  - When `PRIVILEGE_CHECKS_USER` is set to apply replication privilege checks to the channel, setting `REQUIRE_TABLE_PRIMARY_KEY_CHECK` to `ON` or `OFF` means that the user account does not need session administration level privileges to set restricted session variables, which are required to change the value of `sql_require_primary_key` to match the master's setting for each transaction.

(WL #13239)

- Since MySQL 8.0.19, compression has been supported for messages sent over X Protocol connections. Connections can be compressed if the server and the client agree on a compression algorithm to use. By default, the server permits the Deflate, LZ4, and zstd compression algorithms, or you can set the `mysqlx_compression_algorithms` system variable to include only the ones you permit. In MySQL 8.0.19, X Protocol uses the library default compression level for each algorithm, and the client cannot negotiate this.

  From MySQL 8.0.20, the client can request a specific compression level during capability negotiations for an X Protocol connection. X Protocol sets a maximum compression level for each algorithm, which prevents the server from agreeing to high compression levels that are requested by clients if that would consume too much resource on the server. The maximum compression levels are initially set to 5 for Deflate, 8 for LZ4, and 11 for zstd. You can adjust these settings using the new `mysqlx_deflate_max_client_compression_level`, `mysqlx_lz4_max_client_compression_level`, and `mysqlx_zstd_max_client_compression_level` system variables.

  New default compression levels for X Protocol have also been selected through performance testing as being a good trade-off between compression time and network transit time. These defaults are not necessarily the same as the library default for each algorithm. They are applied if the client does not request a compression level for the algorithm. The default compression levels are initially set to 3 for Deflate, 2 for LZ4, and 3 for zstd. You can adjust these settings using the new `mysqlx_deflate_default_compression_level`, `mysqlx_lz4_default_compression_level`, and `mysqlx_zstd_default_compression_level` system variables. (WL #13034)

## Bugs Fixed

- **Incompatible Change:** Some queries that used `ST_Contains()` did not return any results unless `> 0` was added.

  **Note**

  For upgrades from earlier versions of MySQL, you should recreate spatial indexes in tables that have them.

  (Bug #30461595, Bug #97347)

- **Performance:** Certain queries against tables with spatial indexes were not performed as efficiently following an upgrade from MySQL 5.7 to MySQL 8.0. (Bug #94655, Bug #29488350)

  References: See also: Bug #89551, Bug #27499984.

- **NDB Cluster:** `NDB` defines one `SPJ` worker per node owning a primary partition of the root table. If this table used read from any fragment replica, `DBTC` put all `SPJ` workers in the same `DBSPJ` instance, which effectively removed the use of some `SPJ` workers. (Bug #30639165)

- **NDB Cluster:** Executing the `SHOW` command using an `ndb_mgm` client binary from NDB 8.0.16 or earlier to access a management node running NDB 8.0.17 or later produced the error message `Unknown field: is_single_user`. (Bug #30599413)

  References: See also: Bug #16275500.

- **InnoDB:** A `CREATE UNDO TABLESPACE` operation that specified an undo data file name without specifying a path removed an existing undo data file of the same name from the directory specified by

`innodb_undo_directory` variable. The file name conflict check was performed on the data directory instead of the directory specified by the `innodb_undo_directory` variable. (Bug #30908328, Bug #98628)

- **InnoDB:** In debug builds, a regression introduced in MySQL 8.0.19 slowed down mutex and rw-lock deadlock debug checks. (Bug #30886393)

  References: This issue is a regression of: Bug #30628872.

- **InnoDB:** Valgrind testing raised an error indicating that a conditional jump or move depends on an uninitialized value. The error was a false-positive due to invalid validation logic. (Bug #30837136)

- **InnoDB:** Missing barriers in `rw_lock_debug_mutex_enter()` (in source file `sync0debug.cc`) could cause a thread to wait without ever being woken up. (Bug #30819167)

- **InnoDB:** To improve server initialization speed on Linux, `posix_fallocate()` is now used to allocate space for redo log files. (Bug #30804431, Bug #98342)

- **InnoDB:** A data dictionary table open function was implemented with incorrect lock ordering. (Bug #30782103, Bug #97825)

- **InnoDB:** Changes to parallel read threads functionality introduced in MySQL 8.0.17 caused a degradation in `SELECT COUNT(*)` performance. Pages were read from disk unnecessarily. (Bug #30766089)

- **InnoDB:** DDL logging was not performed for SQL operations executed by the bootstrap thread using the `init_file` startup variable, causing files to be left behind that should have been removed during a post-DDL stage. (Bug #30721214, Bug #98131)

- **InnoDB:** Adding an index on a column cast as a JSON array on a table with a specific number of records failed with an “Incorrect key file for table” error. (Bug #30709525, Bug #98098)

- **InnoDB:** A Valgrind error reported that an uninitialized `lock->writer_thread` value was used in a conditional jump. (Bug #30694177)

- **InnoDB:** An internal buffer pool statistics counter (`n_page_gets`) was partitioned by page number to avoid contention when accessed by multiple threads. (Bug #30604841, Bug #97822)

- **InnoDB:** A tablespace import operation failed with a schema mismatch error due to the `.cfg` file and the data dictionary both containing default values for a column that was added using `ALGORITHM=INSTANT`. An error should only occur if default values differ. (Bug #30561144)

- **InnoDB:** A slow shutdown failed to flush some GTIDs, requiring recovery of unflushed GTIDs from the undo log. (Bug #30548229)

- **InnoDB:** A broken alignment requirement in the code that allocates a prefix in memory for Performance Schema memory allocations caused a failure on MySQL builds optimized for macOS and FreeBSD. (Bug #30530857)

- **InnoDB:** Adding a virtual column raised an assertion failure due to data that was missing from the new data dictionary object created for the table. (Bug #30524263)

- **InnoDB:** A required latch was not taken when checking the mode of an undo tablespace. A required latch was also not taken when checking whether an undo tablespace is empty. (Bug #30509134)

- **InnoDB:** Allocating an update undo log segment to an XA transaction for persisting a GTID value before the transaction performed any data modifications caused a failure. (Bug #30456328)

- **InnoDB:** A query executed on a partitioned table with a discarded tablespace raised an assertion failure. (Bug #30437407, Bug #97271)
- **InnoDB:** The `row_upd_clust_rec_by_insert` function, which marks a clustered index record as deleted and inserts an updated version of the record into the clustered index, passed an incorrect `n_ext` value (the total number of external fields) to lower level functions, causing an assertion failure. (Bug #30437378)
- **InnoDB:** During a cloning operation, writes to the data dictionary buffer table at shutdown were too late, causing a failure. Newly generated dirty pages were not being flushed. (Bug #30427369, Bug #30405535, Bug #30405535)
- **InnoDB:** An operation performed with the `innodb_buffer_pool_evict` debug variable set to `uncompressed` caused an assertion failure. (Bug #30405531)
- **InnoDB:** Read-write lock code (`rw_lock_t`) that controls ordering of access to the boolean `recursive` flag and the writer thread ID using GCC builtins or `os_mutex` when the builtins are not available, was revised to use C++ `std::atomic` in some instances. Thanks to Yibo Cai from ARM for the contribution. (Bug #30401416, Bug #97150)
- **InnoDB:** A failure occurred while upgrading from MySQL 5.7 to MySQL 8.0. A server data dictionary object was missing information about the `FTS_DOC_ID` column and `FTS_DOC_ID_INDEX` that remain after dropping a `FULLTEXT` index. (Bug #30357954)
- **InnoDB:** Unnecessary messages about parallel scans were printed to the error log. (Bug #30330448)
- **InnoDB:** During upgrade from MySQL 5.7 to MySQL 8.0, clustered indexes named `GEN_CLUST_INDEX` are renamed to `PRIMARY`, which resulted in duplicate entries for the clustered indexes being added to the `mysql.innodb_index_stats` table. (Bug #30330448)
- **InnoDB:** Various internal functions computed write event slots in an inconsistent manner. (Bug #30228108, Bug #96519)
- **InnoDB:** Under specific circumstances, it was possible that tablespace encryption key information would not be applied during the redo log apply phase of crash recovery. (Bug #30209760)
- **InnoDB:** A file operation failure caused the page tracking archiver to fail, which in turn caused the main thread to hang, resulting in an assertion failure. Also, incorrectly, the page tracking archiver remained enabled in `innodb_read_only` mode. (Bug #30202643)
- **InnoDB:** An index corruption error was reported when attempting to import a tablespace containing a table column that was added using `ALGORITHM=INSTANT`. The error was due to missing metadata associated with the instantly added column. (Bug #30191523, Bug #96477)
- **InnoDB:** A transaction attempting to fetch an LOB record encountered a null LOB reference, causing an assertion failure. However, the null LOB reference was valid in this particular scenario because the LOB value was not yet fully written. (Bug #30144303)
- **InnoDB:** During a parallel read operation, the rollback of a table load operation while `autocommit` was disabled resulted in a server to exit due to assertion code that did not account for the possibility of tree structure changes during a parallel read. (Bug #30060690)
- **InnoDB:** The current size value maintained in a rollback segment memory object was found to be invalid, causing an assertion failure in function `trx_purge_free_segment()`. A validation routine (`trx_rseg_t::validateCurrSize()`) was added to verify the current size value. (Bug #29947027)
- **InnoDB:** A prepared statement executed with invalid parameter values raised an assertion failure. (Bug #29880907)

- **InnoDB:** An add column operation caused an assertion failure. The failure was due to a dangling pointer. (Bug #29866408)

  References: This issue is a regression of: Bug #28491099.

- **InnoDB:** Updating certain `InnoDB` system variables that take string values raised invalid read errors during Valgrind testing. (Bug #29717909, Bug #95215)

- **InnoDB:** Redo log records for modifications to undo tablespaces increased in size in MySQL 8.0 due to a change in undo tablespace ID values, which required additional bytes. The change in redo log record size caused a performance regression in workloads with heavy write I/O. To address this issue, the redo log format was modified to reduce redo log record size for modifications to undo tablespaces. (Bug #29536710)

- **InnoDB:** Additional information about `InnoDB` file writes, including progress data, is now printed to the error log. (Bug #29472295, Bug #94634)

- **InnoDB:** An insert statement on a table with a spatial index raised a record type mismatch assertion due to a tuple corruption. (Bug #29465567)

- **InnoDB:** A function that calculates undo log record size could calculate an incorrect length value in the case of a corrupted undo log record, resulting in a malloc failure. Assertion code was added to detect incorrect calculations. (Bug #29448406, Bug #82734)

- **Replication:** While an SQL statement was in the process of being rewritten for the binary log so that sensitive information did not appear in plain text, if a `SHOW PROCESSLIST` statement was used to inspect the query, the query could become corrupted when it was written to the binary log, causing replication to stop. The process of rewriting the query is now kept private, and the query thread is updated only when rewriting is complete. (Bug #30569003, Bug #97531, Bug #30654405)

- **Replication:** When a `GRANT` or `REVOKE` statement is only partially executed, an incident event is logged in the binary log, which makes the replication slave's applier thread stop so that the slave can be reconciled manually with the master. Previously, if a failed `GRANT` or `REVOKE` statement was the first statement executed in the session, no GTID was applied to the incident event (because the cache manager did not yet exist for the session), causing an error on the replication slave. Also, no incident event was logged in the situation where a `GRANT` statement created a user but then failed because the privileges had been specified incorrectly, again causing an error on the replication slave. Both these issues have now been fixed. (Bug #30566518, Bug #30324661)

- **Replication:** Compression is now triggered for the `mysql.gtid_executed` table when the `thread/sql/compress_gtid_table` thread is launched after the server start, and the effects are visible when the compression process is complete. (Bug #30541799)

- **Replication:** In the event of an unplanned disconnection of a replication slave from the master, the reference to the master's dump thread might not be removed from the list of registered slaves, in which case statements that accessed the list of slaves would fail. The issue has now been fixed. (Bug #29915479)

- **Replication:** When a partitioned table was involved, the server did not correctly handle the situation where a row event could not be written to the binary log due to a lack of cache space. An appropriate error is now returned in this situation. (Bug #29848931)

- **Replication:** With the settings `binlog_format=MIXED`, `tx_isolation=READ-COMMITTED`, and `binlog_row_image=FULL`, an `INSERT ... SELECT` query involving a transactional storage engine omitted any columns with a null value from the row image written to the binary log. This happened because when processing `INSERT ... SELECT` statements, the columns were marked for inserts

before the binary logging format was selected. The issue has now been fixed. (Bug #29110804, Bug #93423)

- **Replication:** Under certain conditions, replication of conditional comments could fail. (Bug #28388217)
- **Group Replication:** The thread used by the Group Replication message service was not correctly registered by the Performance Schema instrumentation, so the thread actions were not visible in Performance Schema tables. (Bug #30824676)
- **Group Replication:** Group Replication initiates and manages cloning operations for distributed recovery, but group members that have been set up to support cloning may also participate in cloning operations that a user initiates manually. In releases before MySQL 8.0.20, you could not initiate a cloning operation manually if the operation involved a group member on which Group Replication was running. From MySQL 8.0.20, you can do this, provided that the cloning operation does not remove and replace the data on the recipient. The statement to initiate the cloning operation must therefore include the `DATA DIRECTORY` clause if Group Replication is running. (Bug #30798640)
- **Group Replication:** For Group Replication channels, issuing the `CHANGE MASTER TO` statement with the `PRIVILEGE_CHECKS_USER` option while Group Replication was running caused the channel's relay log files to be deleted. Transactions that had been received and queued in the relay log, but not yet applied, could be lost in this situation. The `CHANGE MASTER TO` statement can now only be issued when Group Replication is not running. (Bug #30655369)
- **Group Replication:** The Group Replication failure detection mechanism raises a suspicion if a server stops sending messages, and the member is eventually expelled provided that a majority of the group members are still communicating. However, the failure detection mechanism did not take into account the situation where one or more of the group members in the majority had actually already been marked for expulsion, but had not yet been removed from the group. Where the network was unstable and members frequently lost and regained connection to each other in different combinations, it was possible for a group to end up marking all its members for expulsion, after which the group would cease to exist and have to be set up again.

  The Group Replication Group Communication System (GCS) now tracks the group members that have been marked for expulsion, and treats them as if they were in the group of suspect members when deciding if there is a majority. This ensures at least one member remains in the group and the group can continue to exist. When an expelled member has actually been removed from the group, GCS removes its record of having marked the member for expulsion, so that the member can rejoin the group if it is able to. (Bug #30640544)
- **Group Replication:** Performance Schema tables could not be accessed on a MySQL server with Group Replication that was running under high load conditions. (Bug #30112711, Bug #30675790)
- **Group Replication:** Internal queries from Group Replication to the Performance Schema for statistics on local group members failed if they occurred simultaneously with changes to the group's membership. Locking for the internal queries has been improved to fix the issue. (Bug #30049349, Bug #30791583, Bug #30963553)
- **Group Replication:** During the Group Replication distributed recovery process, if a joining member is unable to complete a remote cloning operation with any donor from the group, it uses state transfer from a donor's binary log to retrieve all of the required data. However, if the last attempted remote cloning operation was interrupted and left the joining member with incomplete or no data, an attempt at state transfer immediately afterwards could also fail. Before attempting state transfer following a failed remote cloning operation, Group Replication now checks that the remote cloning operation did not reach the stage of removing local data from the joining member. If data was removed, the joining member leaves the group and takes the action specified by the `group_replication_exit_state_action` system variable. (Bug #29669099, Bug #29944828)

- **Group Replication:** An earlier change to reduce Performance Schema memory instrumentation overhead had the unintended effect of causing Group Replication performance degradation. (Bug #28719976)

  References: This issue is a regression of: Bug #27500610.

- **Group Replication:** Before taking certain actions, Group Replication checks what transactions are running on the server. Previously, the service used for this check did not count transactions that were in the commit phase, which could result in the action timing out. Now, transactions that are in the commit phase are included in the set of currently ongoing transactions. (Bug #28327838)

- **JSON:** When `JSON_TABLE()` was used as part of an `INSERT` statement in strict mode, conversion errors handled by any `ON ERROR` clause could cause the `INSERT` to be rejected. Since errors are handled by an `ON ERROR` clause, the statement should not be rejected unless `ERROR ON ERROR` is actually specified.

  This issue is fixed by ignoring warnings when converting values to the target type if `NULL ON ERROR` or `DEFAULT ... ON ERROR` has been specified or is implied. (Bug #30628330)

- **JSON:** The output from `JSON_TABLE()` was not always correct when used in views. This fix corrects the following issues:

  - Column names were not quoted, causing syntax errors when quoting was needed for these.
  - Some column types were misreported.
  - Some column type attributes such as UNSIGNED were lost.
  - Column character set and collation were lost.

  (Bug #30263373)

- **JSON:** The functions `JSON_SCHEMA_VALID()` and `JSON_SCHEMA_VALIDATION_REPORT()` formerly checked to ensure that their arguments were convertible to `JSON` each time a prepared statement including these was executed, which was neither efficient nor necessary. Now in such cases, the check is performed only once, when the statement is prepared. (Bug #97878, Bug #30622327)

- Privilege requirements were checked incorrectly for stored objects with a `DEFINER` that has the `SYSTEM_USER` privilege. (Bug #31077699)

- A number of errors reported by Clang in the documentation generated from the MySQL sources have been corrected. (Bug #30956093)

- On FreeBSD, the krb5 package is a now a dependency. (Bug #30887620)

- If a query contained multiple references to the same common table expression (CTE) and a pseudo-comment crossed borders of the CTE definition, the parser failed with confusing syntax error messages. (Bug #30871301)

- For installation using Debian packages, the `/var/run/mysqld` directory was not created. (Bug #30855015, Bug #98484)

- `mysqlslap` did not shut down its threads properly when SQL statements returned an error. This could result in attempts to free already freed memory. (Bug #30850310)

- When X Plugin was attempting to add a document to a collection as either an insertion or an update in the case of a duplicate key, in the case where the document failed a unique key constraint in a field other

than the primary key, the error returned by X Plugin did not state that this was the cause of the issue. The appropriate error is now returned. (Bug #30843865)

- An integer value generated by transformations in the resolver was supplied to a test which expected a boolean. (Bug #30837240)
- A query using an `IN` expression that accessed one or more columns holding large string values could lead to a memory leak. (Bug #30814171)
- Statements did not work properly when the target of a `DELETE` was a common table expression. (Bug #30796015, Bug #98330)
- Starting the server with `create_admin_listener_thread` enabled and without `admin_address` enabled caused an abnormal exit during the server shutdown process. (Bug #30785609)
- When a table had both a primary key and a secondary key on the same column, but for different lengths, the range optimizer chose the wrong key part in the secondary index for comparing range values. (Bug #30783011)
- In some cases, errors caused when `DISTINCT` was used with an aggregate function whose argument was of an incorrect type were not propagated correctly. (Bug #30782687)
- For replication using compression, the slave could raise an assertion if the master was restarted. (Bug #30774692)
- For debug builds, the server could exit trying to print an optimizer trace. (Bug #30773218, Bug #98258)
- The `mysql_real_connect_nonblocking()` C API function exhibited blocking behavior. (Bug #30771233)
- With `LOCK TABLES` active, while processing `INFORMATION_SCHEMA` queries, the server could attempt to lock internal temporary tables (which need no locks), causing an assertion to be raised. (Bug #30764651, Bug #98221)
- The `mysqldump` internal network timeout was increased from 700 to 86400 seconds to accommodate connecting to busy or unresponsive servers. (Bug #30755992, Bug #98203)
- Configuring with `-DWITH_SASL=`*`path/to/custom/installation`* inadvertently caused `libsasl` to be linked into the `daemon_memcached` plugin. (Bug #30755301)
- After deleting the temporary table associated with a window function's frame buffer, the temporary table parameter for the frame buffer was not cleaned up, causing string buffers associated with copy fields not to be freed properly. (Bug #30752366)
- The `-libs-compat` RPM package is now built with system `zlib` to avoid problems with unrestricted export of symbols in `libmysqlclient.so.18`. (Bug #30722389, Bug #98130)
- The server exited histogram sampling prematurely, causing an assertion failure. An unnecessary boolean variable that marked the completion of a sampling operation was removed. (Bug #30717778)
- When removing a `WHERE` condition because one of the participating conditions was always false, a materialized derived table was not cleaned up properly, resulting in a memory leak. (Bug #30712243)
- Multiple comparisons with the same `GEOMETRY` value were not always handled correctly. (Bug #30697042)

  References: See also: Bug #30306306.

- `MIN()` and `MAX()` could return an incorrect value for some queries if a `WHERE` clause containing an `IN ()` subquery was added. (Bug #30691682, Bug #98047)
- Server startup failed if MySQL Enterprise Firewall was enabled at startup but the whitelist and user tables were missing. (Bug #30690181)
- For prepared statements, re-execution could cause a server exit if a cleaned-up materialized temporary table was still being referred to. (Bug #30674598)
- The `ER_WARN_DEPRECATED_SQL_CALC_FOUND_ROWS` and `ER_WARN_DEPRECATED_FOUND_ROWS` error messages were incorrectly categorized in the range of messages meant to be written to the error log. They are now correctly categorized as messages meant to be sent to clients. The old errors are now designated as `OBSOLETE_ER_WARN_DEPRECATED_SQL_CALC_FOUND_ROWS` and `OBSOLETE_ER_WARN_DEPRECATED_FOUND_ROWS` in the range of error-log messages. (Bug #30673043)
- Some joins within subqueries where an outer query used `EXISTS` or `NOT EXISTS` were not always handled correctly. (Bug #30671329)
- Queries using `ORDER BY` *`constant`* are permitted but an `ORDER BY` clause of this sort should not have any effect on the result; such queries were not always handled correctly. (Bug #30669493)
- A missing out-of-bounds check in `wild_case_match()` caused a pointer to read out of bounds. (Bug #30668886)
- The `strconvert()` function was not safe for conversions between `filename` and `utf8_general_ci` strings. (Bug #30668847)
- Some filesorts using keys of fixed length were not always handled correctly. (Bug #30665034)
- When performing a hash join on two string columns that were potentially very large (in particular, `BLOB` columns with `PAD SPACE` collations), MySQL stored the entire sort key in the row, which impacted performance by requiring large amounts of memory. Now only a collation-aware hash is stored, with an added equality comparison prevent a wrong answer, even in the event of a 64-bit hash collision. (Bug #30664831)
- When at least two tables were joined to at least two other tables using a semijoin, and the join optimizer chose to use a loose scan, it was possible to place both of the left tables below the deduplicating nested loop iterator, leading to excessive deduplication. We fix this by treating a loose scan across multiple tables as a separate internal structure. (Bug #30659810)
- In unions of a `const` table and zero or more known-zero expressions, derived tables of exactly one row could be read incorrectly as having zero rows. (Bug #30655712, Bug #97967)
- A MySQL 8.0.19 patch set an invalid `INFORMATION_SCHEMA` and data dictionary version number. Assertion code was added to prevent future version information errors. (Bug #30645158, Bug #97948)

  References: This issue is a regression of: Bug #29871530.
- When setting up the iterator tree, the optimizer now filters away and subsequently ignores conditions which are known to be trivially true. (Bug #30644591)
- Under some conditions, `SHOW COLUMNS` on a temporary `MERGE` table could raise an assertion or cause a server exit. (Bug #30640463)

  References: This issue is a regression of: Bug #28811287, Bug #92834.
- The Event Scheduler had a memory leak. (Bug #30628268)

- Using the asynchronous C API functions could result in freeing already freed memory. (Bug #30596999, Bug #97805)
- On tables containing a `CHECK` constraint, certain simple queries were inefficient due to excessive memory allocation and Performance Schema calls. (Bug #30594613)
- Under certain circumstances, a memcached command could result in reading an uninitialized memory buffer, causing a failure. (Bug #30592346)
- A race condition could occur between `InnoDB` issuing requests for schema and table metadata while filling `INFORMATION_SCHEMA.INNODB_TABLES`, and the schema being dropped, leading to user queries on `INNODB_TABLES` reporting an error. (Bug #30591967)
- The client library could be induced into an infinite loop by a malicious server. (Bug #30581726)
- Using `ALTER USER` to reset an account `MAX_USER_CONNECTIONS` value did not take effect until all current account connections terminated, if there were any. (Bug #30578217, Bug #97735)
- When the optimizer sets up a weedout, it notifies all tables that are part of the weedout that they should provide row IDs. For confluent weedouts (weedouts returning at most one row), the optimizer expects that the executor handles the weedout without row IDs. In the iterator executor, confluent weedouts are implemented using `LIMIT 1`; the normal weedout iterator does not handle confluent weedouts, and thus always expects row IDs. In the case of a confluent weedout on the right side of an outer join, the confluent weedout was processed as a normal weedout, causing the iterator executor to ask for row IDs where the tables did not supply them. Now in such cases, the `LIMIT 1` optimization is also applied. (Bug #30566549, Bug #30282693)
- `SET PERSIST` could fail due to attempting to persist variables to the wrong directory. (Bug #30561982)
- Within a stored program with an error handler defined for the error condition of accessing a nonexistent table, the handler was not invoked if the table was nonexistent because it was named in a nonexistent database. (Bug #30561920, Bug #97682)
- The duplicate weedout optimization strategy employed by MySQL (see Optimizing IN and EXISTS Subquery Predicates with Semijoin Transformations) uses an internal table of row IDs which it has already seen, with a unique index on the column containing these IDs. When the key for the unique index became too large, which could happen with very large row IDs, the server reverted to deduplication by hash key instead, with a separate index (not unique) over the hash field only, as with other temporary tables. Because the latter index was not properly initialized, affected queries were not executed properly and could lead to a premature exit. (Bug #30556257)
- For debug builds, under `LOCK TABLES`, the server could mishandle materialized temporary tables and raise an assertion. (Bug #30476213, Bug #97404)
- The internal array of materialized query blocks `SELECT_LEX_UNIT::m_query_blocks_to_materialize` was not reset between executions, which meant that it pointed to objects which were no longer valid when a prepared statement was executed a second time, causing the second execution to fail. (Bug #30438038)
- Altering column collations did not affect unique indexes until a server restart. (Bug #30386119, Bug #97103)
- When using roles, the `EXECUTE` privilege for stored functions was treated as a privilege for stored procedures. As a result, it was not possible to use `EXECUTE` as a role privilege for functions. (Bug #30376231)
- A materialized subquery including a condition in which a column value was used as input to a nondeterministic function produced incorrect results. (Bug #30368937)

- Several fixes were applied to the `InnoDB` memcached plugin. The fixes addressed potential deadlock issues, issues related to connection list latches, and removal of an obsolete flush mutex. (Bug #30354225)
- Strings that used the `utf8mb4_0900_bin` collation could not be compared with `utf8mb4` strings that used a different collation. Now the comparison is done by using `utf8mb4_0900_bin` for both strings. (Bug #30350111)
- During optimization, MySQL removes conditions in which all arguments are considered equal; for example, `1 <> 1` is removed and replaced with `false`. In doing so, conditions containing non-deterministic arguments were also removed, which caused a condition such as `RAND() < RAND()` to be considered an impossible condition. Now, the optimizer no longer removes conditions containing nondeterministic arguments. (Bug #30311271)
- Scheduling of events could be disturbed by removing events. (Bug #30301356, Bug #96849)
- The Event Scheduler reported warnings for Valgrind builds. (Bug #30301340)
- Shutting down the server while using the clone plugin raised a Valgrind error. (Bug #30248419)
- If the `mysqld-auto.cnf` file was malformed, the server did not start (expected), but did not report any error (unexpected). (Bug #30169731, Bug #96501)
- `UPDATE` statements could give an inconsistent number of rows matched (found rows) in cases where not all matched rows were updated, depending on the reason for rows not being updated. For example, rows not updated due to being updated through a view with a `WITH CHECK OPTION` clause were not counted as matching rows, whereas rows not updated due to a failing `CHECK CONSTRAINT` were counted. For consistency, rows that fail a `WITH CHECK OPTION` clause now are counted as matching rows. (Bug #30158954)
- When restarting the MySQL server on a cloned directory, `InnoDB` reported an error indicating that it could not find a tablespace file for a statistics table that was dropped by the server previously. (Bug #30093799)
- The server did not handle correctly a `UNION` in which one of the queries contained a subquery that used `ORDER BY`. (Bug #29952565)
- For `INFORMATION_SCHEMA` queries, a race condition could result in multiple attempts to insert a key when updating the dynamic statistics tables, producing a duplicate-key error. (Bug #29948755, Bug #95929)
- `SHOW CREATE VIEW` could fail with an illegal mix of collations for views defined on a function that returns a string. (Bug #29904087)
- The Performance Schema could fail to remove thread instrumentation when a thread was deleted. (Bug #29859605)
- A query with a `WHERE` clause whose predicate contained a numeric value in scientific notation was not handled correctly.

  In addition, attempting to insert a particular integer specified as a string caused a server exit when the string-to-integer conversion was not successful. (Bug #29723340, Bug #30441969)
- An internal interface was added for retrieving and parsing errors that occur on the donor MySQL server instance (`ER_CLONE_DONOR` errors) and for checking if data on the recipient has been dropped. (Bug #29682642)
- It was not possible to drop any columns from a table when the `DEFAULT` value. (Bug #29661106)

- For the CONNECTION_CONTROL plugin, the Performance Schema instrumentation used keys that were not discoverable to the Performance Schema unless the associated code actually executed. (Bug #29539976)

- For a nullable column c, the optimizer now recognizes when the conditions c < c, c > c, and c <> c are always false and need not be evaluated for every row. Thanks to Daniel Black for the contribution. (For nonnullable columns, the optimizer already recognized always-false conditions.) (Bug #29115386, Bug #93642)

- Reinitialization of character sets from Index.xml could cause a use-after-free error. (Bug #28956360, Bug #93276)

- The sys schema ps_setup_reset_to_default() procedure used MySQL 5.7 defaults, not MySQL 8.0 defaults. (Bug #27636611)

- Some connection encryption ciphers did not work. (Bug #27045306)

- Previously, mysqlpump read the [mysql_dump] and [client] groups from option files. mysqlpump now additionally reads the [mysqlpump] group. The [mysql_dump] group is still accepted but is deprecated. (Bug #24733245, Bug #83144)

- For a query of the form SELECT DISTINCT ... ORDER BY ..., when the ORDER BY was pushed down onto the first table in the join, the result was not always sorted in the correct order. (Bug #98217, Bug #30760534)

- The NULL indicator was not properly written for items used as variable-length keys, such that all such items were assumed to be not NULL, which was considered equal to the empty string when using certain collations. One visible effect of this issue was that ordering by an expression using a nullable string was sometimes not performed correctly. An example of such a query, where column c1 contains both NULL and empty string values, is shown here:

  ```
  SELECT c1, SUBSTR(c1, 1) AS c2 FROM t ORDER BY c2;
  ```

  (Bug #98035, Bug #30687020)

- A query returned inaccurate results when an expression in a GROUP BY clause used a column name differing in case from that used for the name of the column when the table containing this column was created. An example of this would be when the query used GROUP BY id although the column name as shown in the original CREATE TABLE statement was ID.

  This occurred because, the server performed case-sensitive comparisons of column names in expressions with names of columns in tables. This issue is fixed by ensuring that such comparisons are performed in a case-insensitive fashion as expected. (Bug #97628, Bug #98222, Bug #30541701, Bug #30761372)

- A multi-table UPDATE statement which updated a table joined to a derived table that joined two other tables was not optimized properly as it had been in MySQL 5.6, instead being treated as if STRAIGHT_JOIN had been used with the subquery creating the derived table. (Bug #97418, Bug #30488700)

- EXPLAIN now uses hash join instead of block nested loop, since the latter no longer exists and is replaced by a hash join in nearly all cases. (Bug #97299, Bug #30444550)

- References to columns from tables of outer query blocks in the ON condition of a join did not work, and could be used only in a WHERE. The fix for this problem means that a query such as this one now works correctly:

```
SELECT o.order_date FROM orders o
WHERE o.order_date IN  ( SELECT c.contact_name FROM customers c
INNER JOIN order_details od
ON o.order_id = od.discount );
```

Previously this had to be rewritten as shown here:

```
SELECT o.order_date FROM orders o
WHERE o.order_date IN  ( SELECT c.contact_name FROM customers c
INNER JOIN order_details od
ON 1
WHERE o.order_id = od.discount );
```

References to other tables of the same `FROM` clause as the join, as in the query `SELECT * FROM t1 CROSS JOIN (t2 LEFT JOIN t3 ON t1.c=3)`, are not outer references and remain forbidden. In this case, a lateral join is required, like this: `SELECT * FROM t1 JOIN LATERAL (SELECT * FROM t2 LEFT JOIN t3 ON t1.c=3)`. (Bug #96946, Bug #30350696)

- The execution plan for a query that filtered on the first column of a composite hash index wrongly used this index, producing erroneous results. (Bug #94737, Bug #29527115)

- References to columns from tables of outer query blocks in an `ON` condition of a `JOIN` did not work, and could be used only in a `WHERE`. The fix for this problem means that a query such as this one now works correctly:

```
SELECT o.order_date FROM orders o
WHERE o.order_date IN  ( SELECT c.contact_name FROM customers c
                    INNER JOIN order_details od
                    ON o.order_id = od.discount );
```

Previously this had to be rewritten as shown here:

```
SELECT o.order_date FROM orders o
WHERE o.order_date IN  ( SELECT c.contact_name FROM customers c
                    INNER JOIN order_details od
                    ON 1
                    WHERE o.order_id = od.discount );
```

References to other tables of the same `FROM` clause as the `JOIN`, as in the query `SELECT * FROM t1 CROSS JOIN (t2 LEFT JOIN t3 ON t1.c=3)`, are not outer references and remain forbidden. In this case, a lateral join is required, like this: `SELECT * FROM t1 JOIN LATERAL (SELECT * FROM t2 LEFT JOIN t3 ON t1.c=3)`. (Bug #35242, Bug #96946, Bug #11748138, Bug #30350696)

- There could be a mismatch between the version of OpenSSL used to build the server and the version used for other parts of MySQL such as libraries or plugins. This could cause certain features not to work, such as the LDAP authentication plugins. Now the same version of OpenSSL is used for building everything. (WL #13759)

- Previous work in MySQL 8.0 to optimize impossible expressions such as `a=b AND FALSE` as `FALSE` could make for less efficient execution when such expressions appeared as outer join conditions, due to the fact that the join was interpreted as a Cartesian product followed by a filter. (Bug #8202, Bug #89739, Bug #97552, Bug #11745046, Bug #27581277, Bug #30520749)

  References: See also: Bug #98206, Bug #30756135.

# Changes in MySQL 8.0.19 (2020-01-13)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

**Important**

There is an issue for MySQL 8.0.19 installed using MySQL Installer that prevents the server from starting if MySQL Enterprise Firewall is selected during the server configuration steps. If the server startup operation fails, click **Cancel** to end the configuration process and return to the dashboard. You must uninstall the server.

The workaround is to run MySQL Installer *without* MySQL Enterprise Firewall selected. Then install MySQL Enterprise Firewall afterward using the instructions for manual installation (see Installing or Uninstalling MySQL Enterprise Firewall). This problem is corrected in MySQL 8.0.20.


- Account Management Notes
- Audit Log Notes
- Compilation Notes
- Configuration Notes
- Deprecation and Removal Notes
- Error Handling
- SQL Function and Operator Notes
- INFORMATION_SCHEMA Notes
- Keyring Notes
- Logging Notes
- Packaging Notes
- SQL Syntax Notes
- sys Schema Notes
- Thread Pool Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Account Management Notes

- MySQL now enables administrators to configure user accounts such that too many consecutive login failures due to incorrect passwords cause temporary account locking. The required number of failures and the lock time are configurable per account, using the `FAILED_LOGIN_ATTEMPTS` and `PASSWORD_LOCK_TIME` options of the `CREATE USER` and `ALTER USER` statements. See Password Management. (Bug #27733694, Bug #90169, WL #13515)

## Audit Log Notes

- `ANALYZE TABLE` statements now produce `read` audit events. (Bug #29625461)

- Audit log connect events now include any connection attributes passed by the client. Connection attribute logging is supported for new-style XML log file format and JSON format, but not old-style XML format. See Audit Log File Formats. (WL #11378)

## Compilation Notes

- **Microsoft Windows:** On Windows, the minimum version of `CMake` for builds from the command line is now 3.15. (Bug #30332632, Bug #96954)

- `-DMYSQL_MAINTAINER_MODE` now includes `-Wshadow=local` for GCC and `-Wshadow-uncaptured-local` for Clang. (Bug #30099556)

## Configuration Notes

- New `FPROFILE_GENERATE` and `FPROFILE_USE` `CMake` options are available for experimenting with profile guided optimization (PGO) with GCC. See the `cmake/fprofile.cmake` in a MySQL source distribution for information about using them. These options have been tested with GCC 8 and 9, and with Clang.

  Enabling `FPROFILE_USE` also enables `WITH_LTO` (link time optimization). (Bug #30089834, Bug #96314, Bug #30133324, Bug #96410, Bug #30164113, Bug #96486)

- `Innodb_system_rows_read`, `Innodb_system_rows_inserted`, `Innodb_system_rows_deleted` status variables were added for counting row operations on `InnoDB` tables that belong to system-created schemas. The new status variables are similar to the existing `Innodb_rows_read`, `Innodb_rows_inserted`, `Innodb_rows_deleted` status variables, which count operations on `InnoDB` tables that belong to both user-created and system-created schemas.

  The new status variables are useful in replication environments where `relay_log_info_repository` and `master_info_repository` variables are set to `TABLE`, resulting in higher row operation counts on slaves due to operations performed on the `slave_master_info`, `slave_replay_log_info`, and `slave_worker_info` tables, which belong to the system-created `mysql` schema. For a valid comparison of master and slave row operation counts, operations on tables in system-created schemas can now be excluded using the count data provided by the new status variables.

  Thanks to Facebook for the contribution. (Bug #27724674, Bug #90148)

## Deprecation and Removal Notes

- The `thread_pool` plugin used display widths in definitions for integer columns of Performance Schema tables. This resulted in warnings written to the error log because integer column display widths are now deprecated. (Bug #30597673)

- Display width specification for integer data types was deprecated in MySQL 8.0.17, and now statements that include data type definitions in their output no longer show the display width for integer types, with these exceptions:

  - The type is `TINYINT(1)`. MySQL Connectors make the assumption that `TINYINT(1)` columns originated as `BOOLEAN` columns; this exception enables them to continue to make that assumption.

  - The type includes the `ZEROFILL` attribute.

  This change applies to tables, views, and stored routines, and affects the output from `SHOW CREATE` and `DESCRIBE` statements, and from `INFORMATION_SCHEMA` tables.

For DESCRIBE statements and INFORMATION_SCHEMA queries, output is unaffected for objects created in previous MySQL 8.0 versions because information already stored in the data dictionary remains unchanged. This exception does not apply for upgrades from MySQL 5.7 to 8.0, for which all data dictionary information is re-created such that data type definitions do not include display width. (Bug #30556657, Bug #97680, WL #13528)

- Setting the hash_join optimizer switch (see optimizer_switch system variable) no longer has any effect. The same applies with respect to the HASH_JOIN and NO_HASH_JOIN optimizer hints. Both the optimizer switch and the optimizer hint are now deprecated, and subject to removal in a future release of MySQL.

  This also fixes an issue whereby SELECT DISTINCT ... WITH ROLLUP did not always return all distinct rows. (Bug #27549694, Bug #30471809)

- In MySQL 5.7.14, the mysqlx namespace parameter was introduced for X Protocol's StmtExecute request, replacing the xplugin parameter, which was therefore deprecated. X Plugin continued to support the deprecated xplugin namespace for backward compatibility. In MySQL 8.0.19, the xplugin namespace has now been removed. If the xplugin namespace is used from this release on, an error message is returned as for an unknown namespace. X Plugin's Mysqlx_stmt_execute_xplugin status variable, which counted the number of StmtExecute requests received for the xplugin namespace, is no longer used from MySQL 8.0.19. (WL #13057)

- Support for the YEAR(2) data type was removed in MySQL 5.7.5, leaving only YEAR and YEAR(4) as valid specifications for year-valued data. Because YEAR and YEAR(4) are semantically identical, specifying a display width is unnecessary, so YEAR(4) is now deprecated and support for it will be removed in a future MySQL version. Statements that include data type definitions in their output no longer show the display width for YEAR. This change applies to tables, views, and stored routines, and affects the output from SHOW CREATE and DESCRIBE statements, and from INFORMATION_SCHEMA tables.

  For DESCRIBE statements and INFORMATION_SCHEMA queries, output is unaffected for objects created in previous MySQL 8.0 versions because information already stored in the data dictionary remains unchanged. This exception does not apply for upgrades from MySQL 5.7 to 8.0, for which all data dictionary information is re-created such that data type definitions do not include display width.

  The (undocumented) UNSIGNED attribute for YEAR is also now deprecated and support for it will be removed in a future MySQL version. (WL #13537)

## Error Handling

- Error messages regarding crash recovery for XA were revised to indicate XA context to distinguish them from non-XA crash recovery messages. (Bug #30578290, Bug #97743)

- Previously, the server returned this error message for attempts to use LOAD DATA LOCAL with LOCAL capability disabled: The used command is not allowed with this MySQL version. This was misleading because the error condition is not related to the MySQL version. The server now returns an error code of ER_CLIENT_LOCAL_FILES_DISABLED and this message: Loading local data is disabled; this must be enabled on both the client and server side. (Bug #30375698, Bug #29377985, Bug #94396)

## SQL Function and Operator Notes

- Previously, loadable functions took no account of the character set or collation of string arguments or return values. In effect, string arguments and return values were treated as binary strings, with the implication that only string arguments containing single-byte characters could be handled reliably.

Loadable function behavior is still the same by default, but the interface for writing loadable functions has been extended to enable them to determine the character set and collation of string arguments, and to return strings that have a particular character set and collation. These capabilities are optional for loadable function writers, who may take advantage of them as desired. See Loadable Function Character Set Handling

Of the loadable functions distributed with MySQL, those associated with the following features and extensions have been modified to take advantage of the new capabilities: MySQL Enterprise Audit, MySQL Enterprise Firewall, MySQL Enterprise Data Masking and De-Identification, MySQL Keyring (the general-purpose keyring functions only), and Group Replication. The modification applies only where it make sense. For example, a function that returns encrypted data is *intended* to return a binary string, not a character string.

Character-set capabilities for loadable functions are implemented using the `mysql_udf_metadata` component service. For information about this service, see the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html (search for `s_mysql_mysql_udf_metadata` and `udf_metadata_imp`). Source code for the MySQL Keyring functions is available in Community source distributions and may be examined as examples for third-party loadable function writers who wish to modify their own functions to be character set-aware. (WL #12370)

## INFORMATION_SCHEMA Notes

- The `INFORMATION_SCHEMA` contains several new tables that expose role information:
  - `ADMINISTRABLE_ROLE_AUTHORIZATIONS`: Roles the current user can grant; see The INFORMATION_SCHEMA ADMINISTRABLE_ROLE_AUTHORIZATIONS Table.
  - `APPLICABLE_ROLES`: Roles applicable for the current user; see The INFORMATION_SCHEMA APPLICABLE_ROLES Table.
  - `ENABLED_ROLES`: Roles enabled within the current session; see The INFORMATION_SCHEMA ENABLED_ROLES Table.
  - `ROLE_COLUMN_GRANTS`: Column privileges for roles for the current user; see The INFORMATION_SCHEMA ROLE_COLUMN_GRANTS Table.
  - `ROLE_ROUTINE_GRANTS`: Routine privileges for roles for the current user; see The INFORMATION_SCHEMA ROLE_ROUTINE_GRANTS Table.
  - `ROLE_TABLE_GRANTS`: Table privileges for roles for the current user; see The INFORMATION_SCHEMA ROLE_TABLE_GRANTS Table.

  (WL #10895)

## Keyring Notes

- A new `SECRET` key type is available that is intended for general-purpose storage of sensitive data using the MySQL keyring. The keyring encrypts and decrypts `SECRET` data as a byte stream upon storage and retrieval. The `SECRET` key type is supported by all keyring plugins. See Supported Keyring Key Types and Lengths. (WL #12859)

## Logging Notes

- The `SIGUSR1` signal now causes the server to flush the error log, general query log, and slow query log. One use for `SIGUSR1` is to implement log rotation without having to connect to the server (which to flush

logs requires an account that has the RELOAD privilege). The server response to SIGUSR1 is a subset of the response to SIGHUP, enabling SIGUSR1 to be used as a more “lightweight” signal that flushes certain logs without the other SIGHUP effects such as flushing the thread and host caches and writing a status report to the error log. See Unix Signal Handling in MySQL. (WL #13689)

## Packaging Notes

- Binary packages that include curl rather than linking to the system curl library have been upgraded to use curl 7.66.0. (Bug #30356844)
- The zstd library bundled with MySQL has been upgraded from version 1.3.3 to 1.4.3. MySQL uses the zstd library to support connection compression. (Bug #30236685)
- For package types for which OpenSSL shared libraries are included, they are now also included under lib/private if the package has private-to-MySQL libraries located there that need OpenSSL. (Bug #29966296)

## SQL Syntax Notes

- **Important Change:** MySQL now supports explicit table clauses and table value constructors according to the SQL standard. These have now been implemented, respectively, as the TABLE statement and the VALUES statement, each described in brief here:

  - TABLE *table_name* is equivalent to SELECT * FROM *table_name*, and can be used anywhere that the equivalent SELECT statement would be accepted; this includes joins, unions, INSERT ... SELECT statements, REPLACE statements, CREATE TABLE ... SELECT statements, and subqueries.

    You can use ORDER BY with TABLE, which also supports LIMIT with optional OFFSET; these clauses function in the same way in a TABLE statement as they do with SELECT. The following two statements produce the same result:

    ```
    TABLE t ORDER BY c LIMIT 10 OFFSET 3;

    SELECT * FROM t ORDER BY c LIMIT 10 OFFSET 3;
    ```

  - VALUES consists of the VALUES keyword followed by a series of row constructors (ROW()), separated by commas. It can be used to supply row values in an SQL-compliant fashion to an INSERT statement or REPLACE statement. For example, the following two statements are equivalent:

    ```
    INSERT INTO t1 VALUES ROW(1,2,3), ROW(4,5,6), ROW(7,8,9);

    INSERT INTO t1 VALUES (1,2,3), (4,5,6), (7,8,9);
    ```

    You can also select from a VALUES table value constructor just as you would a table, bearing in mind that you must supply a table alias when doing so. Using column aliases, you can also select individual columns, like this:

    ```
    mysql> SELECT a,c FROM (VALUES ROW(1,2,3), ROW(4,5,6)) AS t(a,b,c);
    +---+---+
    | a | c |
    +---+---+
    | 1 | 3 |
    | 4 | 6 |
    +---+---+
    ```

    You can employ such SELECT statements in joins, unions, subqueries, and other constructs in which you normally expect to be able to use such statements.

For more information and examples, see TABLE Statement, and VALUES Statement, as well as INSERT ... SELECT Statement, CREATE TABLE ... SELECT Statement, JOIN Clause, UNION Clause, and Subqueries. (Bug #77639, WL #10358)

- Previously, it was not possible to use `LIMIT` in the recursive `SELECT` part of a recursive common table expression (CTE). `LIMIT` is now supported in such cases, along with an optional `OFFSET` clause. An example of such a recursive CTE is shown here:

```
WITH RECURSIVE cte AS  (
  SELECT CAST("x" AS CHAR(100)) AS a FROM DUAL
  UNION ALL
  SELECT CONCAT("x",cte.a) FROM cte
    WHERE LENGTH(cte.a) < 10
    LIMIT 3 OFFSET 2
)
SELECT * FROM cte;
```

This statement produces the following output in the `mysql` client:

```
+-------+
| a     |
+-------+
| xxx   |
| xxxx  |
| xxxxx |
+-------+
```

Specifying `LIMIT` in this fashion can make execution of the CTE more efficient than doing so in the outermost `SELECT`, since only the requested number of rows is generated.

For more information, see Recursive Common Table Expressions. (Bug #92857, Bug #28816906, WL #12534)

- When `CHECK` constraints were implemented in MySQL 8.0.16, `ALTER TABLE` supported `DROP CHECK` and `ALTER CHECK` syntax as MySQL extensions to standard SQL for modifying check constraints, but did not support the more general (and SQL standard) `DROP CONSTRAINT` and `ALTER CONSTRAINT` syntax for modifying existing constraints of any type. That syntax is now supported; the constraint type is determined from the constraint name. (WL #12787)

- MySQL now supports aliases in the `VALUES` and `SET` clauses of `INSERT INTO ... ON DUPLICATE KEY UPDATE` statement for the row to be inserted and its columns. Consider a statement such as this one:

```
INSERT INTO t
    VALUES (9,5), (7,7), (11,-1)
    ON DUPLICATE KEY UPDATE a = a + VALUES(a) - VALUES(b);
```

Using the alias `new` for the inserted row, you can now rewrite the statement, referring back to the row alias in the `ON DUPLICATE KEY UPDATE` clause, like this:

```
INSERT INTO t
    VALUES (9,5), (7,7), (11,-1) AS new
    ON DUPLICATE KEY UPDATE a = a + new.a - new.b;
```

Using the same row alias, and, additionally, the column aliases `m` and `n` for the columns of the inserted row, you can omit the row alias and use only the column aliases, as shown here:

```
INSERT INTO t
    VALUES (9,5), (7,7), (11,-1) AS new(m,n)
```

```
    ON DUPLICATE KEY UPDATE a = a + m - n;
```

The row alias must be distinct from the table name; column aliases must be distinct from one another.

See INSERT ... ON DUPLICATE KEY UPDATE Statement, for more information and examples. (WL #6312)

## sys Schema Notes

- `sys` schema objects have been reimplemented not to invoke the deprecated `sys.format_bytes()`, `sys.format_time()`, and `sys.ps_thread_id()` stored functions. Instead, they invoke the equivalent built-in SQL functions implemented in MySQL 8.0.16 that format or retrieve Performance Schema data (see Changes in MySQL 8.0.16 (2019-04-25)).

  `sys.format_bytes()`, `sys.format_time()`, and `sys.ps_thread_id()` will be removed in a future MySQL version, so applications that use them should be adjusted to use the built-in functions instead, keeping in mind some minor differences between the `sys` functions and the built-in functions. See Performance Schema Functions. (WL #13439)

## Thread Pool Notes

- By default, the thread pool plugin tries to ensure a maximum of one thread executing in each group at any time. The default algorithm takes stalled threads into account and may temporarily permit more active threads. The plugin now implements a new `thread_pool_max_active_query_threads` system variable for controlling number of active threads per group. If `thread_pool_max_active_query_threads` is 0, the default algorithm applies. If `thread_pool_max_active_query_threads` is greater than 0, it places a limit on the number of active threads per group. See Thread Pool Operation. (WL #12915)

## X Plugin Notes

- X Plugin could not be compiled on Debian with GCC 9. The `--no-as-needed` linker option was added to provide a workaround for the issue. (Bug #30445201)
- Using X Protocol to query the Information Schema table `TRIGGERS` could result in errors being returned or some rows not being returned. (Bug #30318917)

## Functionality Added or Changed

- **Group Replication:** Replication connections to a replication slave, and Group Replication connections for distributed recovery, now have full client side configuration options for the TLSv1.3 protocol. In MySQL releases where TLSv1.3 support was available but these configuration options were not available, if TLSv1.3 was used for these connection types, the client in the connection (the replication slave or the joining Group Replication member that initiated distributed recovery) could not be configured. This meant that the server in the connection (the replication master or the Group Replication existing member that was the donor for distributed recovery) had to permit the use of at least one TLSv1.3 ciphersuite that is enabled by default. From MySQL 8.0.19, you can use the configuration options to specify any selection of ciphersuites for these connections, including only non-default ciphersuites if you want.

  The new configuration options are as follows:

  - Group Replication system variables `group_replication_recovery_tls_version` and `group_replication_recovery_tls_ciphersuites`. `group_replication_recovery_tls_version` specifies a list of permitted TLS protocols for connection encryption for the client instance (the joining member) in the distributed recovery

connection. `group_replication_recovery_tls_ciphersuites` specifies a list of permitted ciphersuites when TLSv1.3 is used for that connection.

- A `MASTER_TLS_CIPHERSUITES` option on the `CHANGE MASTER TO` command, to specify a list of TLSv1.3 ciphersuites permitted by the replication slave for the connection to the replication master. (The `CHANGE MASTER TO` command already had a `MASTER_TLS_VERSION` option to specify the permitted TLS protocol versions for the connection.)

(Bug #29960735, WL #13392)

- **Group Replication:** The Group Replication plugin interacts with MySQL Server using internal sessions to perform SQL API operations. Previously, these sessions counted towards the client connections limit specified by the `max_connections` server system variable. If the server had reached this limit when Group Replication was started or attempted to perform an operation, the operation was unsuccessful and Group Replication or the server itself might stop. From MySQL 8.0.19, Group Replication interactions with MySQL Server use a new component service that handles the internal sessions separately, which means that they do not count towards the `max_connections` limit and are not refused if the server has reached this limit. (Bug #29635001, WL #13378)

- **Microsoft Windows:** Previously, the `system` (`\!`) command for the `mysql` command-line client worked only for Unix systems. It now works on Windows as well. For example, `system cls` or `\! cls` may be used to clear the screen. (Bug #11765690, Bug #58680, WL #13391)

- **JSON:** When using `JSON_SCHEMA_VALID()` to specify a `CHECK` constraint on a table containing one or more JSON columns and experiencing a validation failure, MySQL now provides detailed information about the reasons for such failures. A new error `ER_JSON_SCHEMA_VALIDATION_ERROR_WITH_DETAILED_REPORT` is implemented containing this information, which can be viewed in the `mysql` client by issuing `SHOW WARNINGS` when an `INSERT` statement is rejected by the server.

  For more information and examples, see JSON_SCHEMA_VALID() and CHECK constraints. For more general information, see also CHECK Constraints. (WL #13195)

- Debian packages now contain more general systemd support that better supports manual `mysqld` execution. (Bug #29702050, Bug #95163)

- Duplicate key error information was extended to include the table name of the key. Previously, duplicate key error information included only the key value and key name. Thanks to Facebook for the contribution. (Bug #28686224, Bug #925308, WL #12589)

- When the `mysql` client operates in interactive mode, the `--binary-as-hex` option now is enabled by default. In addition, output from the `status` (or `\s`) command includes this line when the option is enabled implicitly or explicitly:

  ```
  Binary data as: Hexadecimal
  ```

  To disable hexadecimal notation, use `--skip-binary-as-hex` (Bug #24432545, WL #13038)

- MySQL now supports datetime literals with time zone offsets, such as `'2019-12-11 10:40:30-05:00'`, `'2003-04-14 03:30:00+10:00'`, and `'2020-01-01 15:35:45+05:30'`; these offsets are respected but not stored when inserting such values into `TIMESTAMP` and `DATETIME` columns; that is, offsets are not displayed when retrieving the values.

  The supported range for a timezone offset is `-13:59` to `+14:00`, inclusive. Time zone names such as `'CET'` or `'America/Argentina/Buenos_Aires'`, including the special value `'SYSTEM'`, are not supported in datetime literals. In addition, in this context, a leading zero is required for an hour value less than 10, and MySQL rejects the offset `'-00:00'` as invalid.

Datetime literals with time zone offsets can also be used as parameter values in prepared statements.

As part of this work, the allowed range of numeric values for the `time_zone` system variable has been changed, so that it is now also `-13:59` to `+14:00`, inclusive.

For additional information and examples, see The DATE, DATETIME, and TIMESTAMP Types, and MySQL Server Time Zone Support. (Bug #83852, Bug #25108148, WL #10828)

- From MySQL 8.0.19, compression is supported for messages sent over X Protocol connections. Connections can be compressed if the server and the client agree on a compression algorithm to use. By default, X Protocol announces support for the `deflate`, `lz4`, and `zstd` compression algorithms. You can disallow any of these algorithms by setting the new `mysqlx_compression_algorithms` system variable to include only the ones you permit. X Protocol always allows uncompressed connections if the client does not request compression during capability negotiation. Note that X Protocol's list of permitted compression algorithms operates independently of the list of compression algorithms announced by MySQL Server, and X Protocol does not fall back to using MySQL Server's compression settings. You can monitor the effects of message compression for X Protocol using new X Plugin status variables. (WL #9252, WL #13442)

- For multithreaded slaves (replication slaves on which `slave_parallel_workers` is set to a value greater than 0), setting `slave_preserve_commit_order=1` ensures that transactions are executed and committed on the slave in the same order as they appear in the slave's relay log, preserving the same transaction history on the slave as on the master. Previously, this setting required binary logging and slave update logging to be enabled on the slave, with the associated execution costs and disk space requirements. Now, `slave_preserve_commit_order=1` can be set on a slave with no binary log and no slave update logging. This enables you to preserve commit order on the slave, and avoid gaps in the sequence of transactions, without the overhead of binary logging.

  A limitation to preserving the commit order on the slave can occur if statement-based replication is in use, and both transactional and non-transactional storage engines participate in a non-XA transaction that is rolled back on the master. Normally, non-XA transactions that are rolled back on the master are not replicated to the slave, but in this particular situation, the transaction might be replicated to the slave. If this does happen, a multithreaded slave without binary logging does not handle the transaction rollback, so the commit order on the slave diverges from the relay log order of the transactions in that case. (WL #7846)

- The MySQL 8.0.18 release introduced the ability to specify a `PRIVILEGE_CHECKS_USER` account for a replication channel (using a `CHANGE MASTER TO` statement), against which MySQL makes privilege checks when replicated transactions are applied. The use of a `PRIVILEGE_CHECKS_USER` account helps secure a replication channel against the unauthorized or accidental use of privileged or unwanted operations. The use of row-based binary logging is strongly recommended when replication channels are secured with privilege checks.

  In MySQL 8.0.19, a new setting `REQUIRE_ROW_FORMAT` is added for replication channels, which makes the channel accept only row-based replication events. You can specify `REQUIRE_ROW_FORMAT` using a `CHANGE MASTER TO` statement to enforce row-based binary logging for a replication channel that is secured with privilege checks, or to increase the security of a channel that is not secured in this way. By allowing only row-based replication events, `REQUIRE_ROW_FORMAT` prevents the replication applier from taking actions such as creating temporary tables and executing `LOAD DATA INFILE` requests, which protects the replication channel against some known attack vectors. Row-based binary logging (`binlog_format=ROW`) must be used on the replication master when `REQUIRE_ROW_FORMAT` is set.

  Group Replication already requires row-based binary logging, so from MySQL 8.0.19, Group Replication channels are automatically created with `REQUIRE_ROW_FORMAT` set, and you cannot change the option for those channels. The setting is also applied to all Group Replication channels on upgrade.

`mysqlbinlog` has a new `--require-row-format` option, which enforces row-based replication events for `mysqlbinlog`'s output. The stream of events produced with this option would be accepted by a replication channel that is secured using the `REQUIRE_ROW_FORMAT` option. (WL #12968)

- MySQL uses delimiter strings when constructing tablespace names and file names for table partitions. A "`#p#`" delimiter string precedes partition names, and an "`#sp#`" delimiter string precedes subpartition names, as shown:

```
schema_name.table_name#p#partition_name#sp#subpartition_name
table_name#p#partition_name#sp#subpartition_name.ibd
```

  Historically, delimiter strings have been uppercase (`#P#` and `#SP#`) on case-sensitive file systems such as Linux, and lowercase (`#p#` and `#sp#`) on case-insensitive file systems such as Windows. To avoid issues when migrating data directories between case-sensitive and case-insensitive file systems, delimiter strings are now lowercase on all file systems. Uppercase delimiter strings are no longer used.

  Additionally, partition tablespace names and file names generated based on user-specified partition or subpartition names, which can be specified in uppercase or lowercase, are now generated (and stored internally) in lowercase regardless of the `lower_case_table_names` setting to ensure case-insensitivity. For example, if a table partition is created with the name `PART_1`, the tablespace name and file name are generated in lowercase:

```
schema_name.table_name#p#part_1
table_name#p#part_1.ibd
```

  During upgrade, MySQL now checks and modifies if necessary:

  - Partition file names on disk and in the data dictionary to ensure lowercase delimiters and partition names.
  - Partition metadata in the data dictionary for related issues introduced by previous bug fixes.
  - `InnoDB` statistics data for related issues introduced by previous bug fixes.

  During tablespace import operations, partition tablespace file names on disk are checked and modified if necessary to ensure lowercase delimiters and partition names. (WL #13352)

  References: See also: Bug #26925260, Bug #29823032, Bug #30012621, Bug #29426720, Bug #30024653.

- Support was added for efficient sampling of `InnoDB` data for the purpose of generating histogram statistics. The default sampling implementation used by MySQL when storage engines do not provide their own requires a full table scan, which is costly for large tables. The `InnoDB` sampling implementation improves sampling performance by avoiding full table scans. The `sampled_pages_read` and `sampled_pages_skipped` `INNODB_METRICS` counters can be used to monitor sampling of `InnoDB` data pages. See Histogram Statistics Analysis. (WL #8777)

## Bugs Fixed

- **Important Change:** Character set resolution has been changed for the following string functions:
  - `REPLACE(str, from_str, to_str)`
  - `SUBSTRING_INDEX(str, delim, count)`
  - `TRIM([{BOTH | LEADING | TRAILING} [remstr] FROM] str)`

Previously, character set information for all arguments to these functions was aggregated, which could lead to results that were not well formed. This also caused issues with `LPAD()`, which assumes that both input and output are well formed. Now each of the three listed functions always uses the character set employed by *`str`*, and converts all other arguments to this character set at execution time; if any such conversion fails, the function returns an error. (Bug #30114420)

References: This issue is a regression of: Bug #28197977.

- **Important Change:** Subquery materialization no longer requires strict matching of inner and outer types. Different types can now be materialized when one of the following conditions is true:
  - The inner type is numeric (since there is always a way to cast the outer type to a number)
  - The inner type is temporal (since there is always a way to cast the outer type to a temporal)
  - Both types are strings

  (Bug #13960580)

- **NDB Cluster:** Password masking was incomplete for some NDB logging options. (Bug #97335, Bug #30453137)

- **InnoDB:** Initialization of certain internal data structures at startup depend on internal variables derived from the `max_connections` setting. `InnoDB` failed to resize the internal data structures when the `max_connections` setting was modified after startup using `SET PERSIST`. (Bug #30628872)

- **InnoDB:** `os_file_get_parent_dir` warnings were encountered when compiling MySQL with GCC 9.2.0. (Bug #30499288, Bug #97466)

- **InnoDB:** An attempt to access a large object (LOB) value using a null reference raised an assertion failure. To prevent this issue form occurring, a check was added to determine if LOB references are null before they are accessed. (Bug #30499064)

- **InnoDB:** An assertion failure occurred after upgrading the data directory. Prepared XA transaction were still present, which prevented undo tablespaces from being upgraded. Undo tablespaces containing prepared transaction changes must remain active until all prepared XA transactions are committed or rolled back.

  Prepared XA transactions also prevented the completion of an explicit undo tablespace truncation operation after a restart. (Bug #30489497)

- **InnoDB:** Attempting to upgrade a MySQL 5.7 instance on Linux with uppercase table names (partitioned or otherwise) to MySQL 8.0 on macOS raised an assertion failure. Partition file format changes in MySQL 8.0 prevented migration of the data directory to a different platform, and the `lower_case_table_names` setting was changed at upgrade time, which can cause an upgrade failure. Instead of a failure occurring under these circumstances, an error is now reported. (Bug #30450968, Bug #30450979)

- **InnoDB:** On macOS, a failure occurred when attempting to upgrade a MySQL 5.7 instance with uppercase table names to MySQL 8.0. Uppercase table names were not normalized to lowercase. The following errors were reported: `Table is not found in InnoDB dictionary` and `Error in fixing SE data errors`. (Bug #30450944)

- **InnoDB:** On Windows, a failure occurred when attempting to upgrade a MySQL 5.7 instance with uppercase partitioned table names to MySQL 8.0. Opening the table returned a null pointer, which caused a segmentation fault when closing the table. (Bug #30450918)

- **InnoDB:** On Windows, a `mysqld` exception was raised when attempting to upgrade a MySQL 5.7 instance with uppercase partitioned table names to MySQL 8.0. (Bug #30447790)
- **InnoDB:** On Windows, a failure occurred when attempting to upgrade a MySQL 5.7 instance containing general tablespace defined with an uppercase name to MySQL 8.0. The following errors were reported: `Error in fixing SE data` and `Failed to Populate DD`. (Bug #30446798)
- **InnoDB:** Introduction of local minitransactions (mtrs) in LOB-related code resulted in an assertion failure during recovery. (Bug #30417719)
- **InnoDB:** A failure occurred when attempting to upgrade a MySQL 5.7 instance on Windows with uppercase partitioned table names to MySQL 8.0 on Linux. Partition file format changes in MySQL 8.0 prevented migration of the data directory to a different platform. Instead of a failure, an error is now reported. (Bug #30411118)
- **InnoDB:** Updating the same compressed LOB data repeatedly caused the tablespace file to increase in size. (Bug #30353812)
- **InnoDB:** When the `temptable_max_ram` limit was reached, the TempTable storage engine incorrectly reported an out-of-memory error instead of falling back to disk-based storage. (Bug #30314972, Bug #96893)
- **InnoDB:** After importing an encrypted table and restarting the server, the following error was returned when attempting to access the table: `ERROR 3185 (HY000): Can't find master key from keyring, please check in the server log if a keyring plugin is loaded and initialized successfully`. The tablespace key was not written to disk after it was encrypted with the destination master key. (Bug #30313734)
- **InnoDB:** The internal `InnoDB` `dict_create_foreign_constraints()` function that parsed SQL statements and performed foreign key related DDL checks was removed. The function became redundant with introduction of the data dictionary in MySQL 8.0 and the subsequent relocation of foreign key related DDL checks to the SQL layer.

  Removal of the `dict_create_foreign_constraints()` function also addressed the following foreign key issues:

  - Spaces around dots (“.”) in a fully qualified referenced table name were not permitted by the `InnoDB` parser.
  - Adding a foreign key and removing partitioning in the same `ALTER TABLE` statement was not permitted. The `InnoDB` parser did not detect that the new table version was no longer partitioned.
  - A foreign key constraint could not reference a table inside a schema named “AUX”. The function that parsed referenced table names did not recognize that special names such as AUX are encoded.
  - Conditional comments in foreign key definitions were ignored.

  Additionally, a check was added to the SQL layer to detect attempts to create multiple foreign keys of the same name on a table at an early stage in the execution of an `ALTER TABLE` statement. (Bug #30287895, Bug #22364336, Bug #28486106, Bug #28703793, Bug #16904122, Bug #92567, Bug #11754659, Bug #46293)
- **InnoDB:** A comparison function found two records to be equal when attempting to merge non-leaf pages of a spatial index. The function was unable to handle this unexpected condition, which resulted in a long semaphore wait and an eventual assertion failure. (Bug #30287668)
- **InnoDB:** A locally acquired latch required for freeing a large object (LOB) page could have caused a deadlock if a subsequent caller attempted to acquire a latch for the same page before the page

was freed. Similarly, a latch taken on a compressed or uncompressed LOB during a rollback related operation could have caused a deadlock due to a latching order issue. (Bug #30258536)

References: This issue is a regression of: Bug #29846292.

- **InnoDB:** A race condition between a purge thread that was purging a compressed LOB page and an update thread that is using a delete-marked record caused an assertion failure. (Bug #30197056)
- **InnoDB:** A tablespace import operation that failed due to the source and destination tables being defined with different `DATA DIRECTORY` clauses reported an insufficiently descriptive schema mismatch error. Moreover, if a `.cfg` file was not present, the same operation would raise an assertion failure. A more informative error message is now reported in both cases before the import operation is terminated due to the data directory mismatch. (Bug #30190199, Bug #30190227, Bug #20644698, Bug #76142)
- **InnoDB:** A purge operation failed when attempting to purge a LOB value larger than the buffer pool. (Bug #30183982)
- **InnoDB:** Update operations that moved externally stored LOB data to inline storage failed to mark the old LOB data as purgeable. (Bug #30178056, Bug #96466)
- **InnoDB:** Index key part sort order information was not stored to the `.cfg` metadata file used by `ALTER TABLE ... IMPORT TABLESPACE` operations. The index key part sort order was therefore assumed to be ascending, which is the default. As a result, records could be sorted in an unintended order if one table involved in the import operation is defined with a `DESC` index key part sort order and the other table is not. To address this issue, the `.cfg` file format was updated to include index key part sort order information. (Bug #30128418)
- **InnoDB:** Criteria used by the `btr_cur_will_modify_tree()` function, which detects whether a modifying record needs a modifying tree structure, was insufficient. (Bug #30113362)
- **InnoDB:** Startup was slow on instances with a large number of tables due the tablespace file scan that occurs at startup to retrieve space IDs. A multithreaded scan was only initiated if the number of tablespace files exceed 50,000, and three tablespace pages were read to retrieve a space ID. To improve startup times, additional threads are now allocated for the tablespace file scan, and only the first tablespace page is read to retrieve a space ID. If a space ID is not found on the first page of the tablespace, three pages are read to determine the space ID, as before. (Bug #30108154, Bug #96340)
- **InnoDB:** Startup failed on a case insensitive file system with an error indicating that multiple files were found for the same tablespace ID. A file path comparison did not recognize that `innodb_data_home_dir` and `datadir` paths were the same due to the paths having different lettercases. (Bug #30040815)
- **InnoDB:** A storage engine error occurred when accessing the `mysql.innodb_index_stats` and `mysql.innodb_table_stats` persistent optimizer statistics tables after upgrading a MySQL 8.0.13 instance on Linux with partitioned tables and a `lower_case_table_names=1` setting to MySQL 8.0.14 or MySQL 8.0.15. The persistent optimizer statistics tables contained duplicate entries. (Bug #30012621)

  References: This issue is a regression of: Bug #26925260.
- **InnoDB:** `CREATE TABLESPACE` failed with an error indicating that the tablespace already exists. The error was due to the failure of a preceding `CREATE TABLESPACE` operation where the DDL failed but related changes were not rolled back due to rollback being disabled prior to transaction commit. Rollback is now disabled after the transaction commits successfully. (Bug #29959193, Bug #95994)
- **InnoDB:** Changed pages belonging to imported tablespaces were not being tracked. (Bug #29917343)

- **InnoDB:** Renaming of full-text search auxiliary tables during upgrade failed due to a tablespace name conflict when upgrading from MySQL 5.7 to MySQL 8.0 on a case-insensitive file system. (Bug #29906115)
- **InnoDB:** Rollback of an `INSERT` operation that inserted a LOB value larger than a buffer pool caused a deadlock. (Bug #29846292, Bug #95572)
- **InnoDB:** A code regression was addressed by prohibiting unnecessary implicit to explicit secondary index lock conversions for session temporary tables. (Bug #29718243)
- **InnoDB:** A tablespace import operation raised an assertion when the cursor was positioned on a corrupted page while purging delete-marked records. Instead of asserting when encountering a corrupted page, the import operation is now terminated and an error is reported. (Bug #29454828, Bug #94541)
- **InnoDB:** Delete marked rows were able to acquire an external read lock before a partial rollback was completed. The external read lock prevented conversion of an implicit lock to an explicit lock during the partial rollback, causing an assertion failure. (Bug #29195848)
- **InnoDB:** After a server exit that occurred while an undo tablespace truncation operation was in progress, warning messages were printed at startup stating that doublewrite pages could not be restored for undo tablespace pages. The warning messages are no longer printed for undo tablespaces that are being truncated. (Bug #28590016)
- **InnoDB:** In read-only mode (`innodb_read_only=ON`), `SHOW CREATE TABLE` output did not include information about foreign key constraints. (Bug #21966795, Bug #78754)
- **Partitioning:** When upgrading a database with a subpartitioned table from MySQL 8.0.16 or lower and then executing `ALTER TABLE ADD COLUMN`, an assertion or error would occur. (Bug #30360695, Bug #97054)
- **Partitioning:** During upgrade of partitioned tables from MySQL 5.7 to 8.0, when a prefix key was used by the partitioning function, the prefix length was ignored, and the full column length was considered instead. Consequently, the table might incorrectly be rejected from being upgraded because its partition field length was found to be too large. (Bug #29941988, Bug #95921)
- **Partitioning:** `ALTER TABLE ... EXCHANGE PARTITION` could cause indexes to become corrupted. This was due to the fact that the server assumed that the order in which an index is created in a partitioned table is the same as that of the table which is not partitioned. This led to the wrong index data being exchanged. (Bug #29706669)
- **Replication:** An assertion was raised when privilege checks were carried out for a replication channel if the slave had more columns in the relevant table than the master. The check now references the number of columns in the event, rather than in the table definition. (Bug #30343310)
- **Replication:** Replication connection parameters that are held in the `mysql.slave_relay_log_info` table are now preserved in the event of a server crash or deliberate restart after issuing `RESET SLAVE` but before issuing `START SLAVE`. This action applies to the `PRIVILEGE_CHECKS_USER` account setting for replication privilege checks (introduced in MySQL 8.0.18) and the `REQUIRE_ROW_FORMAT` setting (introduced in MySQL 8.0.19). Note that if `relay_log_info_repository=FILE` is set on the server (which is not the default and is deprecated), replication connection parameters are not preserved in this situation. (Bug #30311908)
- **Replication:** When a replication channel is secured by specifying a `PRIVILEGE_CHECKS_USER` account, which should not have ACL privileges, a `GRANT` statement that is replicated to the channel causes the replication applier to stop. In this situation, the behavior was correct but an assertion was being raised. The assertion has now been removed. (Bug #30273684)

- **Replication:** For multithreaded replication slaves, setting `slave_preserve_commit_order=1` now preserves the order of statements with an `IF EXISTS` clause when the object concerned does not exist. Previously, these updates might have committed before transactions that preceded them in the relay log, which might have resulted in gaps in the sequence of transactions that have been executed from the slave's relay log. (Bug #30262096)

- **Replication:** When privilege checks were carried out for a replication channel, the permissions required for setting the session value of the `sql_require_primary_key` system variable were not being checked. The check is now carried out. (Bug #30254917)

- **Replication:** A memory leak could occur when a failed replication group member tried to rejoin a minority group and was disallowed from doing so. (Bug #30162547, Bug #96471)

- **Replication:** When a group member rejoins a replication group, it begins the distributed recovery process by checking the relay log for its `group_replication_applier` channel for any transactions that it already received from the group, and applying these. The joining member then initiates state transfer from an existing online member, which might begin with a remote cloning operation. Previously, the `group_replication_applier` channel was not explicitly stopped when a remote cloning operation was started, so it was possible that the applier might still be applying existing transactions at that time, which might lead to errors. The `group_replication_applier` channel is now stopped before a remote cloning operation is requested, and restarted when the distributed recovery process moves on to state transfer from a donor's binary log. (Bug #30152028, Bug #96447)

- **Replication:** If `STOP GROUP_REPLICATION` was issued while the member's XCom port was blocked, the XCom thread hung and the shutdown did not complete. XCom is now terminated in this situation. (Bug #30139794)

- **Replication:** The slave status logs `mysql.slave_relay_log_info` (the relay log info log) and `mysql.slave_worker_info` (the slave worker log) are now copied from the donor to the recipient during a local or remote cloning operation. The slave status logs hold information that can be used to resume replication correctly after the cloning operation, including the relay log position from which to restart replication, the `PRIVILEGE_CHECKS_USER` account setting, and the new `REQUIRE_ROW_FORMAT` setting. Note that the relay logs themselves are not copied from the donor to the recipient, only the information about them that is held in these tables. Also note that if `relay_log_info_repository=FILE` is set on the server (which is not the default and is deprecated), the slave status logs are not cloned; they are only cloned if `TABLE` is set.

  Before this patch, the following replication-related behaviors occurred on a replication slave that had been provisioned by a cloning operation:

  - The default replication channel would fail to start if it was the only channel on the slave, because it was considered to be not initialized due to the missing relay log information.

  - Any `PRIVILEGE_CHECKS_USER` account setting that had been applied to replication channels on the donor was absent and had to be respecified.

  - Replication channels that used GTID auto-positioning (as specified by the `MASTER_AUTO_POSITION` option on the `CHANGE MASTER TO` statement) were able to resume replication automatically.

  - Replication channels that used binary log file position based replication (as specified by the `MASTER_LOG_FILE` and `MASTER_LOG_POS` options on the `CHANGE MASTER TO` statement) had to have the `MASTER_LOG_FILE` and `MASTER_LOG_POS` options reapplied manually before restarting replication in order to resume correctly. If the channels were configured to start replication automatically at server startup, without the options reapplied they would attempt to start replication from the beginning. They were therefore likely to attempt to replicate data that had already been

copied to the slave by the cloning operation, causing replication to stop and possibly corrupting the data on the slave.

With this change, the following replication-related behaviors now occur on a replication slave that has been provisioned by a cloning operation:

- The default replication channel can now always start after the cloning operation if it is configured to do so.
- All channels now have the donor's `PRIVILEGE_CHECKS_USER` account setting and `REQUIRE_ROW_FORMAT` setting.
- Replication channels that use GTID auto-positioning (as specified by the `MASTER_AUTO_POSITION` option on the `CHANGE MASTER TO` statement) are still able to resume replication automatically. For Group Replication channels, which use GTID auto-positioning, an internal equivalent of the `RESET MASTER` statement is now used to ensure that replication resumes optimally.
- Replication channels that use binary log file position based replication now have the correct `MASTER_LOG_FILE` and `MASTER_LOG_POS` options in place after cloning. Because the relay logs themselves are not cloned, these channels now attempt to carry out the relay log recovery process, using the cloned relay log information, before restarting replication. For a single-threaded slave (`slave_parallel_workers` is set to 0), relay log recovery should succeed in the absence of any other issues, enabling the channel to resume replication correctly. For a multithreaded slave (`slave_parallel_workers` is greater than 0), relay log recovery is likely to fail because it cannot usually be completed automatically, but an informative error message is issued, and the data will not be corrupted.

(Bug #29995256, Bug #30510766)

- **Replication:** An internal deadlock could occur on a multi-threaded replication slave when the `relay_log_space_limit` system variable was set to limit the size of relay logs on the slave, and the coordinator thread acquired locks related to this limit and to the end position of the log. (Bug #29842426)
- **Replication:** If a replication slave was set up using a `CHANGE MASTER TO` statement that did not specify the master log file name and master log position, then shut down before `START SLAVE` was issued, then restarted with the option `--relay-log-recovery` set, replication did not start. This happened because the receiver thread had not been started before relay log recovery was attempted, so no log rotation event was available in the relay log to provide the master log file name and master log position. In this situation, the slave now skips relay log recovery and logs a warning, then proceeds to start replication. (Bug #28996606, Bug #93397)
- **Group Replication:** When a member is joining or rejoining a replication group, if Group Replication detects an error in the distributed recovery process (during which the joining member receives state transfer from an existing online member), it automatically switches over to a new donor, and retries the state transfer. The number of times the joining member retries before giving up is set by the `group_replication_recovery_retry_count` system variable. The Performance Schema table `replication_applier_status_by_worker` displays the error that caused the last retry. Previously, this error was only shown if the group member was configured with parallel replication applier threads (as set by the `slave_parallel_workers` system variable). If the group member was configured with a single applier thread, the error was cleared after each retry by an internal `RESET SLAVE` operation, so it could not be viewed. This was also the case for the output of the `SHOW SLAVE STATUS` command whether there were single or multiple applier threads. The `RESET SLAVE` operation is now no longer carried out after retrying distributed recovery, so the error that caused the last retry can always be viewed. (Bug #30517160, Bug #30517172, Bug #97540)

- **Group Replication:** When a replication group member leaves a group, either because `STOP GROUP_REPLICATION` was issued or due to an error, Group Replication now stops the binary log dump thread so that the former group member cannot send unwanted binary log data to the members that have remained in the group. (Bug #30315614)

- **Group Replication:** When Group Replication was started following either provisioning with a cloning operation, execution of `RESET MASTER`, or removal of a partial transaction from the relay log, `RESET SLAVE ALL` was used internally to clear any unwanted state on the server. However, in MySQL 8.0.18, this caused any `PRIVILEGE_CHECKS_USER` account that was specified for a Group Replication channel to be removed. `RESET SLAVE` is now used instead, which does not remove the account. (Bug #30262225)

- **Group Replication:** When Group Replication is running in single-primary mode, and a new primary server is elected, the messages logged at this time now provide the newly elected primary server's `gtid_executed` set, and the set of GTIDs retrieved by the replication applier. (Bug #30049310)

- **Group Replication:** If a replication group member stops unexpectedly and is immediately restarted (for example, because it was started with `mysqld_safe`), it automatically attempts to rejoin the group if `group_replication_start_on_boot=on` is set. Previously, if the restart and rejoin attempt took place before the member's previous incarnation had been expelled from the group, the member could not rejoin. Now in this scenario, Group Replication automatically uses a Group Communication System (GCS) feature to retry the rejoin attempt for the member 10 times, with a 5-second interval between each retry. This should cover most cases and allow enough time for the previous incarnation to be expelled from the group, letting the member rejoin. Note that if the `group_replication_member_expel_timeout` system variable is set to specify a longer waiting period before the member is expelled, the automatic rejoin attempts might still not succeed. (Bug #29801773)

- **macOS:** On macOS, configuring MySQL with `-DWITH_SSL=system` caused `mysql_config` output to incorrectly include internal `CMake` names for the static SSL libraries. (Bug #30541879, Bug #97632)

- **macOS:** Builds on macOS with Ninja could fail with an error trying to create a symbolic link multiple times. (Bug #30368985)

- **Microsoft Windows; JSON:** On Windows platforms, memory used for a multi-valued index was not released after the table containing it was dropped. (Bug #30227756)

- **Microsoft Windows:** On Windows, `-DWITH_SSL=system` failed to find the installed OpenSSL headers if Strawberry Perl was installed. (Bug #30359287)

- **Microsoft Windows:** On Windows, the `-DWITH_SSL=system` option did not work if the path name leading to the system OpenSSL libraries contained a space. This is now handled. Also, `-DWITH_SSL=yes` is treated like `-DWITH_SSL=system`, as on other platforms. (Bug #30261942, Bug #96739)

- **Microsoft Windows:** MSVC 2019 produced garbled source file names for compilation errors. A workaround in the `CMake` configuration was implemented to correct for this. (Bug #30255096, Bug #96720)

- **JSON:** Updating a value in a `JSON` column by replacing a character string element with a binary string containing the same byte sequence as the `utf8mb4` representation of the character string had no effect.

  The root cause of this issue was a change in the behavior of comparisons between JSON strings and JSON opaque values introduced by the implementation of multi-valued indexes in MySQL 8.0.17, previous to which, JSON strings and JSON opaque values were never considered equal. After the change, they were considered equal if their binary data matched.

An analysis of this change showed that it was not needed; in addition, the new behavior conflicted with the existing documentation for comparisons of JSON values. This issue is fixed by restoring the original behavior. (Bug #30348554)

- **JSON:** A view that used `JSON_TABLE()` did not preserve the character set in which JSON path arguments were encoded. This meant that, if the view was evaluated with a different character set in effect from the one in which it was defined, it could produce wrong results. This is fixed by ensuring that `JSON_TABLE()` preserves the original character set in such cases. (Bug #30310265)

- **JSON:** Adding a functional index on a JSON column changed the collation used for comparing strings, causing the result returned by the same query selecting the column to differ from that obtained without the index. (Bug #29723353)

- **JSON:** If the first argument to `JSON_TABLE()` was `const` during the execution of a stored procedure, but not during preparation, it was not re-evaluated when a statement was subsequently executed again, causing an empty result to be returned each time following the first execution of the procedure. (Bug #97097, Bug #30382156)

- **JSON:** In some cases, such as when a query uses `FORCE INDEX`, the cost of reading the table is `DBL_MAX`; this was rounded up to `2e308` when printed, which is too large for the JSON parser, so that it was not possible to extract parts of the optimizer trace using a query such as `SELECT JSON_EXTRACT(trace, '$**.table_scan') FROM INFORMATION_SCHEMA.OPTIMIZER_TRACE`. Now in such cases, values greater than `1.5e308` are rounded down and printed as `1e308` instead. (Bug #96751, Bug #30226767)

- After upgrading from MySQL 5.7 to MySQL 8.0, a `CLONE INSTANCE` operation failed with the following error: `ERROR 3862 (HY000): Clone Donor Error: 1016 : Can't open file: './undo001'.` The upgrade process left behind orphaned in-memory undo tablespaces. Thanks to Satya Bodapati for the contribution. (Bug #30602218, Bug #97784, Bug #30239255, Bug #96637)

- The MySQL optimizer's hash join algorithm uses the join buffer to store intermediate results. If this buffer overflows, the server uses a spill-to-disk algorithm, which writes one of the hash join operands to a temporary file, to handle this gracefully. If one of the operands was a table that was a member of a pushed join operation, this strategy conflicted with the pushed join requirement for all child result rows to use nested-loop reads whenever one of their pushed join ancestors was the current row in the join evaluation, which could in some cases result in incorrect query results being returned. (Bug #30573733)

- Access to the `INFORMATION_SCHEMA.VIEWS` table was not properly restricted to the correct user. (Bug #30542333)

- When creating hash values used for lookups during a hash join, the server did not respect the `PAD SPACE` attribute, meaning that `'foo'` and `'foo '` did not match when using a `PAD SPACE` collation. This is fixed by padding all strings up to the same length as the longest possible string, where the longest possible string is deduced from the data type length specifier *`N`* in `CHAR(`*`N`*`)` or `VARCHAR(`*`N`*`)`. (Bug #30535541)

- When retrieving large result sets containing `DECIMAL` columns from a secondary engine, conversion of the column values to strings for transport over the text protocol acted as a bottleneck. The performance of the functions responsible for such conversions has been improved in some cases by as much as 50%, as reflected in internal testing. (Bug #30528427)

- When the `FORMAT_PICO_TIME()` function was invoked to process several rows, once a `NULL` argument was found in a row, every result after that was set to `NULL`. (Bug #30525561)

- When a Performance Schema event was timed, the event duration reported in `events_`*`xxx`* tables could be `NULL` instead of 0 for events where the timer start and end values are equal. (Bug #30525560)

- Adding a `LIMIT` clause to a parenthesized query suppressed locking clauses within the parentheses. For example, this query would not lock the table:

  ```
  (SELECT ... FOR UPDATE) LIMIT ...;
  ```

  Adding a `LIMIT` clause outside of a parenthesized query is intended to override a `LIMIT` clause within the parentheses. However, the outer `LIMIT` suppressed `ORDER BY` within the parentheses as well. For example, for this query, the `ORDER BY` was suppressed:

  ```
  (SELECT ... ORDER BY ... LIMIT a) LIMIT b;
  ```

  Now inner locking and `ORDER BY` clauses are not suppressed by an outer `LIMIT` clause. (Bug #30521098, Bug #30521803)

- When optimizer extracts conditions on constant tables for early evaluation, it does not include `WHERE` conditions that are expensive to evaluate, including conditions involving stored functions. When the extracted condition evaluated to true because it involved only const tables, the entire `WHERE` condition was incorrectly removed. Now in such cases, a check for expensive conditions is performed prior to any removal of the `WHERE` condition. (Bug #30520714)

- When a lateral materialized derived table used `DISTINCT`, the derived table was not rematerialized for each outer row as expected. (Bug #30515233)

- `EXPLAIN ANALYZE` did not work correctly with a common table expression using `WITH RECURSIVE`. (Bug #30509580)

- The GNU `gold` loader could cause memory exhaustion on some platforms. Now it is used by default only on Intel 64-bit platforms. (Bug #30504760, Bug #96698)

- Some Linux platforms experienced high overhead with `EXPLAIN ANALYZE` due to use of a system call by `libstdc++` instead of `clock_gettime()`. (Bug #30483025)

- On Solaris 11.4, the LDAP authentication plugins could not be built. (Bug #30482553)

- Queries that used the `MEMBER OF()` operator were not always handled correctly. (Bug #30477993)

- Boost compilation failed under Visual Studio due to a Boost workaround for a VC++ 2013 bug that has since been fixed. The workaround is now patched for Boost compilation with MySQL. (Bug #30474056, Bug #97391)

- When retrieving large result sets containing many integers from a secondary engine, conversion of the integers to strings for sending over the text protocol could act as a bottleneck. To avoid this problem, the performance of internal functions performing such conversions has been improved. (Bug #30472888)

- Docker packages were missing the LDAP authentication plugins. (Bug #30465247)

- Corrected a typo in a `mysys/my_handler_errors.h` error message. Thanks to Nikolai Kostrigin for the contribution. (Bug #30462329, Bug #97361)

- A GTID table update while `innodb_force_recovery` was enabled caused a debug assertion failure. (Bug #30449531, Bug #97312)

- MySQL failed to compile against Protobuf 3.10. (Bug #30428543, Bug #97246)

- Buffered log lines during system startup could be lost. (Bug #30422941, Bug #97225)

- If the `mysql.user` system table was renamed, the server could exit. (Bug #30418070)

- Revoking a role specified with no host name could cause a server exit. (Bug #30416389)

- When determining whether to pull out a semijoin table when other tables inside the semijoin depended on this table, only those semijoin tables which were base tables were considered; those in nested joins were ignored. (Bug #30406241)

  References: See also: Bug #12714094, Bug #11752543, Bug #43768.

- The AppArmor profile on Ubuntu platforms was not able to read the OpenSSL configuration. (Bug #30375723)

- Some Fedora 30 packages had missing obsoletes information that could cause problems upgrading an existing MySQL installation. (Bug #30348549, Bug #96969)

- Altering only the default encryption in an `ALTER SCHEMA` statement caused the schema default character set and collation to be reset to the system defaults. (Bug #30344462, Bug #96994)

- Columns declared with both `AUTO_INCREMENT` and `DEFAULT` value expressions (a nonpermitted combination) could raise an assertion or cause a server exit. (Bug #30331053)

- `SHOW GRANTS` for an anonymous user could result in a server exit under some conditions. (Bug #30329114)

- `GREATEST()` and `LEAST()` did not always handle time values correctly. (Bug #30326848)

  References: This issue is a regression of: Bug #25123839.

- The list of subpartitions in partition objects was not serialized and therefore not included in serialized dictionary information (SDI). To address this issue, support was added for serialization and deserialization of subpartition dictionary information. The patch for this bug also includes minor SDI code refactoring and format changes. Due to the format changes, the SDI version number was incremented. (Bug #30326020, Bug #96943)

- Following execution of `ANALYZE TABLE`, the optimizer trace for a given query differed when another query was executed previously to it, but also after the `ANALYZE TABLE`. (Bug #30321546)

- `innodb_buffer_pool_instances` was not initialized correctly at server startup if it had been set using `SET PERSIST` or `PERSIST_ONLY`. (Bug #30318828)

- A low `max_allowed_packet` value caused the following error: `ERROR 1153 (08S01) at line 1: Got a packet bigger than 'max_allowed_packet' bytes`. The error message was revised to indicate the minimum required `max_allowed_packet` value for cloning operations. (Bug #30315486, Bug #96891)

- An assertion could be raised when server code tried to send to clients an error code intended to be written to the error log. These instances are fixed by sending a code intended to be sent to clients. (Bug #30312874)

- `CREATE VIEW` did not always succeed when the body of the view definition contained a join and multiple subselects. (Bug #30309982)

  References: This issue is a regression of: Bug #25466100.

- Dependency information for SLES 12 RPM packages was incorrect, causing MySQL installation failure. (Bug #30308305)

- When restoring `GEOMETRY` data from hash join chunk files to a `GEOMETRY` column, the server did not copy the data to the column, but instead stored a pointer to the data, which resided in a temporary buffer, meaning that the `GEOMETRY` column pointed to random data as soon as this buffer was reused.

Now, the server always copies the data from this buffer into the `GEOMETRY` column when executing a hash join. (Bug #30306279)

- Some `ALTER TABLE` operations using the `COPY` algorithm did not handle columns with expression default values properly. (Bug #30302907, Bug #96864)
- The `CONV()` function did not always handle returning the proper number of characters correctly. (Bug #30301543)
- Parser recursion checks were insufficient to prevent stack overflow. (Bug #30299881)
- The removal of a subquery because the condition in which it occurred was always false was expected to be performed during resolution, but when the subquery did not involve any tables, the server executed it while resolving it. This resulted in the failure of a subsequent check to confirm that the subquery was only being resolved and not yet optimized. Now in such cases, the server also checks to see whether the subquery was already executed. (Bug #30273827)
- For debug builds, attempts to add to an empty temporary table a column with an expression default that was not valid raised an assertion. (Bug #30271792)
- Construction of the iterator tree may yield a non-hierarchical structure; this can happen when, for example, `b` and `c` from `a LEFT JOIN b LEFT JOIN c` also make up the right side of a semijoin. The iterator executor solves this by adding a weedout on top of the entire query, which means that is also necessary to iterators interacting with row IDs that they need to store and restore them. This was not done in all such cases, causing wrong results. Now the addition of a top-level weedout is always communicated to the iterators as soon as it is known that this is being done, before any affected iterators are constructed. (Bug #30267889)
- Foreign key-handling code duplication between the SQL layer and the data dictionary was eliminated. A side effect is that some error messages now are more informative and clear. (Bug #30267236, Bug #96765)
- During startup, the server could handle incorrect option values for persisted variables improperly, resulting in a server exit. (Bug #30263773)
- In some queries involving materialized semijoins, when using the iterator executor, conditions were evaluated outside the materialization, causing inefficient query plans to be used and sometimes also producing wrong results. (Bug #30250091)
- `ALTER TABLE` statements that renamed a column used in `CHECK` constraints could result in an incorrect error message. (Bug #30239721)
- For `SELECT` statements, an `INTO` *`var_name`* clause prior to a locking clause is legal but the parser rejected it. (Bug #30237291, Bug #96677)
- `FLUSH TABLES WITH READ LOCK` caused a deadlock when a `LOCK INSTANCE FOR BACKUP` statement was previously executed within the same session and there was a concurrent `ALTER DATABASE` statement running in another session against the same database specified (implicitly or explicitly) for the `FLUSH TABLES WITH READ LOCK` statement. (Bug #30226264)
- Slow query logging could result in a server exit for connections that did not use the classic client/server protocol. (Bug #30221187)
- A statement that added a foreign key without an explicit name failed when re-executed as a prepared statement or in a stored program with an unwarranted duplicate foreign key name error. (Bug #30214965, Bug #96611)

  References: This issue is a regression of: Bug #30171959.

- With multiple sessions executing concurrent `INSERT ... ON DUPLICATE KEY UPDATE` statements into a table with an `AUTO_INCREMENT` column but not specifying the `AUTO_INCREMENT` value, inserts could fail with a unique index violation. (Bug #30194841, Bug #96578)

- Client programs could load authentication plugins from outside the plugin library. (Bug #30191834, Bug #30644258)

- When switching between table scans and index lookups, `AlternativeIterator` did not reset the handler, which could lead to assertion failures. (Bug #30191394)

- Setting `open_files_limit` to a large value, or setting it when the operating system rlimit had a value that was large but not equal to `RLIM_INF` could cause the server to run out of memory. As part of this fix, the server now caps the effective `open_files_limit` value to the the maximum unsigned integer value. (Bug #30183865, Bug #96525)

- References to fully qualified `INFORMATION_SCHEMA` tables could fail depending on the lettercase in which `INFORMATION_SCHEMA` was specified. (Bug #30158484)

- Slow queries with an execution time greater than 35 days could cause corruption of the `mysql.slow_log` system table requiring a `REPAIR TABLE` operation. (Bug #30113119, Bug #96373)

- MySQL did not support sending systemd notification messages to a socket specified using the `NOTIFY_SOCKET` environment variable, if the variable named an abstract namespace socket. (Bug #30102279)

- Using `SET PERSIST_ONLY` to set a boolean system variable to a numeric value resulted in the server being unable to restart. (Bug #30094645, Bug #30298191, Bug #96848)

- A fix for a previous issue combined two `TABLE_LIST` constructors in an unfortunate way. One of these created a `TABLE_LIST` object from a `TABLE` object representing a temporary table. Previously, the table name was made the same as the alias; this was changed to copying the name from the `TABLE` object. Due to the fact that, for a temporary table, the table name is a file path, it was possible to exceed the limit for `MDL_key` names, leading to a failed assertion. Fixed by reintroducing dedicated constructors which behave in the manner that they did prior to the fix. (Bug #30083125)

  References: This issue is a regression of: Bug #27482976.

- For `UNIX_TIMESTAMP()` errors occurring within stored functions, the number of fractional seconds for subsequent function invocations could be incorrect. (Bug #30034972, Bug #96166)

- When a common table expression contained a nondeterministic expression (such one that used `RAND()`) and the common table expression was referenced more than once in the outer query, it was merged in some cases. This caused the common table expression to return a different result for each reference. Now in such cases, the common table expression is not merged, but rather is materialized instead. (Bug #30026353)

- In debug build of MySQL started on Linux with a `lower_case_table_names=1` setting, discarding a tablespace for a partitioned table after an in-place upgrade from MySQL 8.0.16 caused a serious error. The partition tablespace name stored in the data dictionary was invalid, and the metadata lock key prepared for the partition tablespace in MySQL 8.0.17 did not match the key stored in the `mysql.tablespaces` table. (Bug #30024653)

- `KILL QUERY` could kill the statement subsequent to the one intended. (Bug #29969769)

- With `lower_case_table_names=2`, `SHOW TABLES` could fail to display tables with uppercase names. (Bug #29957361)

- The error message reported for attempts to upgrade tables with invalid expressions for generated columns did not provide sufficient information. The error message now includes the generated column name and the expression used to create the generated column. (Bug #29941887, Bug #95918)
- Attempting to display an unresolvable view could result in a server exit rather than an error. (Bug #29939279)
- Incorrect checking of temporal literals for `CREATE TABLE` statements could lead to a server exit. (Bug #29906966, Bug #95794)
- Attempting to spawn a thread for a parallel read operation while system resources were temporary unavailable raised system error. (Bug #29874480)
- Writing unexpected values to the `mysql.global_grants` system table could cause a server exit. (Bug #29873343)
- The `LAST_EXECUTED` value in the `INFORMATION_SCHEMA.EVENTS` table was incorrectly reported in UTC, not in the event time zone. (Bug #29871530, Bug #95649)
- With `keyring_encrypted_file_password` set on the command line at server startup, the password value could be visible to system utilities. (Bug #29848634)
- Changing the `lower_case_table_names` setting when upgrading from MySQL 5.7 to MySQL 8.0 could cause a failure due to a schema or table name lettercase mismatch. If `lower_case_table_names=1`, table and schema names are now checked by the upgrade process to ensure that all characters are lowercase. If table or schema names are found to contain uppercase characters, the upgrade process fails with an error. For related information, see Preparing Your Installation for Upgrade. (Bug #29842749, Bug #95559)
- With a `LOCK TABLES` statement in effect, a metadata change for the locked table could cause Performance Schema or `SHOW` queries for session variables to hang in the `opening_tables` state. (Bug #29836204, Bug #92387)
- A `SELECT` using a `WHERE` condition of the form *`A AND (B OR C [OR ...])`* resulting in an impossible range led to an unplanned exit of the server. (Bug #29770705)
- For JSON-format audit logging, the `id` field now may contain values larger than 65535. Previously, with heaving logging activity, more than 65536 queries per second could be executed, exceeding the 16 bits permitted for `id` values. (Bug #29661920)
- An incomplete connection packet could cause clients not to properly initialize the authentication plugin name. (Bug #29630767)
- Out-of-memory errors from the parser could be ignored, resulting in a server exit. (Bug #29614521)
- On Linux, an assertion could be raised when the Performance Schema file instrumentation was disabled and re-enabled. (Bug #29607570)
- For a column defined as a `PRIMARY KEY` in a `CREATE TABLE` statement, a default value given as an expression was ignored. (Bug #29596969, Bug #94668)
- The `TABLE_ENCRYPTION_ADMIN` privilege, added in MySQL 8.0.16, was incorrectly granted to the system-defined `mysql.session` user during upgrade. (Bug #29596053, Bug #94888)
- For connections encrypted with OpenSSL, network I/O at the socket level was not reported by the Performance Schema. Also, network I/O performed while the server was in an `IDLE` state was not reported by the Performance Schema. (Bug #29205129, Bug #30535558, Bug #97600)

- When a query used a subquery that was merged into the outer query block (due to a semijoin transformation or merge of a derived table), and the subquery itself contained a subquery with an aggregate function with an aggregation query block that differed from its base query block, the query could sometimes fail to return any rows unless executed a second time or preceded with `FLUSH TABLES`. This was because, when merging, the information regarded tables used and the aggregation information for the aggregate function was not updated properly. In the case which raised this bug report, this meant that the comparison operation containing a scalar subquery was regarded as const-for-execution and therefore the range optimizer attempted to evaluate it, and the scalar subquery contained a `MIN()` function referring to an outer reference which had not yet been read. Thus, when the aggregator object was populated, it was based on uninitialized data, leading to unpredictable results. (Bug #28941154)

- Changing the `mandatory_roles` system variable could cause `SHOW GRANTS` in concurrent sessions to produce incorrect results. (Bug #28699403)

- Failure of `keyring_aws` initialization caused failure of SSL socket initialization. (Bug #28591098)

- Under certain conditions, enabling the `read_only` or `super_read_only` system variable did not block concurrent DDL statements executed by users without the `SUPER` privilege. (Bug #28438114, Bug #91852)

- The current `GROUP BY` plan is improved so that every gap attribute is allowed to have a disjunction of equality predicates. Predicates from different attributes must still be conjunctive to each other in order to take advantage of this enhancement.

  Our thanks to Facebook for this contribution. (Bug #28056998, Bug #15947433, WL #13066)

- In some cases, `BIGINT` arguments to the `FLOOR()` and `CEILING()` functions were resolved as the wrong type. (Bug #27125612)

- `mysqlpump` exits rather than dumping databases that contain an invalid view, by design, but it also failed if an invalid view existed but was not in any of the databases to be dumped. (Bug #27096081)

- Foreign key information is now retrieved from the data dictionary, not from `InnoDB`. (Bug #25583288)

- Foreign key definitions used in `CREATE TABLE` and `ALTER TABLE` statements for `InnoDB` tables were ignored if the statements were wrapped in conditional comments (such as `/*!50101 ... */` or `/*! ... */`). (Bug #21919887, Bug #78631)

- The `--log-raw` option is now available at runtime as the `log_raw` system variable. The system variable is set at startup to the option value, and may be set at runtime to change password masking behavior. (Bug #16636373, Bug #68936)

- `EXPLAIN ANALYZE` did not execute subqueries in the `SELECT` list, and thus did not take them into account in its calculations of time or cost. (Bug #97296, Bug #30444266)

- An inner scalar subquery containing an outer reference did not return the same result using a nested set of `SELECT` expressions on the right hand side as when using a single `SELECT` that was equivalent. (Bug #97063, Bug #30381092)

- A materialized subquery could yield different results depending on whether it used an index. (Bug #96823, Bug #30289052)

- When a query terminated due to exceeding the time specified using the `MAX_EXECUTION_TIME` hint, the error produced differed depending on the stage of the query. In particular, if the query terminated during a filesort, the error raised was `ER_FILSORT_ABORT`, even though in such cases the query should always exit with `ER_QUERY_TIMEOUT`. This made it unnecessarily difficult to trap such errors and to handle them correctly.

This fix removes the error codes `ER_FILSORT_ABORT` and `ER_FILESORT_TERMINATED`. (Bug #96537, Bug #30186874)

- If a stored procedure had a parameter named **member** or **array**, and it had been defined without quoting the parameter names, the database in which it was defined could not be upgraded to 8.0.17 or 8.0.18. (Bug #96288, Bug #30084237)

  References: See also: Bug #96350, Bug #30103640.

- When a function such as `COALESCE()` or `IFNULL()` was passed a `BIGINT` column value, casting a negative return value from this function to `UNSIGNED` unexpectedly yielded zero.

  Our thanks to Oleksandr Peresypkin for this contribution. (Bug #95954, Bug #29952066)

- `EXPLAIN` output showed `Select tables optimized away` for a query using `MAX()` on an indexed column, but if `MAX()` on the same column was called in a user function, it showed `Using index` instead. (Bug #94862, Bug #29596977)

# Changes in MySQL 8.0.18 (2019-10-14)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.


- Account Management Notes
- Compilation Notes
- Configuration Notes
- Connection Management Notes
- Deprecation and Removal Notes
- Keyring Notes
- Logging Notes
- Optimizer Notes
- Packaging Notes
- Pluggable Authentication
- Security Notes
- Spatial Data Support
- sys Schema Notes
- Test Suite Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed

## Account Management Notes

- The `CREATE USER`, `ALTER USER`, and `SET PASSWORD` statements now have the capability of generating random passwords for user accounts, as an alternative to requiring explicit administrator-specified literal passwords. See Password Management. (WL #11772)

## Compilation Notes

- **Incompatible Change:** The `my_ulonglong` type is no longer used in MySQL source code. Any third-party code that used this type should use the `uint64_t` C type instead. Also, it is possible that `printf()` format strings will need adjustment if used to print `my_ulonglong` variables. (Bug #29453827)
- For building MySQL 8.0, the minimum required version of Protobuf is now 3.0 rather than 2.5. (Bug #30141272)
- The minimum version of the Boost library for server builds is now 1.70.0. (Bug #29639344)
- A `DBUG_TRACE` macro is available to assist in writing debug code. It is a convenience that replaces pairs of enter/leave macros. For example, instead of writing this:

  ```
  void foo() {
    DBUG_ENTER("foo");
    bar();
    DBUG_VOID_RETURN;
  }
  ```

  Write this instead:

  ```
  void foo() {
    DBUG_TRACE;
    bar();
  }
  ```

  (Bug #29589102)

## Configuration Notes

- `CMake` now enables use of `fastcov` if it is available. `fastcov` is faster than `lcov` or `gcov`. This requires GCC and `gcov` versions of 9 or higher. (Bug #30011512)
- The `DISABLE_SHARED` `CMake` option was unused and has been removed. (Bug #29971049, Bug #96027)
- The `CMake` code to find Protobuf executables now works on platforms that split these into multiple packages. (Bug #29953773)
- The new `ADD_GDB_INDEX` `CMake` option determines whether to enable generation of a `.gdb_index` section in binaries, which makes loading them in a debugger faster. The option is disabled by default. It has no effect if a linker other than `lld` or GNU `gold` is used. (Bug #29925009, Bug #95857)
- For the `INSTALL_LAYOUT` `CMake` option, the `SLES` and `WIN` option values were not used and have been removed. (Bug #29871520, Bug #95654)
- The `max_prepared_stmt_count` system variable maximum value has been increased from 1 million (1,048,576) to 4 million (4,194,304). The default value remains unchanged at 16,382. (WL #13342)
- MySQL 8.0 no longer supports building using wolfSSL. All MySQL builds now use OpenSSL. (WL #13290)

- The RE2 library is no longer used by MySQL. The library is no longer bundled with source distributions and the `WITH_RE2` `CMake` option is obsolete.

## Connection Management Notes

- MySQL now provides more control over the use of compression to minimize the number of bytes sent over connections to the server. Previously, a given connection was either uncompressed or used the `zlib` compression algorithm. Now, it is also possible to use the `zstd` algorithm (zstd 1.3), and to select a compression level for `zstd` connections. The permitted compression algorithms can be configured on the server side, as well as on the connection-origination side for connections by client programs and by servers participating in master/slave replication or Group Replication. For more information, see Connection Compression Control.

  Connection compression using the `zstd` algorithm requires that the server be built with `zstd` library support. The new `WITH_ZSTD` `CMake` option indicates whether to use the bundled or system `zstd` library.

  Legacy compression-control parameters, such as the `--compress` client option, are deprecated and will be removed in a future MySQL version.

  Thanks to Facebook for the contribution on which some of this work was based. (WL #12039, WL #12475)

## Deprecation and Removal Notes

- Use of the `MYSQL_PWD` environment variable to specify a MySQL password is considered insecure because its value may be visible to other system users. `MYSQL_PWD` is now deprecated and will be removed in a future MySQL version. (WL #13449)

- The `relay_log_info_file` system variable and `--master-info-file` option are now deprecated and will be removed in a future MySQL version. These were used to specify the name of the relay log info log and master info log when `relay_log_info_repository=FILE` and `master_info_repository=FILE` were set, but those settings have been deprecated. The use of files for the relay log info log and master info log has been superseded by crash-safe slave tables, which are the default in MySQL 8.0. (WL #11031)

- The `slave_rows_search_algorithms` system variable is now deprecated and will be removed in a future MySQL version. This variable was used to control how rows were searched for matches when preparing batches of rows for row-based logging and replication. The default setting `INDEX_SCAN,HASH_SCAN` has been found to be optimal for performance and works correctly in all scenarios. (WL #12892)

- The `log_bin_use_v1_row_events` system variable is now deprecated and will be removed in a future MySQL version. When set to `ON`, the variable made `mysqld` write the binary log using Version 1 binary log row events, instead of Version 2 binary log row events which are the default from MySQL 5.6. (The default is `OFF`.) The use of Version 1 binary log row events enabled row-based replication with slaves running MySQL Server 5.5 and earlier, which could not use Version 2 binary log row events. (WL #12926)

- The `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()` function is now deprecated, and the `WAIT_FOR_EXECUTED_GTID_SET()` function should be used instead. Both functions wait until all of the specified transactions have been applied, or until the optional timeout has elapsed. However, `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()` applied to a specific replication channel, and stopped only after the transactions had been applied on that channel, for which the applier had to be running. In contrast, `WAIT_FOR_EXECUTED_GTID_SET()` stops after the specified transactions have been applied

on the server, regardless of how they were applied (on any replication channel or from any user client), and whether or not any replication channels are running. `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()` could hang indefinitely if an expected transaction arrived on a different replication channel or from a user client, for example in a failover or manual recovery situation, and no timeout was set. (WL #13178)

## Keyring Notes

- MySQL Enterprise Edition now includes a `keyring_hashicorp` plugin that uses HashiCorp Vault as a back end for keyring storage. For more information, see The MySQL Keyring. (WL #12522)

## Logging Notes

- Log buffering during server startup was adjusted for less delay in the appearance of output in the error log. (Bug #30019632)

## Optimizer Notes

- Queries involving first match split jump operations were handled wrongly in the iterator executor. They are now rewritten to weedout. (Bug #30220791)

- Hash joins have been implemented as a way of executing inner equi-joins in MySQL. For example, a query such as this one can be executed as a hash join beginning with this release:

  ```
  SELECT *
      FROM t1
      JOIN t2
          ON t1.c1 = t2.c1;
  ```

  Multi-table joins using equi-joins can also take advantage of this optimization.

  A hash join requires no index for execution. In most cases, a hash join is more efficient than the block-nested loop algorithm previously used for equi-joins without indexes.

  By default, beginning with this release, a hash join is used whenever a join includes at least one equi-join condition, and no indexes can be applied to the join condition. (A hash join can still be used where there are indexes applying to single-table predicates only.) This preference can be overridden by setting the `hash_join` optimizer switch to `off`, or by using the `NO_HASH_JOIN` optimizer hint. In addition, you can control the amount of memory used by a hash join by setting `join_buffer_size`. A join whose memory requirement exceeds this amount is executed on disk; an on-disk hash join uses a number of disk files and may not be executable if this number exceeds `open_files_limit`.

  A hash join cannot be employed if the join conditions for any pair of joined tables do not include at least one equi-join condition among all join conditions used. A hash join *is* used for a Cartesian product—that is, a join that specifies no join conditions at all.

  You can see whether hash joins have been used to optimize a query in the output of `EXPLAIN FORMAT=TREE` or `EXPLAIN ANALYZE`.

  In addition, inner joins using hash joins can now also take advantage of Batched Key Access (BKA). Outer joins still allocate the entire join buffer.

  For more information, see Hash Join Optimization. (WL #2241, WL #13002)

- Added `EXPLAIN ANALYZE`, which provides iterator-based timing, cost, and other information about queries in `TREE` format. This statement produces output similar to that of `EXPLAIN`, but with additional information about how optimizer estimates match actual execution. (WL #4168)

- MySQL now performs injection of casts into queries to avoid certain data type mismatches; that is, the optimizer now adds casting operations in the item tree inside expressions and conditions in which the data type of the argument and the expected data type do not match. This makes the query as executed equivalent to one which is compliant with the SQL standard while maintaining backwards compatibility with previous releases of MySQL.

  Such implicit casts are now performed between temporal types and numeric types by casting both arguments as `DOUBLE` whenever they are compared using any of the standard numeric comparison operators. They are also now performed for such comparisons between `DATE` or `TIME` values and `DATETIME` values, in which case the arguments are cast as `DATETIME`.

  For example, a query such as `SELECT * FROM t1 JOIN t2 ON t1.int_col = t2.date_col` is rewritten and executed as `SELECT * FROM t1 JOIN t2 ON CAST(t1.int_col AS DOUBLE) = CAST(t2.date_col AS DOUBLE)`, and `SELECT * FROM t1 JOIN t2 ON t1.time_col = t2.date_col` is transformed to `SELECT * FROM t1 JOIN t2 ON CAST(t1.time_col AS DATETIME) = CAST(t2.date_col AS DATETIME)` prior to execution.

  It is possible to see when casts are injected into a given query by viewing the output of `EXPLAIN ANALYZE`, `EXPLAIN FORMAT=JSON`, or `EXPLAIN FORMAT=TREE`. `EXPLAIN` can also be used, but in this case it is also necessary to issue `SHOW WARNINGS` afterwards.

  This change is not expected to cause any difference in query results or performance. (WL #12108)

## Packaging Notes

- The `component_test_page_track_component.so` test plugin has been moved to `-test` packages. (Bug #30199634)
- Binary packages that include `curl` rather than linking to the system `curl` library have been upgraded to use `curl` 7.65.3. (Bug #30015512)

## Pluggable Authentication

- To assist in debugging failed connections for accounts that use an LDAP authentication plugin, the `authentication_ldap_simple_log_status` and `authentication_ldap_sasl_log_status` system variables now accept a maximum value of 6 (formerly 5). Setting either variable to 6 causes debugging messages from the LDAP library to be written to the error log for the corresponding plugin. (Bug #29771393)

## Security Notes

- For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.1.1c. Issues fixed in the new OpenSSL version are described at https://www.openssl.org/news/cl111.txt and https://www.openssl.org/news/vulnerabilities.html. (Bug #29868818)

## Spatial Data Support

- Previously, for geometry arguments in a geographic SRS, `ST_Distance()` supported only argument types of `Point` and `Point`, or `Point` and `MultiPoint`. `ST_Distance()` now supports distance calculations for geographic SRS arguments of all geometry types. For more information, see Spatial Relation Functions That Use Object Shapes. (WL #12216)

## sys Schema Notes

- The `sys.schema_unused_indexes` view now filters out unique indexes. Thanks to Gillian Gunson for the contribution. (Bug #24798995, Bug #83257)
- The `sys.ps_is_consumer_enabled()` function now produces an error rather than returning `NULL` if the argument is an unknown non-`NULL` consumer name. (Bug #24760317)
- Previously, `sys` schema sources were maintained in a separate Git repository. `sys` schema sources now are included with and maintained within MySQL source distributions (under `scripts/sys_schema`). As a consequence of this change, to simplify maintenance of `sys` schema scripts going forward, the acceptable format of statements in the file named by the `init_file` system variable has been expanded. For details, see the description of that variable in Server System Variables.

  The `sys.version` view is deprecated and will be removed in a future MySQL version. Affected applications should be adjusted to use an alternative instead. For example, use the `VERSION()` function to retrieve the MySQL server version. (WL #12673)

## Test Suite Notes

- MySQL tests were updated to use the latest version of googletest. (Bug #30079649)

## X Plugin Notes

- X Protocol did not correctly display large numbers of warning messages that were issued for the same query. (Bug #30055869)
- When the `NO_BACKSLASH_ESCAPES` SQL mode was enabled, X Plugin incorrectly reported collections as tables. (Bug #28208285)

## Functionality Added or Changed

- **Replication; Group Replication:** By default, MySQL replication (including Group Replication) does not carry out privilege checks when transactions that were already accepted by another server are applied on a replication slave or group member. From MySQL 8.0.18, you can create a user account with the appropriate privileges to apply the transactions that are normally replicated on a channel, and specify this as the `PRIVILEGE_CHECKS_USER` account for the replication applier. MySQL then checks each transaction against the user account's privileges to verify that you have authorized the operation for that channel. The account can also be safely used by an administrator to apply or reapply transactions from `mysqlbinlog` output, for example to recover from a replication error on the channel. The use of a `PRIVILEGE_CHECKS_USER` account helps secure a replication channel against the unauthorized or accidental use of privileged or unwanted operations.

  You grant the `REPLICATION_APPLIER` privilege to enable a user account to appear as the `PRIVILEGE_CHECKS_USER` for a replication applier thread, and to execute the internal-use `BINLOG` statements used by `mysqlbinlog`. After setting up the user account, use the `GRANT` statement to grant additional privileges to enable the user account to make the database changes that you expect the applier thread to carry out, such as updating specific tables held on the server. These same privileges enable an administrator to use the account if they need to execute any of those transactions manually on the replication channel. If an unexpected operation is attempted for which you did not grant the appropriate privileges, the operation is disallowed and the replication applier thread stops with an error. (WL #12966)
- **Group Replication:** For group communication connections, Group Replication now supports the TLSv1.3 protocol, which was supported by MySQL Server from 8.0.16. To use the TLSv1.3 protocol,

MySQL Server must be compiled using OpenSSL 1.1.1 or higher. For information on configuring encrypted connections for Group Replication, see Securing Group Communication Connections with Secure Socket Layer (SSL). (WL #12990)

- **Group Replication:** A new option `OFFLINE_MODE` is available for the `group_replication_exit_state_action` system variable, which specifies how Group Replication behaves when a server instance leaves the group unintentionally, for example after encountering an applier error, or in the case of a loss of majority, or when another member of the group expels it due to a suspicion timing out.

  When `OFFLINE_MODE` is specified as the exit action, Group Replication switches MySQL to offline mode by setting the system variable `offline_mode` to `ON`. When the member is in offline mode, connected client users are disconnected on their next request and connections are no longer accepted, with the exception of client users that have the `CONNECTION_ADMIN` or `SUPER` privilege. Group Replication also sets the system variable `super_read_only` to `ON`, so clients cannot make any updates, even if they have connected with the `SUPER` privilege.

  The `OFFLINE_MODE` exit action prevents updates like the default `READ_ONLY` exit action does, but also prevents stale reads (with the exception of reads by client users with the stated privileges), and enables proxy tools such as MySQL Router to recognize that the server is unavailable and redirect client connections. It also leaves the instance running so that an administrator can attempt to resolve the issue without shutting down MySQL, unlike the existing alternative `ABORT_SERVER` exit action, which shuts down the instance. (WL #12895)

- **Group Replication:** An internal message service has been added to Group Replication. MySQL modules can use the service to transmit generic messages with an identifying tag to all group members, using Group Replication's existing group communication connections. (WL #12896)

- When the server is run with `--initialize`, there is no reason to load non-early plugins. The server now logs a warning and ignores any `--plugin-load` or `--plugin-load-add` options given with `--initialize`. (Bug #29622406)

- The number of diagnostic messages relating to `INFORMATION_SCHEMA` upgrade during server startup has been reduced. (Bug #29440725, Bug #94559)

- Prior to MySQL 8.0, `REVOKE` produced an error for attempts to revoke an unknown privilege. In MySQL 8.0, an account can possess a privilege that is currently unknown to the server if it is a dynamic account that was granted while the component or plugin that registers the privilege was installed. If that component or plugin is subsequently uninstalled, the privilege becomes unregistered, although accounts that possess the privilege still possess it. Revoking such a privilege cannot be distinguished from revoking a privilege that actually is invalid, so `REVOKE` no longer produces an error for attempts to revoke an unknown privilege. However, to indicate that the privilege is currently unknown, `REVOKE` now produces a warning. (Bug #29395197)

- The new `innodb_idle_flush_pct` variable permits placing a limit on page flushing during idle periods, which can help extend the life of solid state storage devices. See Limiting Buffer Flushing During Idle Periods.

  Thanks to Facebook for the contribution. (Bug #27147088, Bug #88566, WL #13115)

- `mysqld` invoked with the `--help` option no longer aborts if the `secure_file_priv` argument does not exist. (Bug #26336130)

- The Protobuf libraries are now dynamically linked. Their default locations are `/usr/lib64/mysql/private/` for RPMs, `/usr/lib/mysql/private/` for DEBs, and `/lib/private/` for TAR builds. A new `INSTALL_PRIV_LIBDIR` CMake variable controls the location. (WL #13126)

## Bugs Fixed

- **NDB Cluster:** A query handled using a pushed condition produced incorrect results when it included an `ORDER BY` clause. (Bug #29595346)

  References: This issue is a regression of: Bug #28672214.

- **NDB Cluster:** The `NDB` transporter layer limits the size of messages to 32768 bytes; send buffers place additional (and stricter) limitations on message size. Whenever a message is appended to a send buffer, page checks are performed to ensure that the message fits in the available space; if not, a new page is used. The current issue arose on account of the fact that no check was performed to make sure that this message could fit in the empty page; when the size of the message exceeded the empty page, this resulted in a buffer overwrite and in the overwriting of the next page in memory. For data nodes the largest message supported by the send buffer (`thr_send_page`) is 32756 bytes; for API and management nodes, this maximum is 32752 bytes. (Signals sent within an individual data node are not subject to these limitations since no send or transporter buffers are used in this case). Now, when a new page is used, the size of the message is checked against that which is available in a new page.

  As part of the work done to fix the problem just described, three new `DUMP` commands are added to facilitate related testing and debugging: `DUMP 103003` (`CmvmiRelayDumpStateOrd`) sends a `DUMP` command using another node; `DUMP 103004` (`CmvmiDummySignal`) and `DUMP 103005` (`CmvmiSendDummySignal`) can be used to send long messages. (Bug #29024275)

- **NDB Cluster:** `EXPLAIN FORMAT=TREE` did not provide proper explanations of conditions and joins pushed down to the `NDBCLUSTER` storage engine. Issues included the following:

  - Pushed conditions were not shown.
  - The root of a pushed join was not shown.
  - The child of a pushed join did not include any reference to its parent operation.

- **InnoDB:** An internal function (`btr_push_update_extern_fields()`) used to fetch newly added externally stored fields and update them during a pessimistic update or when going back to a previous version of a record was no longer required. Newly added externally stored fields are updated by a different function. Also, the method used to determine the number of externally stored fields was corrected. (Bug #30342846)

- **InnoDB:** An `DROP UNDO TABLESPACE` operation on an undo tablespace with a missing data file caused a segmentation fault. (Bug #30155290)

- **InnoDB:** An error in the internal `trx_rseg_add_rollback_segments` function was corrected. (Bug #30114226, Bug #96372)

- **InnoDB:** Problematic assertion code in the Contention-Aware Transaction Scheduling (CATS) code was revised. (Bug #30086559)

- **InnoDB:** Arguments passed to a derived class in calls from the `ib::fatal` and `ib::fatal_or_error` constructors could be ignored, resulting in invalid error messages. Additionally, in the case of a fatal error, the `ib::fatal_or_error` destructor could cause a server exit before printing a message. (Bug #30071930)

- **InnoDB:** It was possible for a corrupted table to be removed from the table cache before the reference count for the table reached zero. (Bug #30065947, Bug #96224)

- **InnoDB:** A code path for the internal `row_update_inplace_for_intrinsic()` function did not include a required mini-transaction (mtr) commit, causing a debug assertion failure. (Bug #30065518)

- **InnoDB:** The internal `fsp_srv_undo_tablespace_fixup()` function did not take an `undo::ddl_mutex` lock when called during startup, which could lead to an assertion failure under certain circumstances. (Bug #30029433, Bug #30461911, Bug #97356)
- **InnoDB:** Inspection of `rename_tablespace_name()` function showed that if `old_shard->get_space_by_id(space_id)` did not find the tablespace ID, it would return without calling `old_shard->mutex_release()`. (Bug #30027771)
- **InnoDB:** A tablespace with “FTS” in its name was incorrectly determined to be the tablespace of a full-text index table and not registered with the data dictionary during upgrade, causing the upgrade operation to fail. (Bug #29992589)
- **InnoDB:** The `ibuf_merge_or_delete_for_page()` function, responsible for merging and deleting pages in the change buffer, is no longer called for undo tablespaces and temporary tablespaces. The change buffer does not contain entries for those tablespace types. (Bug #29960394, Bug #95989)
- **InnoDB:** Some combinations of buffer pool size variables (`--innodb-buffer-pool-instances`, `--innodb-buffer-pool-size`, and `--innodb-buffer-pool-chunk-size`) lead to a chunk size that is 0, 1, or a value that is not a multiple of the page size, and so on, causing errors during buffer pool creation and sizing. (Bug #29951982)
- **InnoDB:** A server exit while an undo tablespace was being truncated caused the following error when a clone operation was rolled forward after restarting the server: `[ERROR] [MY-011825] [InnoDB] [FATAL] Clone File Roll Forward: Invalid File State: 0.` (Bug #29949917)
- **InnoDB:** An `ALTER TABLE ... DISCARD TABLESPACE` operation caused a hang condition. (Bug #29942556, Bug #30324703)
- **InnoDB:** Cloning operations failed with the following error after an archive thread failure: `ERROR 3862 (HY000): Clone Donor Error: 1317 : Query execution was interrupted.` (Bug #29930839)
- **InnoDB:** A clone related regression caused a minor drop in performance for index and non-index update operations. (Bug #29925409)
- **InnoDB:** `InnoDB` now ignores hidden directories and files during the tablespace discovery scan that occurs at startup. Hidden directories and files include those beginning with "." and hidden and system directories and files on Windows that are identified by attributes. (Bug #29900671, Bug #95071, Bug #30068072)
- **InnoDB:** To improve deadlock detection, the task of detecting a deadlock cycle among data locks was moved from the transaction thread to a dedicated background thread. (Bug #29882690)
- **InnoDB:** Updating the `mysql.gtid_executed` table during `mysqld` initialization caused the following warning to be printed to the error log: `[Warning] [MY-010015] [Repl] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened`. The update and associated warning no longer occur. (Bug #29871809)
- **InnoDB:** The `LATEST DETECTED DEADLOCK` section in `InnoDB` Standard Monitor output (also printed by `SHOW ENGINE INNODB STATUS`) was extended to include additional information about transactions that participate in a deadlock cycle. (Bug #29871641)
- **InnoDB:** An incorrect argument was used to compare serialized dictionary information (SDI) input and output values when checking the size of the buffer used to store SDI. (Bug #29871525, Bug #95606)
- **InnoDB:** An undo tablespace file was overwritten during a cloning operation when undo tablespaces files with the same name were copied from different directories on the donor to the same directory on the recipient. A cloning operation now reports an error if duplicate undo tablespace names are encountered.

When data is cloned to the recipient, undo tablespace files are cloned to the directory defined by the `innodb_undo_directory` variable. Because the files are cloned to the same directory, duplicate undo tablespace file names are not permitted. (Bug #29837617)

- **InnoDB:** A cloning operation generated a large number of redo log files when cloning from a MySQL server instance with a small redo log file size and a large number of transactions. The large number of redo log files sometimes caused startup to hang or assert. Otherwise, startup was permitted proceed without cleaning up the excessive number of redo log files. (Bug #29837490)
- **InnoDB:** During recovery, missing table errors that occur when applying redo logs to encrypted tables could be ignored, permitting startup to proceed. Startup should be halted in this case. (Bug #29820184, Bug #95183)
- **InnoDB:** The wrong redo log file size was printed to the server error log at startup. (Bug #29818711)
- **InnoDB:** A restriction that prevented reuse of lock objects (`lock_t` structs) in the lock queue when waiting record lock requests were present was removed. (Bug #29814308)
- **InnoDB:** When converting a long partitioned table name to a file name, the buffer that holds the file name did not have enough space, causing an assertion failure. (Bug #29813582)
- **InnoDB:** Some transaction lock structure fields (`trx->lock`) were not properly mutex protected. (Bug #29809137)
- **InnoDB:** An empty undo segment update during recovery raised an assertion. (Bug #29802703)
- **InnoDB:** A cloning operation failed when attempting to drop data on the recipient if the recipient data included a table with a full-text index. (Bug #29796507)
- **InnoDB:** After a file-per-table tablespace was discarded, an error was reported when the old table was renamed and a new table was created with the same name. The error was due to stale file path information. (Bug #29793800)
- **InnoDB:** A test case that attempts to open a file in read-only mode while the server is in a disk-full state caused a debug assertion failure. The assertion was removed to permit the server to retry opening the file, and to report an error if unsuccessful after a number of attempts. (Bug #29692250, Bug #95128)
- **InnoDB:** Foreign-key-related tables constructed during a `RENAME TABLE` operation contained the old table name. (Bug #29686796)
- **InnoDB:** The server passed a NULL value for a column with a non-zero data length. (Bug #29654465)
- **InnoDB:** Importing a partitioned table from a MySQL 8.0.13 instance (or earlier) to a MySQL 8.0.14, 8.0.15, or 8.0.16 instance failed with a “tablespace is missing” error for source and target instances defined with `lower_case_table_names=1`. The tablespace file and the metadata file used by the import operation could not be found due to a file name case mismatch. (Bug #29627690, Bug #94850)

  References: This issue is a regression of: Bug #26925260.
- **InnoDB:** The read set (`TABLE::read_set`) was not set properly for handler method calls in `QUICK_SKIP_SCAN_SELECT::get_next()`, causing an assertion failure. (Bug #29602393)
- **InnoDB:** An exclusive backup lock is now used to block tablespace truncate operations during master key rotation. Previously, metadata locks on undo tablespace names were used to synchronize the operations. This patch also addresses a deadlock that could occur between master key rotation and drop undo tablespace operations. (Bug #29549938, Bug #30461225, Bug #97352)
- **InnoDB:** An unnecessary next key lock was taken when performing a `SELECT...FOR [SHARE|UPDATE]` query with a `WHERE` condition that specifies a range, causing one too many rows to be locked.

The most common occurrences of this issue have been addressed so that only rows and gaps that intersect the searched range are locked. (Bug #29508068)

- **InnoDB:** Adding a virtual column in the first position together with a normal column raised an assertion failure. (Bug #29501324)
- **InnoDB:** The shutdown process did not wait for `InnoDB` background threads to exit, which could cause shutdown to stall, waiting for background threads to remove transactions from the MySQL transaction list. A check now occurs to ensure that `InnoDB` background threads have exited properly. Also, an intermediate shutdown state was added to improve shutdown timing for the `InnoDB` master thread. (Bug #29417503)
- **InnoDB:** A memory leak occurred during TempTable storage engine operation due a failure to deallocate that last block of allocated memory, which TempTable holds in thread-local storage until thread exit. (Bug #29300927)
- **InnoDB:** Throughput stalled under a heavy workload with a small `innodb_io_capacity_max` setting, a single page cleaner thread, and multiple buffer pool instances. (Bug #29029294)
- **InnoDB:** A long running `ALTER TABLE ... ADD INDEX` operation with concurrent inserts caused semaphore waits. Thanks to Satya Bodapati for the patch. (Bug #29008298)
- **InnoDB:** The `INFORMATION_SCHEMA.INNODB_COLUMNS` table did not display partitioned table columns after upgrading from MySQL 5.7 to MySQL 8.0. For partitioned tables created on the MySQL 8.0 release, `INFORMATION_SCHEMA.INNODB_COLUMNS` only displayed columns for the first table partition. (Bug #28869903, Bug #93033)
- **InnoDB:** The internal `rtr_page_split_and_insert()` function is called recursively. An inner call to the function released an object still being used by an outer call to the same function, causing an assertion failure. (Bug #28569379)
- **InnoDB:** Under specific circumstances, setting the `innodb_limit_optimistic_insert_debug` variable to 2 raised a debug assertion when it should have reported an error. (Bug #28552330, Bug #92187)
- **InnoDB:** A deadlock was possible when a transaction tries to upgrade a record lock to a next key lock. (Bug #23755664, Bug #82127)
- **InnoDB:** The `INFORMATION_SCHEMA.INNODB_METRICS` `lock_deadlocks` counter did not count all deadlocks. Thanks to Laurynas Biveinis for the contribution. (Bug #21278148, Bug #77399)
- **InnoDB:** An error reported during a read operation did not identify the name of the file that was read. (Bug #21120885, Bug #76020)
- **Replication:** If the `group_replication_set_communication_protocol()` function was used to set a communication protocol version that some group members did not support, the protocol change procedure was not stopped correctly. The issue has now been fixed. (Bug #30209596)
- **Replication:** The session value of the `default_table_encryption` system variable is replicated, so that replication slaves preserve the correct encryption setting for databases and tablespaces. For a replication channel where the `table_encryption_privilege_check` system variable is set to `ON`, the `TABLE_ENCRYPTION_ADMIN` privilege is required to apply events where the session value of `default_table_encryption` is changed. If this privilege had not been granted on the replication slave, replication stopped with an error in some situations where the required encryption setting for the event was actually the same as the slave's own default encryption setting, so the action ought to have worked. It was also possible for a statement with a different encryption setting from the slave to succeed when it ought not to have worked. The behaviors have now been corrected so that a replication slave without the `TABLE_ENCRYPTION_ADMIN` privilege is permitted to apply events that match its own default

encryption setting, and is not requested to set the encryption setting unnecessarily, but is still subject to appropriate privilege checks if the event requires a different encryption setting. (Bug #29818605)

- **Replication:** The timeout in seconds specified by the `group_replication_unreachable_majority_timeout` system variable was only checked every two seconds. The timeout is now checked every second so that odd-numbered values are respected. (Bug #29762005)

- **Replication:** A deadlock involving three threads could occur if a `START SLAVE` statement was issued to start the SQL thread on a slave while the SQL thread was still in the process of being stopped, and a request for the slave status was made at the same time. The issue has now been fixed by releasing a lock earlier while the SQL thread is being stopped. (Bug #29697588, Bug #95115)

- **Replication:** A locking issue in the `WAIT_FOR_EXECUTED_GTID_SET()` function could cause the server to hang in certain circumstances. The issue has now been corrected. (Bug #29550513)

- **Replication:** A deadlock could occur if the value of the `binlog_encryption` system variable was changed while a `START SLAVE` statement was being executed. (Bug #29515210)

- **Replication:** The heartbeat interval for a replication slave, which is controlled by the `MASTER_HEARTBEAT_PERIOD` option of the `CHANGE MASTER TO` statement, can be specified with a resolution in milliseconds. Previously, the master's binary log dump thread used a granularity of seconds to calculate whether a heartbeat signal should be sent to the slave, causing excessive heartbeat activity in the case of multiple skipped events. To remove this issue, all heartbeat-related calculations by the master and slave are now carried out using a granularity of nanoseconds for precision. Thanks to Facebook for the contribution. (Bug #29363787, Bug #94356)

- **Replication:** Creation of a table with functional index creates a hidden generated column on the table, but this column did not always appear in the same position in the table, which meant that later adding a column (or columns) to the table might leave the hidden column in a different position as compared to an otherwise identical table already having the additional column or columns in a single CREATE TABLE statement. For example, this difference could be evident between two tables `t1` and `t2`, with `t1` created and then altered by the two statements `CREATE TABLE t (a INT, INDEX ((a+1)))` and `ALTER TABLE t ADD COLUMN b INT`, and `t2` created using `CREATE TABLE t2 (a INT, b INT, INDEX ((a+1)))`, even though `SHOW CREATE TABLE t1` and `SHOW CREATE TABLE t2` produced identical results (other than for the names of the tables).

  The issue just described could become problematic due to the fact that the internal definitions for the tables as described by `Table_map_log_events` in the binary log would differ. This meant that, if such a table was created using `CREATE TABLE` followed by `ALTER TABLE` on the master, and then a slave was deployed using `mysqldump`, the slave's version of the table would be created using a single `CREATE TABLE` statement, and subsequent replication of row events would then fail with an error due to mismatching types.

  This issue is fixed by forcing all hidden generated columns always to be located at the end of the list of columns, with multiple generated columns sorted by column name. If a new functional index part is added to the table, the new hidden column is inserted according to this rule. In addition, if the user creates a new column that is not a generated column, the column is always added before the first hidden column.

  **Note**

  The structure of tables affected by this change is not changed automatically during upgrade; this must done explicitly by executing `ALTER TABLE`.

  (Bug #29317684)

- **Replication:** If a `RESET MASTER` statement was stopped during its execution, the binary log was not available for writes afterwards. (Bug #29125121)

- **Replication:** On a multi-threaded slave with GTIDs in use and `MASTER_AUTO_POSITION` set to `ON`, following an unexpected halt the slave would attempt relay log recovery, which failed if relay logs had been lost, preventing replication from starting. However, this step was unnecessary as GTID auto-positioning can be used to restore any missing transactions. In a recovery situation, the slave now checks first whether `MASTER_AUTO_POSITION` is set to `ON`, and if it is, omits the step of calculating the transactions that should be skipped or not skipped, so that the old relay logs are not needed and recovery can proceed without them. (Bug #28830834, Bug #92882)

- **Group Replication:** If the TCP connections used by Group Replication time out due to a lengthy network error, the group communication engine (XCom) cannot re-establish its local connection, requiring a restart of Group Replication on the instance (see Bug #25656508). Previously, attempting to stop Group Replication in this situation caused XCom to hang, preventing the restart. This issue has now been fixed so that XCom terminates correctly and Group Replication can be restarted to re-establish the local connection. (Bug #30132500)

  References: See also: Bug #22158368.

- **Group Replication:** When the `ANSI_QUOTES` SQL mode was enabled, Group Replication returned an error when checking for the presence of MySQL Server's clone plugin to perform remote cloning operations for distributed recovery. The issue has now been fixed. (Bug #30099796)

- **Group Replication:** `default_table_encryption` could be changed while Group Replication was running by issuing `SET PERSIST`, even though this variable should never be settable insuch cases. (Bug #30031228, Bug #96158)

- **Group Replication:** If `group_replication_local_address` was changed for a member which then rejoined the group, copies of the old local address were retained in some locations by the Group Communication System (GCS). To fix this, we now fetch the local address when needed, and no longer obtain it from the cache. (Bug #29910699)

- **Group Replication:** When a remote cloning operation was used to provide transactions to a joining member and the joining member was stopped during the operation, it did not shut down cleanly. (Bug #29902389)

- **Group Replication:** The input channel introduced in this release uses a shared memory queue instead of a TCP socket for communication between the Group Communication System (GCS) component of Group Replication and the local group communication engine (XCom) instance. This input channel could not be established on SELinux installations, which meant members upgraded to MySQL 8.0.14 or later were unable to rejoin the group. When Group Replication was started, the XCom instance temporarily opened a port from the ephemeral port range to allow GCS to establish a connection for the input channel, but on SELinux the `mysqld` process did not have permission to connect to this port. A workaround was to amend the SELinux policy to allow MySQL to connect to any port, but this reduced security. This issue is now fixed; XCom and GCS no longer use an ephemeral port to establish a connection for the input channel, but instead use the Group Replication communication port set by `group_replication_local_address`. This action must be permitted by SELinux (see Frequently Asked Questions). (Bug #29742219, Bug #30087757)

- **Group Replication:** Server certificates whose Common Name value used a wildcard were incorrectly rejected. (Bug #29683275, Bug #95068)

- **Group Replication:** It was possible for the Group Communication System (GCS) to deliver messages from a member after `STOP GROUP_REPLICATION` had been issued and the member's status was `OFFLINE`, resulting in an error. GCS now verifies that the member belongs to a group before delivering any such messages. (Bug #29620900)

- **Group Replication:** Some threads that executed transient or minor tasks, such as the delayed plugin initialization thread, were not visible in the Performance Schema `threads` table. (Bug #28930537, Bug #93212)
- **Group Replication:** The implementation for the process of a member leaving the group due to an error has been standardized across components. This ensures that the member carries out exactly the same actions and issues the same error messages consistently, regardless of the original reason for leaving the group. (Bug #28866495, Bug #93027)
- For `XA COMMIT` statements, invocation order of the plugins involved in statement execution was nondeterministic, which could lead to replication problems. Thanks to Dennis Gao for contributing a fix. (Bug #31082237, Bug #99051)
- The `keyring_aws` plugin was not included in Commercial Docker RPM packages. (Bug #30199423)
- When generating C source from SQL scripts, Some utf8-encoded characters were split across lines. Thanks to Przemysław Skibiński for the patch. (Bug #30152555, Bug #96449)
- The `ARRAY` reserved word was listed as unreserved in the `INFORMATION_SCHEMA.KEYWORDS` table. (Bug #30134275, Bug #96416)
- Some statements containing `||` produced a parse error even with the `PIPES_AS_CONCAT` SQL mode enabled. (Bug #30131161, Bug #96405)

  References: This issue is a regression of: Bug #29305022.
- `LOAD DATA` statements incorrectly ignored the `NO_AUTO_VALUE_ON_ZERO` SQL mode if a `SET` clause was present. (Bug #30126375)
- Automatic upgrades failed with a server exit if the `thread_pool` plugin was enabled. (Bug #30121742)
- When determining whether to recalculate a materialized derived table for each execution, the uncacheability of the parent query block was used instead of the rematerialize flag on the table (currently only true for `JSON_TABLE()`). This could cause unneeded rematerializations, especially when making queries against non-merged views, leading to reduced performance. (Bug #30110851)
- With an `IGNORE` clause, `LOAD DATA` should skip rows that produced a `CHECK` constraint violation and continue with the following rows, but it stopped with an error. (Bug #30084966, Bug #96296)
- With `sql_require_primary_key` enabled, clone plugin initialization failed because it created two dynamic Performance Schema tables, which do not support indexes or primary keys. The effect of `sql_require_primary_key` is now limited to storage engines that can participate in replication (currently all storage engines except the Performance Schema) (Bug #30083300, Bug #96281)
- Protobuf compilation failed on macOS. (Bug #30079536, Bug #96263)
- For automatic upgrades, the `audit_log` plugin was not reloaded, causing auditing to start in legacy mode. (Bug #30068110)
- In order to remove duplicate rows, the weedout optimization needs a unique identifier for each row, which is provided by the storage engine by calling `handler::position()`, but this function was not always called at the correct time, or was sometimes called when not needed. Issues regarding how and when to place such calls were the source of many bugs in optimizations. This fix moves the responsibility of calling `handler::position()` to the iterators, whose default implementations usually handle this task adequately on their own. (Bug #30060691)

  References: See also: Bug #29693294, Bug #30049217, Bug #30153695.
- The server could exit due to mishandling a `COM_PROCESS_INFO` command. (Bug #30032302)

- Warnings are normally generated when deprecated system variables are assigned a value, but this did not occur when persisted system variables were processed during server startup. (Bug #30030648)
- With `bind_address` set to a value containing multiple addresses, setting `host_cache_size=0` caused a server exit at client connect time. (Bug #30018958)
- On Windows, Protobuf compilation failed for the Ninja build type. (Bug #30018894)
- `SHOW PROCESSLIST` output could include statements that had completed and were no longer in process. (Bug #29999818)
- A `mysqldump` error-message buffer was too small, potentially leading to message truncation. (Bug #29999782, Bug #96074)
- Last used auto-increment values instead of “next to be used” auto-increment values were cached while importing tables from MySQL 5.7 to MySQL 8.0 during an in-place upgrade, which resulted in any tables read from the table cache after upgrade having incorrect auto-increment values. (Bug #29996434)
- When executing a prepared statement that used constant folding, it was possible to register a location for rollback which was not allocated on the `MEMROOT`, but rather on the stack, and so was no longer in scope. (Bug #29990693)

  References: See also: Bug #29939331.
- A warning about `ZEROFILL` being deprecated was produced for `CREATE TABLE ... AS` and `CREATE TABLE ... LIKE` statements that created tables that contained a `YEAR` column. This could be confusing because the `ZEROFILL` attribute may only have been added implicitly to the `YEAR` column. To avoid confusion, the warning is no longer raised for such statements. (Bug #29961060)
- A query using bitwise operators, such as in `WHERE text_col < (int_col & int_col) AND int_col = -1`, failed to return any rows as expected. (Bug #29957969)
- Malformed resource group names in optimizer hints could lead to unexpected server behavior. (Bug #29955732)
- `EXPLAIN`, when using `FORMAT=TREE`, did not show the same costs as with `FORMAT=JSON`. In particular, when using `FORMAT=TREE`, the cost calculated took into account `eval_cost` for joins, but not for base tables. (Bug #29953579)
- For Solaris, `mysqld.cc` contained a prototype for `memcntl()` that is no longer needed. The prototype has been removed. (Bug #29953495, Bug #95971)
- For Solaris, `-DWITH_SSL=system` did not work when compiling with GCC. (Bug #29953460, Bug #95970)
- MySQL builds configured with `-DWITHOUT_SERVER=1` failed. (Bug #29948728, Bug #95740)
- The internal method `Field_tiny::pack()` did not always perform bounds checking as expected. (Bug #29948029)

  References: See also: Bug #31591391.
- Improper handling of plugin unloading could cause a server exit. (Bug #29941948)
- The internal `MEM_ROOT` class did not handle all out-of-memory errors correctly. (Bug #29940846)
- `JSON_SCHEMA_VALIDATION_REPORT()` did not distinguish between a JSON document that did not validate according to a JSON schema and a JSON document which was too deeply nested, which resulted in wrong behavior in some cases. (Bug #29940833)

- Use of the `<=>` operator could yield incorrect results for comparisons involving very large constants. (Bug #29939331, Bug #95908)
- For the `keyring_aws` plugin, some valid region values for the `keyring_aws_region` system variable were rejected. (Bug #29933758)
- For debug builds, an assertion could be raised during `UNION` queries when computing the combined data type of a `GEOMETRY` column and `SELECT * FROM (SELECT NULL)`. (Bug #29916900, Bug #95827)
- For columns defined with the `binary` character set, `SHOW CREATE TABLE` could generate `CREATE TABLE` statements that produced a syntax error when executed. (Bug #29909573, Bug #95801)
- Data dictionary APIs were added for fetching table names in a schema that use a specific storage engine, and for fetching table names in a schema that are created and hidden by the storage engine. The former is required by NDB, and the latter is required for `DROP DATABASE` operations. (Bug #29906844, Bug #95792)
- `mysqldump` leaked memory when run with the `--order-by-primary` option. (Bug #29906736)
- For queries involving functional indexes, `EXPLAIN FORMAT=TREE` printed the hidden column name instead of the indexed expression. (Bug #29904996)
- For debug builds, `CREATE TABLE ... IGNORE SELECT ...` statements did not properly clean up the table state when the last row to be inserted was skipped due to a failed `CHECK` constraint. This could cause the next statement using the table to raise an assertion. (Bug #29903865)
- MySQL did not handle execution of a recursive common table expression (CTE) correctly when the termination condition of the recursive query for this CTE had an `IN` predicate using another recursive CTE. (Bug #29899614)
- Deserialization of serialized dictionary information (SDI) failed for a table with partitions residing in different tablespace files. (Bug #29898965)
- For authentication using an LDAP authentication plugin, if the user DN portion was empty and group mapping was configured, authentication assigned an incorrect user DN and skipped the user search. (Bug #29897624)
- Constraints defined without a name could in some cases cause a server exit. (Bug #29892876)
- An in-place upgrade from MySQL 5.7 to MySQL 8.0 failed due a missing NDB tablespace. (Bug #29889869, Bug #30113440)
- `mysqlpump` produced an error when run against a server older than MySQL 5.7. (Bug #29889253)
- When Boost was downloaded, `CMake` configuration logic for determining the size of the downloaded file was incorrect, and could remove that file after a successful download operation. (Bug #29881279)
- A possible integer overflow due to unsigned integer type casting could lead to later buffer overflow due to arbitrary size memory allocation. (Bug #29878914)
- When the client character set was other than `latin1`, the server converted the string representation of a `DECIMAL` value from `latin1` to the client character set. This conversion was not necessary, since all supported client character sets encode numbers in the same way as `latin1`, and so is no longer performed. (Bug #29875672)
- The `vio_description()` debugging function was called in nondebug builds. (Bug #29871361)

- The fix for a previous issue in MySQL 8.0.3 changed a test for whether an expression over constant `DATE` values could be cached from allowing any expressions except global variables to a blanket denial of all functions, regardless of whether they were over constants or not, which significantly impacted performance of related queries in which the optimizer needed to perform conversion of a string to a `DATE` value. This test has been reverted to its original form. (Bug #29871018)

  References: This issue is a regression of: Bug #85471, Bug #28576018.

- The `INFORMATION_SCHEMA` can fetch dynamic table statistics from storage engines, but this did not work for partitioned tables. (Bug #29870919, Bug #95641)

- Retrying a failed access-control statement could permit another thread to acquire a lock on the access-control cache during a window when metadata locks where released and reacquired, resulting in a deadlock. The locks are now not released during the retry operation. (Bug #29870899, Bug #95612)

- An assertion could be raised if a user without the proper privilege attempted to enable the `offline_mode` system variable. (Bug #29849046)

- Fedora packaging configuration put debug information for `mysql_config_editor` in the wrong package. (Bug #29841342)

- Attempted use of a freed object during MeCab plugin initialization caused a segmentation fault. (Bug #29832534)

- The function used to generate entropy for random passwords could sometimes not provide much entropy. (Bug #29808492)

- With `super_read_only` enabled, the server could process `DROP TABLESPACE` improperly. (Bug #29802833)

- The server did not handle a query with a left join containing a materialized semijoin correctly. (Bug #29800741)

- For MySQL installed using RPM packages, an initialization script that tested server connectivity misbehaved if the client account authenticated using an LDAP authentication plugin. (Bug #29786782)

- Improper locking during storage engine initialization could cause a server exit. (Bug #29782379)

- On a GTID-enabled server, concurrent execution of `DROP USER` and a prepared statement that accessed a view could deadlock. (Bug #29772622)

- When performing `EXPLAIN FORMAT=JSON` on a query of the form `SELECT WHERE NOT EXISTS (SELECT FROM (`*`derived_table`*`) WHERE `*`false_condition`*`)` involving an antijoin transformation, the subquery was eliminated but its own subqueries were moved up instead of being eliminated along with it. (Bug #29759277)

  References: See also: Bug #30017509.

- An interrupted tablespace encryption operation caused a discrepancy between data dictionary and storage engine metadata, resulting in an assertion failure on a subsequent attempt to access the tablespace. (Bug #29756808)

- Simultaneous execution of the event scheduler and `DROP EVENT` could result in lock acquisition hanging until lock wait timeout. (Bug #29742901, Bug #95223)

- `CHECK` constraints were not enforced when a column took its default value from an expression. (Bug #29706689, Bug #95192)

- When performing two sorts on the same table with no temporary table, both sorts tried to use the same buffer, resulting in a read out of bounds. This could occur when the same query used both `DISTINCT` and `ORDER BY`. (Bug #29699759)
- When performing a weedout involving a sorted table, sorting was not done by row IDs. (Bug #29693294)
- VS2019 produced compilation errors with debug compilation selected due to use of the `/ZI` flag. Now `/Z7` is used instead. (Bug #29691691, Bug #95125)
- In source distributions, these changes were made in relation to the DBUG package, to clean up old and unmaintained code in the `dbug` directory:
  - Moved `dbug.cc` to the `mysys` directory, removed the `dbug` library.
  - Removed unused functions in `dbug.cc`.
  - Removed the `DBUG_LEAVE` macro; no longer needed.
  - Wrapped the body of the `DBUG_LOG` macro in `if _db_enabled_()`, to avoid formatting of strings that likely will not be printed anyway.
  - Removed the `dbug` directory.

  (Bug #29680868, Bug #95082)
- While allocating the reference item array, the count for any window functions was not taken into consideration. As a result, when expressions having window functions were split, the extra space needed to store the new item references was not properly allocated. (Bug #29672621)
- Setting the collation of a user variable did not work correctly in all cases. (Bug #29665165)
- Processing of `WHERE` clauses in `SHOW` statements was not performed consistently. (Bug #29664758)
- If clauses to drop and add constraints were specified within the same `ALTER TABLE` statement, they were executed in incorrect order. (Bug #29652464)
- `EXPLAIN` now shows when a result is being sorted by row ID. (Bug #29634196)
- Unnecessary materialization is no longer performed in either of the following cases:
  - When performing a sort at the end of a join, such as for grouping, and the sorting does not make use of row IDs.
  - When materializing a derived table which is not to be read from more than once.

  (Bug #29634179)
- `CAST(arg AS FLOAT)` did not returned the expected out-of-range error when *`arg`* was too small to be represented as a `FLOAT` value. (Bug #29621333)
- Internal conversion of `FLOAT` values sometimes occurred at the wrong time, which could lead to excess precision in results. Consider the table created and populated as shown here:

```
CREATE TABLE t(f FLOAT);
INSERT INTO t VALUES (2.3);
```

  Prior to the fix, the query `SELECT f, IF(f > 0, f, 0) FROM t` against this table returned `(2.3, 2.299999952316284)` instead of `(2.3, 2.3)` as expected. (Bug #29621062)
- `log0meb.cc` failed to compile with some build options. (Bug #29616525)

- The client library could dereference a null pointer while fetching result set metadata from the server. (Bug #29597896, Bug #30689251)
- A runtime error could occur for interval value checks when summing date and interval values. (Bug #29587536)
- In builds with Undefined Behavior Sanitizer enabled, multiplication with -9223372036854775808 could produce warnings and incorrect results. (Bug #29581610)
- During startup, the server attempted to write operations to the binary log that should not have been logged. (Bug #29557747, Bug #94835)
- A statement of the form `CREATE VIEW v1 AS SELECT * FROM table WHERE (constant IN (SELECT constant) IS UNKNOWN)` led to an assertion. (Bug #29525304)

  References: This issue is a regression of: Bug #25466100.
- On EL6, if a FIPS-enabled OpenSSL library was not available, attempting to enable FIPS mode could cause a server malfunction. (Bug #29519794)
- `SET PERSIST_ONLY` can be used to persist some system variables that `SET PERSIST` cannot (due to the way variable settings occur). If such a variable is found in `mysqld-auto.cnf` at startup, the server now issues a warning and ignores the setting. (Bug #29511118)
- The server did not properly aggregate a partial revoke and a schema-level privilege in some cases. (Bug #29484519)
- For debug builds, `ENUM` or `SET` columns defined with a `DEFAULT` clause raised an assertion. (Bug #29480711)
- MySQL Installer did not install OpenSSL DLL dependencies if the Development component was not selected. (Bug #29423421, Bug #94168, Bug #30199579, Bug #96573)
- In `READ UNCOMMITTED` isolation level, a segmentation fault occurred under heavy load from `memcached` clients. An externally stored `BLOB` column that was being updated by one transaction was read by another transaction as having a NULL value and a non-zero data length. (Bug #29396364, Bug #93961)
- On systems with working IPv6 address resolution, IPv6 socket creation failure at connect time resulted in a memory leak. (Bug #29374606, Bug #94384)
- In replication scenarios, if multiple clients concurrently executed `XA COMMIT` or `XA ROLLBACK` statements that used the same XID value, replication inconsistency could occur. (Bug #29293279, Bug #94130)
- Arguments for the `TIMESTAMPADD()` function could be reversed for prepared statements. (Bug #29268394)
- The `DEFAULT ROLE` option for `CREATE USER` statements was not written to the binary log. (Bug #28948915, Bug #93252)
- Starting the server from the command line with invalid memcached plugin variable settings caused the server to exit. (Bug #28575863)
- The `SET` clause for `LOAD DATA` did not work to set `GEOMETRY NOT NULL` columns. (Bug #28460369, Bug #91893)
- On Windows 8 and higher, the `keyring_aws` plugin was not able to communicate with Amazon KMS servers. (Bug #28377961)

- `INFORMATION_SCHEMA` tables and `SHOW COLUMNS` could produce incorrect view column types when a query on the view had multiple query blocks combined with the `UNION` operator. (Bug #28278220, Bug #91486)
- On Debian, long `InnoDB` recovery times at startup could cause systemd service startup failure. The default systemd service timeout is now disabled (consistent with RHEL) to prevent this from happening. (Bug #28246585, Bug #91423)
- With the `thread_pool` plugin enabled, the `sys.processlist` and `sys.session` views displayed a thread name rather than the actual user name. (Bug #25906021, Bug #85976)
- The `delete_latency` column in the `sys.schema_index_statistics` view incorrectly referred to the `SUM_TIMER_INSERT` column of the Performance Schema `table_io_waits_summary_by_index_usage` table rather than the `SUM_TIMER_DELETE` column. (Bug #25521928)
- In output from the `sys.diagnostics()` procedure, the `latency` column for the `user_summary_by_file_io_type` view was incorrectly displayed in raw picoseconds rather than as a formatted value. (Bug #25287996)
- MySQL Enterprise Encryption functions could apply Diffie-Hellman (DH) methods to non-DH keys, resulting in unpredictable results or server exit. (Bug #22839007)
- Password masking was incomplete for `SHOW PROCESSLIST` and some `INFORMATION_SCHEMA` and Performance Schema tables. (Bug #20712046)
- With strict SQL mode enabled, the `STR_TO_DATE()` function did not properly handle values with time parts only. Thanks to Daniel Black for the contribution. (Bug #18090591, Bug #71386)
- A query using `GREATEST()` in the `WHERE` clause could, in certain cases, fail to return a row where one was expected or raise a spurious error. (Bug #96012, Bug #29963278)
- Explicitly setting `sort_buffer_size` to its maximum value or close to it caused some queries to fail with an out of memory error. (Bug #95969, Bug #29952775)

  References: See also: Bug #22594514.
- Late `NULL` filtering, to avoid index lookups if the lookup key contains at least one `NULL`, was not performed for backwards index scans, although it was for forward index scans. (Bug #95967, Bug #29954680)
- Using a function such as `IFNULL()` or `ABS()` in a functional index led to the error `Value is out of range for functional index...` when an `UNSIGNED` column used as an argument to such a function contained a value exceeding that of the corresponding signed integer type. (Bug #95881, Bug #29934661)
- The resolution procedure for the `IFNULL()` function differed from that for all other functions derived from `CASE` operations, including `COALESCE()`, which caused incorrect length information to be generated for certain numeric expressions. (Bug #94614, Bug #29463760)
- Queries using `UNION ALL ... LIMIT 1` evaluated an excessive number of rows as shown by `Handler_read_key` and `Handler_read_next`. (Bug #79340, Bug #22292995)

  References: See also: Bug #79040, Bug #22158368, Bug #92994, Bug #28866942.
- The `-DWITH_EXAMPLE_STORAGE_ENGINE=1` `CMake` option was ignored but should not have been. If `-DWITH_EXAMPLE_STORAGE_ENGINE=0` is given, the `EXAMPLE` storage engine is built as a plugin. (Bug #70859, Bug #17772560, Bug #30133062)

References: See also: Bug #18324650.

# Changes in MySQL 8.0.17 (2019-07-22)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.


- Audit Log Notes
- C API Notes
- Character Set Support
- Component Notes
- Configuration Notes
- Debugging Notes
- Deprecation and Removal Notes
- Installation Notes
- Keyring Notes
- Packaging Notes
- Performance Schema Notes
- Plugin Notes
- Security Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Audit Log Notes

- Encryption and decryption operations for encrypted MySQL Enterprise Audit log files use a password stored in the MySQL keyring. Previously, only a single password was stored. Generating a new password made the old password inaccessible, rendering MySQL Enterprise Audit unable to read log files encrypted with the old password. MySQL Enterprise Audit now implements password history in the keyring, which includes password archiving and expiration capabilities. The audit log plugin includes in each encrypted log file name the ID of the password required to read the file. To enable expiration and removal of old archived passwords in the keyring, the new `audit_log_password_history_keep_days` system variable is available. See Configuring Audit Logging Characteristics. (WL #11246)

## C API Notes

- These C API changes were made:
  - `HOSTNAME_LENGTH` was changed from 60 to 255 and moved from `include/mysql_com.h` to `include/my_hostname.h`.

- `USER_HOST_BUFF_SIZE` was moved from `include/mysql_com.h` to `sql/auth/auth_common.h`.

(Bug #29590300)

## Character Set Support

- The `utf8mb4` character set has a new binary collation, `utf8mb4_0900_bin`, which differs from the existing `utf8mb4_bin` binary collation as follows:
  - For collating weights, `utf8mb4_bin` uses code points, possibly with leading zero bytes added, whereas `utf8mb4_0900_bin` uses the `utf8mb4` encoding bytes. The sort order is the same for both collations, but sorting for `utf8mb4_0900_bin` is much faster.
  - The pad attribute for `utf8mb4_bin` is `PAD SPACE`, whereas for `utf8mb4_0900_bin` it is `NO PAD`. Consequently, operations involving `utf8mb4_0900_bin` do not add trailing spaces, and comparisons involving strings with trailing spaces may differ for the two collations.

  For more information, see Unicode Character Sets. (WL #10354)

## Component Notes

- A new `mysql_current_thread_reader` component service is available to enable components to obtain a handle to the current thread. For example, the service enables components to access properties of the current session by passing its thread handle to other services. For information about this service, see the Component Subsystem section of the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html. (WL #12727)

## Configuration Notes

- The source files in the `mysys_ssl` directory have been moved to the `mysys` directory and the `mysys_ssl` library is no longer built. (Bug #29488066)
- MySQL configuration now requires a minimum `CMake` version of 3.5.1. (Bug #29337090)
- The maximum permitted length of host names throughout MySQL has been raised to 255 ASCII characters, up from the previous limit of 60 characters. This applies to, for example, host name-related columns in the data dictionary, `mysql` system schema, Performance Schema, `INFORMATION_SCHEMA`, and `sys` schema; the `MASTER_HOST` value for the `CHANGE MASTER TO` statement; the `Host` column in `SHOW PROCESSLIST` statement output; host names in account names (such as used in account-management statements and in `DEFINER` attributes); and host name-related command options and system variables.

  Caveats:

  - The increase in permitted host name length can affect tables with indexes on host name columns. For example, tables in the `mysql` system schema that index host names now have an explicit `ROW_FORMAT` attribute of `DYNAMIC` to accommodate longer index values.
  - Some file name-valued configuration settings might be constructed based on the server host name. The permitted values are constrained by the underlying operating system, which may not permit file names long enough to include 255-character host names. This affects the `general_log_file`, `log_error`, `pid_file`, `relay_log`, and `slow_query_log_file` system variables and corresponding options. If host name-based values are too long for the OS, explicit shorter values must be provided.

- Although the server now supports 255-character host names, connections to the server established using the `--ssl-mode=VERIFY_IDENTITY` option are constrained by maximum host name length supported by OpenSSL. Host name matches pertain to two fields of SSL certificates, which have maximum lengths as follows: Common Name: maximum length 64; Subject Alternative Name: maximum length as per RFC#1034.

  Applications that expect host names to be a maximum of 60 characters should be adjusted to account for this change. (Bug #13548245, Bug #63814, Bug #27925782, Bug #90601, Bug #27955121, Bug #29584642, Bug #29602081, Bug #94907, WL #12571)

## Debugging Notes

- The MySQL server is a multithreaded application that uses numerous internal locking primitives such as mutexes. To enable detection of lock-acquisition deadlocks and enforcement that runtime execution is free of them, MySQL now supports LOCK_ORDER tooling. This enables a lock-order dependency graph to be defined as part of server design, and server runtime checking to ensure that lock acquisition is acyclic and that execution paths comply with the graph. LOCK_ORDER support includes:

  - A `lock_order_dependencies.txt` file that defines the server lock-order dependency graph.
  - A `WITH_LOCK_ORDER` `CMake` option that configures whether MySQL is built with LOCK_ORDER tooling.
  - A set of system variables that configure LOCK_ORDER tool operation during server execution.
  - A `--lock-order` option for `mysql-test-run.pl` that controls whether to enable the LOCK_ORDER tool during test case execution.

  To use the LOCK_ORDER tool, you must build MySQL from source with tooling enabled. See The LOCK_ORDER Tool. It is intended for debugging the server, not for production use.

## Deprecation and Removal Notes

- **X DevAPI:** For `Collection` objects, the following methods have been deprecated and are scheduled to be removed in a future release:

  - `Collection.find().where()`
  - `Collection.modify().where()`
  - `Collection.remove().where()`

  Any `Collection` code relying on the `.where()` method should be updated and the expression in the `.where()` method should be provided directly in the appropriate `.find()`, `.remove()`, and `.modify()` method. (WL #12902)

- A `mysql_upgrade_info` file created by the `mysql_upgrade` program during a previous upgrade could only be modified the operating system user that executed the `mysql_upgrade` program, causing an upgrade error. A warning is now issued instead of an error, which permits the upgrade operation to proceed. The `mysql_upgrade_info` file is deprecated and will be removed in a future MySQL version. (Bug #29702060, Bug #95165)

- `FLOAT(M,D)` and `DOUBLE(M,D)` syntax to specify the number of digits for columns of type `FLOAT` and `DOUBLE` (and any synonyms) is a nonstandard MySQL extension. This syntax is deprecated and support for it will be removed in a future MySQL version. (Bug #25328973, Bug #84363, WL #12575)

- For string data types, the `BINARY` attribute is a nonstandard MySQL extension that is shorthand for specifying the binary (`_bin`) collation of the column character set (or of the table default character set if no column character set is specified). In MySQL 8.0, this nonstandard use of `BINARY` is ambiguous because the `utf8mb4` character set has multiple `_bin` collations, so the `BINARY` attribute is deprecated and support for it will be removed in a future MySQL version. Applications should be adjusted to use an explicit `_bin` collation instead.

  The use of `BINARY` to specify a data type or character set remains unchanged. (WL #13068)

- The nonstandard C-style `&&`, `||`, and `!` operators that are synonyms for the standard SQL `AND`, `OR`, and `NOT` operators, respectively, are deprecated and support for them will be removed in a future MySQL version. Applications that use the nonstandard operators should be adjusted to use the standard operators.

  **Note**

  Use of `||` is deprecated unless the `PIPES_AS_CONCAT` SQL mode is enabled. In that case, `||` signifies the SQL-standard string concatenation operator).

  (WL #13070)

- The `ZEROFILL` attribute is deprecated for numeric data types, as is the display width attribute for integer data types. Support for `ZEROFILL` and display widths for integer data types will be removed in a future MySQL version. Consider using an alternative means of producing the effect of these attributes. For example, applications could use the `LPAD()` function to zero-pad numbers up to the desired width, or they could store the formatted numbers in `CHAR` columns. (WL #13127)

- The `UNSIGNED` attribute is deprecated for columns of type `FLOAT`, `DOUBLE`, and `DECIMAL` (and any synonyms) and support for it will be removed in a future MySQL version. Consider using a simple `CHECK` constraint instead for such columns. (WL #12391)

- `AUTO_INCREMENT` support is deprecated for columns of type `FLOAT` and `DOUBLE` (and any synonyms) and will be removed in a future MySQL version. Consider removing the `AUTO_INCREMENT` attribute from such columns, or convert them to an integer type. (WL #12595)

- The `SQL_CALC_FOUND_ROWS` query modifier and accompanying `FOUND_ROWS()` function are now deprecated and will be removed in a future MySQL version. As a replacement, considering executing your query with `LIMIT`, and then a second query with `COUNT(*)` and without `LIMIT` to determine whether there are additional rows. For example, instead of these queries:

  ```
  SELECT SQL_CALC_FOUND_ROWS * FROM tbl_name WHERE id > 100 LIMIT 10;
  SELECT FOUND_ROWS();
  ```

  Use these queries instead:

  ```
  SELECT * FROM tbl_name WHERE id > 100 LIMIT 10;
  SELECT COUNT(*) WHERE id > 100;
  ```

  `COUNT(*)` is subject to certain optimizations. `SQL_CALC_FOUND_ROWS` causes some optimizations to be disabled. (WL #12615)

## Installation Notes

- An automatic upgrade at server startup can take some time to complete. For better status notification in systemd environments, the server now sends messages to the system notification socket when upgrades start and end. (Status can be monitored with `server mysqld status`.) (Bug #29493201)

## Keyring Notes

- The `keyring_aws` plugin has been updated to use the latest AWS Encryption SDK and so that it works with OpenSSL 1.1.

  The `keyring_aws_region` variable supports the additional AWS regions supported by the new SDK. Refer to the variable description for a list of supported AWS regions. (WL #12886)

## Packaging Notes

- To reduce the download size and disk footprint of `mysql-community-server` Linux packages, debug binaries and plugins have been moved to separate packages for these platforms:

  - EL8, Fedora: The `mysql-community-server` package with debug binaries and associated plugins is now split into a `mysql-community-server` package without debug binaries or plugins and a `mysql-community-server-debug` package with debug binaries and plugins.
  - Debian: The `mysql-community-server` package with debug binaries and associated plugins is now split into a `mysql-community-server` package without debug binaries or plugins, a `mysql-community-server-debug` package with debug binaries, and a `mysql-community-test-debug` package with debug plugins.

  In all cases, the debug packages are dependent on the corresponding `mysql-community-server` package. (Bug #29769061, Bug #28647754, Bug #92415, Bug #29702765, Bug #95169, Bug #29681301)

## Performance Schema Notes

- Performance Schema version checking at compile time was improved to prevent incompatibilities with the server version. (Bug #29550156)
- Previously, the Performance Schema instrumentation for RWLOCK named priority read/write locks as `rwlock` (thus failing to distinguish plain from priority locks) and did not collect information about the kind of unlock operation performed. Priority read/write locks now are named `prlock`, so that events for them begin with `wait/synch/prlock`. Also, information about unlock operations is provided. (Bug #29270712)

## Plugin Notes

- Not all plugins can operate properly if loaded “early” in the server startup sequence using the `--early-plugin-load` option (that is, before `InnoDB` is initialized). However, `InnoDB` requires keyring backend plugins to operate on encrypted tables. To enable plugins to indicate to the server whether they can be loaded early, a new `PLUGIN_OPT_ALLOW_EARLY` flag is available for use in the plugin descriptor. See Server Plugin Library and Plugin Descriptors. Keyring plugins included in MySQL distributions now have the `PLUGIN_OPT_ALLOW_EARLY` flag enabled because `InnoDB` requires them, but the flag is not limited to keyring plugins. It can be set for other plugins that are able to initialize successfully early in the server startup sequence.

  This flag has no effect on whether a plugin can be loaded at server startup with the `--plugin-load` or `--plugin-load-add` option, or at runtime with the `INSTALL PLUGIN` statement.

  All plugins compiled using MySQL distributions prior to 8.0.17 do not have this flag set. When loading these into pre-8.0.17 servers this does not matter, but attempts to use `--early-plugin-load` to load plugin binaries compiled using pre-8.0.17 MySQL distributions into a 8.0.17 or higher server will fail. The plugins must be recompiled against MySQL 8.0.17 or higher. (Bug #29040456, Bug #93550, WL #12935)

## Security Notes

- For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated to version 1.0.2s. Issues fixed in the new OpenSSL version are described at https://www.openssl.org/news/cl102.txt and https://www.openssl.org/news/vulnerabilities.html. (Bug #29881152)

## X Plugin Notes

- The `createIndex()` method did not support the `DOUBLE(M,D)` syntax for specifying double-precision values due to a regression in MySQL 8.0.16. (Bug #29748841)
- X Protocol's handling of messages with arguments encoded as octets was corrected to support non-scalar data such as an array of strings. (Bug #29721046)
- When host name identity verification was active for SSL connections (`--ssl-mode=VERIFY_IDENTITY`), X Protocol did not check for matches to Subject Alternative Names (SANs) in the server Certificate Authority (CA) certificate. This could lead to connection requests being rejected unnecessarily because they used a valid host name that was specified as an SAN rather than as the certificate's Common Name value. (Bug #29691694)
- When prepared statements were used with X Plugin, using `IN` or `NOT IN` in a modify or find operation produced invalid JSON, resulting in an error. (Bug #29259501)
- On Windows, X Plugin logged some messages that were unnecessary or insufficiently informative. The messages have been removed or improved as appropriate. (Bug #27839153)
- X Plugin's list of SQL functions was out of date, and has been updated to add new functions and remove functions that are no longer available. (Bug #26574971)

## Functionality Added or Changed

- **InnoDB; JSON:** `InnoDB` now supports multi-valued indexes on `JSON` arrays. A multi-valued index is an index in which multiple index records can point to the same data record. This can be useful for indexing `JSON` documents such as `{"user":"Bob","zipcode":[94477,94536]}` in which, if we wish to search all zip codes, it is necessary to have two index records for each zip code in the document. We can create such an index on the `zipcode` array using a `CREATE INDEX` statement such as this one:

  ```
  CREATE INDEX zips ON t1( (CAST(data->'$.zipcode' AS UNSIGNED ARRAY)) )
  ```

  Effectively, this is a functional index using the `CAST()` function, which has been extended with the `ARRAY` keyword to enable casting of JSON arrays to SQL data type arrays. The expression must be a valid JSON expression, and must point to an array in the JSON document in order to be effective. All type specifiers supported by `CAST()` can be used except for `BINARY` and `JSON`. Such usage of the `CAST()` function is supported only by `InnoDB`, and only for creating multi-valued indexes on `JSON` arrays.

  As part of this work, MySQL adds a new function `JSON_OVERLAPS()` as well as a new `MEMBER OF()` operator for working with `JSON` documents, as described here:

  - `JSON_OVERLAPS()` compares two `JSON` documents. If they contain any key-value pairs or array elements in common, the function returns TRUE (1); otherwise it returns FALSE (0). If both values are scalars, the function performs a simple test for equality. If one argument is a JSON array and the other is a scalar, the scalar is treated as an array element. Thus, `JSON_OVERLAPS()` acts as a complement to `JSON_CONTAINS()`, which indicates whether all key-value pairs or array elements are present in both `JSON` documents.

- `MEMBER OF()` tests whether the first operand (a scalar or JSON document) is a member of the JSON array passed as the second operand, returning TRUE (1) if it is, and FALSE (0) if it is not. No type conversion of the operand is performed.

The MySQL optimizer uses multi-valued indexes automatically for any suitable query—that is, a query using in its `WHERE` clause any of `JSON_CONTAINS()`, `JSON_OVERLAPS()`, or `MEMBER OF()` on an array within a JSON column. You can see whether such an index is actually used by checking the output of `EXPLAIN` for the given query.

Multi-Valued Indexes, provides more detailed information about multi-valued indexes, including examples. For more information about `JSON_OVERLAPS()` and `MEMBER OF()`, with examples of use, see Functions That Search JSON Values.

**Note**

`ARRAY` and `MEMBER` now are reserved words and cannot be used as identifiers without identifier quoting.

(WL #8955, WL #8763)

- **Group Replication:** It is now possible to use a remote cloning operation for state transfer to a joining member during distributed recovery. The remote cloning operation lets you add new members to the group without transferring the group's data to the server manually beforehand. To use this functionality, you must install the Clone plugin on the donor and joining member, grant the `BACKUP_ADMIN` permission to the replication user for distributed recovery, and set the new `group_replication_clone_threshold` system variable to an appropriate level. Required Clone plugin settings are now made and the remote cloning operation is now handled automatically. When cloning is complete and the joining member has restarted, any transactions that the group applied while the remote cloning operation was in progress are transferred to the joining member by replication from a donor's binary log to complete distributed recovery. (WL #12827)

- **Microsoft Windows:** A new warning message now reminds DBAs that connections made using the MySQL named pipe on Windows has limited the permissions a connector can request on the named pipe.

  Previously, the `named_pipe_full_access_group` system variable was set to a value that maps to the built-in Windows `Everyone` group (SID S-1-1-0) by default. However, this group is not ideal and should be replaced with a group that restricts its membership for connectors that are unable to request fewer permissions on the MySQL named pipe.

  The new warning is written to the error log at startup if the string value assigned to `named_pipe_full_access_group` is `'*everyone*'` (or the Windows System Language equivalent) and named pipes are enabled. In addition, the warning is written to the error log and raised to the client if the system variable is reset to the `Everyone` group at runtime. (WL #12670)

- **JSON:** MySQL now supports JSON schema validation using two functions `JSON_SCHEMA_VALID()` and `JSON_SCHEMA_VALIDATION_REPORT()`, both of which validate a JSON document against a JSON schema conforming to Draft 4 of the JSON Schema specification. `JSON_SCHEMA_VALID()` returns true if the document validates against the schema and false if it does not. `JSON_SCHEMA_VALIDATION_REPORT()` returns a JSON document containing detailed information about the results of the validation.

  For both of these functions, the following statements apply:

  - `required` attributes are supported.

  - Regular expressions are supported (invalid expressions are silently ignored).
  - External resources in schemas and the `$ref` keyword are not supported.

  For more information, including examples, see JSON Schema Validation Functions. (WL #11999, WL #13005)

- The `time_zone` session variable is now hintable using the `SET_VAR` optimizer hint. (Bug #29776464)
- The minor version of the `libmysqlclient.so` C client library has been increased to 1 (21.0 to 21.1), to signal that new symbols have been added. This was done to correct an oversight in the MySQL 8.0.16 release. To address compatibility concerns, the version of all symbols is unchanged. This means the filename of the library is `libmysqlclient.so.21.1.17`, whereas all symbols inside the library are tagged as 21_0 (unchanged from the 8.0.16 release). (Bug #29584073, Bug #29642146)
- Thanks to Josh Braden, Daniël van Eeden, Simon Mudd, and Zhou Mengkang, who contributed corrections to comments and messages in the MySQL source code. (Bug #29403708, Bug #94464, Bug #29428435, Bug #94527, Bug #29262200, Bug #94049, Bug #29468128, Bug #94625)
- The `mysql` client program now sends `os_user` and `os_sudouser` connection attributes, when available, to indicate the name of the operating system user running the program and the value of the `SUDO_USER` environment variable, respectively. For general information about connection attributes, see Performance Schema Connection Attribute Tables. Thanks to Daniël van Eeden for the contribution on which this feature was based. (Bug #29210935, Bug #93916, WL #12955)
- The optimizer now transforms a `WHERE` condition having `NOT IN (`*`subquery`*`)`, `NOT EXISTS (`*`subquery`*`)`, `IN (`*`subquery`*`) IS NOT TRUE`, or `EXISTS (`*`subquery`*`) IS NOT TRUE` internally into an antijoin, thus removing the subquery. This is similar to the existing `IS NULL` (`Not exists`) outer join optimization; see EXPLAIN Extra Information, for further information.

  In addition, the semijoin materialization can now be used for a `WHERE` condition having `IN (`*`subquery`*`) IS TRUE`, or `EXISTS (`*`subquery`*`) IS TRUE`, or when the `IN` condition belongs to a left join such as `SELECT * FROM t1 LEFT JOIN t2 ON t2.x IN (SELECT * FROM t3)`. See Optimizing IN and EXISTS Subquery Predicates with Semijoin Transformations.

  Also as a result of this work, MySQL is now able to simplify a condition of the form `(x IS TRUE) IS FALSE` as `x IS NOT TRUE`, which can be tested more quickly and optimized more easily than the condition as originally written. (Bug #29027883, WL #4245)
- Password hash values displayed in the `IDENTIFIED WITH` clause of output from `SHOW CREATE USER` may contain unprintable characters that have adverse effects on terminal displays and in other environments. Enabling the new `print_identified_with_as_hex` system variable causes `SHOW CREATE USER` to display such hash values as hexadecimal strings rather than as regular string literals. Hash values that do not contain unprintable characters still display as regular string literals, even with this variable enabled. For compatibility with this change, `CREATE USER` and `ALTER USER` now accept hash values specified either as regular string literals or as hexadecimal strings. (Bug #28053446, Bug #90947, WL #12803)
- In MySQL 8.0, the `lower_case_table_names` variable can only be configured when the MySQL server is initialized. Because a MySQL server installation on Debian and Ubuntu performed using APT initializes the MySQL server for you, there was no opportunity to enable `lower_case_table_names`. To work around this issue, you can now use the `debconf-set-selection` utility to enable

`lower_case_table_names` (set `lower_case_table_names=1`) prior to installing MySQL using APT.

To enable `lower_case_table_names` prior to installing MySQL using APT, execute the following command:

```
shell> sudo debconf-set-selections <<< "mysql-server mysql-server/lowercase-table-names select Enabled
```

(Bug #27948395, Bug #90695)

- The server now checks the SSL server certificate better at startup and writes a warning to the error log if it finds a problem. (Bug #25945005)
- The umask for files created using `SELECT ... INTO OUTFILE` or `SELECT ... INTO DUMPFILE` was changed from 0666 to 0640. The `LOAD_FILE()` function no longer requires files to be world-readable, just readable by the server. (Bug #24513720)
- The `mysqldump` option `--set-gtid-purged` controls whether or not a `SET @@GLOBAL.gtid_purged` statement is added to the `mysqldump` output. The statement updates the value of `gtid_purged` on a server where the dump file is reloaded, to add the GTID set from the source server's `gtid_executed` system variable. A new choice `--set-gtid-purged=COMMENTED` is now available. When this value is set, if GTIDs are enabled on the server you are backing up, `SET @@GLOBAL.gtid_purged` is added to the output (unless `gtid_executed` is empty), but it is commented out. This means that the value of `gtid_executed` is available in the output, but no action is taken automatically when the dump file is reloaded. With `COMMENTED`, you can control the use of the `gtid_executed` set manually or through automation. For example, you might prefer to do this if you are migrating data to another server that already has different active databases. Thanks to Facebook for this contribution. (Bug #94332, Bug #29357665, WL #12959)
- MySQL now supports explicit casts to `DOUBLE`, `FLOAT`, and `REAL` using either of the functions `CAST()` or `CONVERT()`. For more information, see Cast Functions and Operators. (Bug #30524, Bug #11747058, WL #529)
- `InnoDB` now supports redo log archiving. Backup utilities that copy redo log records may sometimes fail to keep pace with redo log generation while a backup operation is in progress, resulting in lost redo log records due to those records being overwritten. The redo log archiving feature addresses this issue by sequentially writing redo log records to an archive file. Backup utilities can copy redo log records from the archive file as necessary, thereby avoiding the potential loss of data.

  For more information, see Redo Log Archiving. (WL #12009, WL #12329, WL #12348)
- To provide additional indexing options for `JSON` data, `InnoDB` now supports multi-valued indexes. A multi-valued index is a secondary index defined on a column that contains an array of values. (WL #8763)
- MySQL now provides a clone plugin that permits cloning `InnoDB` data locally or from a remote MySQL server instance. A local cloning operation stores cloned data on the same server or node where the MySQL instance runs. A remote cloning operation transfers cloned data over the network from a donor MySQL server instance to the recipient server or node where the cloning operation was initiated.

  The clone plugin supports replication. In addition to cloning data, a cloning operation extracts and transfers replication coordinates from the donor and applies them on the recipient, which enables using the clone plugin for provisioning Group Replication members and replication slaves. Using the clone plugin for provisioning is considerably faster and more efficient than replicating a large number of transactions. Group Replication members can also be configured to use the clone plugin as an alternative method of recovery, so that members automatically choose the most efficient way to retrieve group data from seed members.

For more information, see The Clone Plugin, and Cloning for Distributed Recovery. (WL #9209, WL #9210, WL #9211, WL #9212, WL #11636, WL #9682)

- Compatibility policies for member versions in groups now consider member's MySQL release version. Previously, only the minor version was considered. Using the release version means Group Replication can better maintain replication safety for mixed version groups during group reconfigurations and upgrade procedures.

  The compatibility policies are applied when a member joins the group, either for the first time or following its upgrade, when a donor is selected for state transfer, and when a primary member election takes place. Members running MySQL 8.0.16 or earlier take into account only the major version in these situations. For primary member election, so that all members come to the same decision, members running MySQL 8.0.17 or later adjust their policies to match members running lower releases if any are in the group.

  In a multi-primary mode group where members running multiple MySQL Server versions are online, for example during a rolling online upgrade procedure, Group Replication now automatically manages the read-write and read-only status of members running releases. If a member leaves the group, the members running the version that is now the lowest are automatically set to read-write mode. When you change a mixed version group that was running in single-primary mode to run in multi-primary mode, using the `group_replication_switch_to_multi_primary_mode()` function, members are set automatically to read-write or read-only mode depending on their MySQL server versions.

  These improved compatibility policies influence the behavior of group members during an online upgrade procedure from one patch version to another, in the same ways as the behavior during upgrades from one major version to another was influenced previously. For a multi-primary mode group, the number of members in read-write mode is reduced during the upgrade procedure; their read-write status is now handled automaticaly when the upgrade is otherwise complete. For a single-primary mode group, if you want the primary to remain as the primary, it must be upgraded last. (WL #12826)

- Data that is held in the binary log transaction and statement caches during a transaction is in unencrypted format in the memory buffer that stores the cache. The data is written to a temporary file on disk if it exceeds the space available in the memory buffer. From MySQL 8.0.17, when binary log encryption is active on the server (`binlog_encryption=ON`), the temporary files used for the binary log cache are now encrypted using AES-CTR (AES Counter mode) for stream encryption. Because the temporary files are volatile and tied to a single process, they are encrypted using single-tier encryption, using a randomly generated file password and initialization vector that exist only in memory and are never stored on disk or in the keyring. After each transaction is committed, the binary log cache is reset: the memory buffer is cleared, any temporary file used to hold the binary log cache is truncated, and a new file password and initialization vector are randomly generated for use with the next transaction. This reset also takes place when the server is restarted after a normal shutdown or an unexpected halt. (WL #12079)

- An incomplete SQL predicate has the form `WHERE value`, in which `value` is a column name or constant expression and no comparison operator is used. MySQL now rewrites any predicate of this type internally as `WHERE value <> 0` during the contextualization phase, so that the query resolver, query optimizer, and query executor need work only with complete predicates. The principal visible effect of this change is that, for Boolean values, `EXPLAIN` output now shows `true` and `false`, rather than `1` and `0`. (WL #12358)

- `InnoDB` parallel read thread performance for large data sets was improved through better utilization of read threads, through a reduction in read thread I/O for prefetch activity that occurs during parallel scans, and through support for parallel scanning of partitions.

  The parallel read thread feature is controlled by the `innodb_parallel_read_threads` variable. The maximum setting is now 256, which is the total number of threads for all client connections. If the thread limit is reached, connections fall back to using a single thread. (WL #12978)

- `mysqlbinlog` now supports the `--compress` (or `-C`) option to enable compression in the client/server protocol. (WL #2726)

## Bugs Fixed

- **NDB Cluster:** Compile times for NDB Cluster using the included `compile_cluster` build script have been improved by removing options for software components not strictly necessary for running MySQL Cluster. (Bug #29355872)

- **NDB Cluster:** Attempting to change an `NDB` table's column properties (such as `COLUMN_FORMAT`) using `ALTER TABLE ALGORITHM=INPLACE` was rejected, which is correct behavior, but which raised a misleading error message. (Bug #28929906, Bug #27645777)

- **InnoDB:** A failure occurred when attempting to kill a process that was executing a row count. (Bug #29939617)

- **InnoDB:** Due to a regression introduced in MySQL 8.0.14, in-place upgrade on a case sensitive file system from MySQL 5.7 or a MySQL 8.0 release prior to MySQL 8.0.14 to MySQL 8.0.16 failed for instances with partitioned tables and `lower_case_table_names=1`. The failure was caused by a case mismatch issue related to partitioned table file names. The fix that introduced the regression was reverted, which permits upgrades to MySQL 8.0.17 from MySQL 5.7 or MySQL 8.0 releases prior to MySQL 8.0.14 to function as normal. However, the regression is still present in the MySQL 8.0.14, 8.0.15, and 8.0.16 releases.

  In-place upgrade on a case sensitive file system from MySQL 8.0.14, 8.0.15, or 8.0.16 to MySQL 8.0.17 fails with the following error when starting the server after upgrading binaries or packages to MySQL 8.0.17 if partitioned tables are present and `lower_case_table_names=1`:

  ```
  Upgrading from server version version_number with
  partitioned tables and lower_case_table_names == 1 on a case sensitive file
  system may cause issues, and is therefore prohibited. To upgrade anyway,
  restart the new server version with the command line option 'upgrade=FORCE'.
  When upgrade is completed, please execute 'RENAME TABLE part_table_name
  TO new_table_name; RENAME TABLE new_table_name
  TO part_table_name;' for each of the partitioned tables.
  Please see the documentation for further information.
  ```

  If you encounter this error when upgrading to MySQL 8.0.17, perform the following workaround:

  1. Restart the server with `--upgrade=FORCE` to force the upgrade operation to proceed.

  2. Identify partitioned table file names with lowercase partition name delimiters (`#p#` or `#sp#`):

     ```
     SELECT FILE_NAME FROM INFORMATION_SCHEMA.FILES
         WHERE FILE_NAME LIKE '%#p#%' OR FILE_NAME LIKE '%#sp#%';
     ```

  3. For each file identified, rename the associated table using a temporary name, then rename the table back to its original name.

     ```
     mysql> RENAME TABLE table_name TO temporary_table_name;
     mysql> RENAME TABLE temporary_table_name TO table_name;
     ```

4. Verify that there are no partitioned table file names with lowercase partition name delimiters (an empty result set should be returned).

```
mysql> SELECT FILE_NAME FROM INFORMATION_SCHEMA.FILES
    ->     WHERE FILE_NAME LIKE '%#p#%' OR FILE_NAME LIKE '%#sp#%';
Empty set (0.00 sec)
```

5. Run `ANALYZE TABLE` on each renamed table to update the optimizer statistics in the `mysql.innodb_index_stats` and `mysql.innodb_table_stats` tables.

Because of the regression still present in the MySQL 8.0.14, 8.0.15, and 8.0.16 releases, importing partitioned tables from MySQL 8.0.14, 8.0.15, or 8.0.16 to MySQL 8.0.17 is not supported on case sensitive file systems where `lower_case_table_names=1`. Attempting to do so results in a "Tablespace is missing for table" error. (Bug #29823032, Bug #29917793, Bug #95834)

References: This issue is a regression of: Bug #26925260.

- **InnoDB:** `lock_sys` mutex contention caused by lock-wait functions (`lock_wait_suspend_thread()` and `lock_wait_table_release_slot()`) was reduced. (Bug #29814339)
- **InnoDB:** The `fseg_n_reserved_pages_low()` function, which determines the number of pages reserved by a segment, did not validate results read from the segment inode. (Bug #29761998)
- **InnoDB:** Creation of the transaction rollback list (`hit_list`) was decoupled from lock acquisition calls (`lock_rec_lock` calls) to permit the use of different latching schemes. (Bug #29753800)
- **InnoDB:** Disabling Performance Schema consumers caused an `ALTER TABLESPACE ... ENCRYPTION` operation to assert. (Bug #29646974, Bug #95005)
- **InnoDB:** Error messages were revised to remove references to `.frm` files, which are not used in MySQL 8.0. (Bug #29639655)
- **InnoDB:** It was possible for a background thread to check the encryption status of an undo tablespace before the undo tablespace was fully initialized and the encryption flag set. (Bug #29600309)
- **InnoDB:** A table name parsing function call returned false when parsing serialized dictionary information (SDI) tables names, which are not formatted to include a database name. The buffer that holds the database name remained uninitialized, causing Valgrind errors. (Bug #29550527)
- **InnoDB:** The space reserved in the mini-transaction (mtr) log buffer for dynamic metadata logging was insufficient. (Bug #29524260)
- **InnoDB:** Inaccuracies in the Contention-Aware Transaction Scheduling (CATS) implementation raised a signed integer overflow error in an UBSan build of MySQL. (Bug #29508517, Bug #91959)
- **InnoDB:** Insufficient memory barriers in the rw-lock implementation caused deadlocks on ARM.

  Thanks to Yibo Cai from Arm Technology for the contribution. (Bug #29508001, Bug #94699)
- **InnoDB:** The `INFORMATION_SCHEMA.INNODB_TABLESPACES ENCRYPTION` column was not updated after enabling undo tablespace encryption. (Bug #29492911, Bug #94665)
- **InnoDB:** A relocated table could not be accessed due to incorrect parsing of a schema or table name that included a forward slash character (/), which the server incorrectly interpreted as a directory separator. (Bug #29492113)
- **InnoDB:** Various fixes and revisions were applied to the `InnoDB` memcached source code. (Bug #29485891)

- **InnoDB:** To enable global access, the value of `innodb_directories` variable is now stored internally as a global variable instead of a static variable. (Bug #29471990)
- **InnoDB:** In debug builds, the `thd_innodb_tmpdir()` function did not accept a NULL argument. (Bug #29471846)
- **InnoDB:** A subtraction operation in the file space allocation code incorrectly stored the result as an unsigned variable, which raised an assertion failure. (Bug #29466680)
- **InnoDB:** After moving or deleting default undo tablespaces and restarting the server with a new `innodb_undo_directory` value, MySQL recreated the undo tablespaces in the new location but failed to update the undo directory path in the data dictionary. (Bug #29461900)
- **InnoDB:** A previously freed LOB page was accessed while rolling back a transaction during recovery. (Bug #29440408)
- **InnoDB:** A request was issued during recovery to read pages into the buffer pool when there were no pages to be read. A check was added to avoid the unnecessary read request. (Bug #29440208)
- **InnoDB:** A regression introduced in MySQL 8.0.14 caused a “Invalid (old?) table or database name” error when creating a partitioned table on a MySQL instance with `lower_case_table_names=1`. The change that caused the regression has been reverted. (Bug #29426720, Bug #94519)

  References: This issue is a regression of: Bug #26925260.
- **InnoDB:** A `FULLTEXT` index table created in MySQL 5.6 in a database with a hyphen in its name caused a startup failure after upgrading from MySQL 5.7 to MySQL 8.0. Tablespace file paths for `FULLTEXT` auxiliary tables were not found in the data dictionary, and the hyphen in the database name was not handled correctly in subsequently generated file paths. (Bug #29411899, Bug #94431)
- **InnoDB:** There was potential for data loss to occur if the redo log was not logically empty and comprised a single block, and the server exited during recovery after an insert buffer merge generated new redo records but before the new records could be flushed to disk. (Bug #29411832, Bug #94448)
- **InnoDB:** `InnoDB` returned an unknown generic error when attempting to create a tablespace for which the path and file name exceeded the `MAX_PATH` limit on Windows. `InnoDB` now returns a more meaningful error. (Bug #29341634)
- **InnoDB:** An undo tablespace file was not found after being moved to a different directory. (Bug #29328158)
- **InnoDB:** The server failed to start with an `innodb_buffer_pool_size=default` setting. The default value was not checked for compatibility with dependent system variable settings. (Bug #29267814, Bug #94065)
- **InnoDB:** The `CREATE TABLESPACE ... ADD DATAFILE` clause no longer permits circular directory references. For example, the circular directory reference (`/../`) in the following statement is not permitted:

  ```
  CREATE TABLESPACE ts1 ADD DATAFILE ts1.ibd 'any_directory/../ts1.ibd';
  ```

  An exception to this restriction exists on Linux, where a circular directory reference is permitted if the preceding directory is a symbolic link. For example, the data file path in the example above is permitted if *`any_directory`* is a symbolic link. (It is still permitted for data file paths to begin with '`../`'.)

  To avoid upgrade issues, remove any circular directory references from tablespace data file paths before upgrading to MySQL 8.0.17 or higher. To inspect tablespace paths, query the `INFORMATION_SCHEMA.INNODB_DATAFILES` table. (Bug #29157265)

- **InnoDB:** Manually changing the system time while the MySQL server was running caused page cleaner thread delays. (Bug #29138644, Bug #93708)
- **InnoDB:** `UPDATE` statements did not always disable semi-consistent reads properly when encountering errors, which could lead to assertion errors in debug mode. (Bug #29047894)
- **InnoDB:** When purging a deleted row, the logic that governs lock inheritance did not correctly determine the type of lock that should be inherited to satisfy constraint checks by active transactions. (Bug #29004362)
- **InnoDB:** An unnecessary read lock was taken on implicitly opened data dictionary tables when executing a prepared statement in `LOCK TABLES` mode. (Bug #28875646)
- **InnoDB:** During log application, after an `OPTIMIZE TABLE` operation, `InnoDB` did not populate virtual columns before checking for virtual column index updates. (Bug #28834208)
- **InnoDB:** An operation that copied data from the clustered index was performed incorrectly, causing spatial indexes to use spatial rows with stale pointers to the clustered index. (Bug #28758961)
- **InnoDB:** An `INSERT` operation involving a generated virtual `BLOB` column resulted a secondary index being updated with an incorrect value. (Bug #28652826)
- **InnoDB:** Configuring `innodb_data_file_path` and `innodb_temp_data_file_path` using `SET PERSIST_ONLY = default` incorrectly set the variable values to NULL. (Bug #28590014)
- **InnoDB:** A `CREATE TABLE ... REPLACE SELECT` operation raised a `lock_rec_get_rec_not_gap(lock)` assertion failure. The operation set a flag on the transaction object indicating that a `REPLACE` operation was requested, but did not clear the flag before updating dependent views, causing a subsequent `INSERT` operation to be interpreted as a `REPLACE` operation, leading to the wrong row locks being taken. (Bug #28523025, Bug #92068)
- **InnoDB:** With `super_read_only` enabled, an attempted `RENAME TABLE` operation on a temporary table raised an assertion instead of returning an error. (Bug #28490368, Bug #91975)
- **InnoDB:** Valgrind errors were reported due to uninitialized bytes read during a virtual index prefix search. (Bug #28184025)
- **InnoDB:** `InnoDB` initialization failed when attempting to create an system tablespace greater than 2GB in size. (Bug #27538464)
- **InnoDB:** A full-text cache lock taken when data is synchronized was not released if the full-text cache size exceeded the full-text cache size limit. (Bug #25289359)
- **InnoDB:** The `INNODB_METRICS` `metadata_table_reference_count` counter reported a negative value. (Bug #20584149, Bug #75966)
- **InnoDB:** Client sessions using different `auto_increment_increment` values while performing concurrent insert operations could cause a duplicate key error. (Bug #15851528, Bug #67526)

  References: Reverted patches: Bug #14049391, Bug #65225.
- **Partitioning:** For partitioned tables, an `ALTER TABLE` statement could cause incorrect query results under these conditions:
  - The statement swapped columns by renaming them, either directly with `RENAME COLUMN` or by replacement with `DROP COLUMN`/`ADD COLUMN`.
  - A swapped column was used in a partitioning expression.

- The alteration was performed as an in-place operation that did not redistribute rows between partitions.

Such column renames are now prohibited unless the same `ALTER TABLE` statement satisfies one of these conditions:

- The statement makes the table nonpartitioned.
- The statement redefines the table partitioning or partitioning expression (which causes a table rebuild that redistributes rows). This enables supporting existing scenarios in which partitioning expressions are updated to follow column renaming.
- Partitioning is specified using `PARTITION BY KEY()` with an empty column list. This partitions using the primary key, which tracks column renaming.

(Bug #29541665, Bug #94792)

- **Partitioning:** `ALTER TABLE ... EXCHANGE PARTITION` failed with the error `Non matching attribute 'ROW_FORMAT' between partition and table` when the partitioned table had partitions using different row formats, even when the partition to be exchanged used the same row format as the non-partitioned table. (Bug #28687608)
- **Replication:** When events generated by one MySQL server instance were written to the binary log of another instance, the second server implicitly assumed that the first server supported the same number of binary log event types as itself. Where this was not the case, the event header was handled incorrectly. The issue has now been fixed. Thanks to Facebook for the contribution. (Bug #29417234, Bug #94500)
- **Replication:** If a `LOCK INSTANCE FOR BACKUP` statement was used to acquire an instance-level backup lock, then a `STOP SLAVE` statement was issued, a deadlock could be created with the SQL thread waiting on the backup lock and the `STOP SLAVE` statement waiting on the SQL thread to complete its current action. To prevent this situation, the `STOP SLAVE` process now tries to acquire the backup lock before proceeding, and returns an error if the lock cannot be acquired. (Bug #29386503, Bug #93649)
- **Replication:** From MySQL 8.0.13, if any replication channel has open temporary tables, the binary logging format cannot be changed using `SET @@global.binlog_format` or `SET @@persist.binlog_format`. Previously, if this operation was attempted after the new restriction was implemented, the wrong error message was returned to the client (referencing a running replication channel applier as the issue, rather than an open temporary table). The appropriate error message is now returned. (Bug #29370024, Bug #94340)
- **Replication:** Binary log checksums were handled incorrectly when de-serializing a format description event. (Bug #29355110)
- **Replication:** With row-based replication in use, when the replication applier thread unpacked a row change event, index values for any functional indexes were calculated for both the “before” image and the “after” image. In the case of the “before” image, the values were not necessary. This calculation has therefore been removed for the before image in order to optimize row unpacking. (Bug #29304076)
- **Replication:** When a `MEMORY` table is implicitly deleted on a master following a server restart, the master writes a `DELETE` statement to the binary log so that slaves also empty the table. This generated event now includes a comment in the binary log so that the reason for the `DELETE` statement is easy to identify. Thanks to Daniël van Eeden for the contribution. (Bug #29157796, Bug #93771)

- **Replication:** If an invalid starting offset was provided in a `SHOW BINLOG EVENTS FROM` statement, the invalid offset was returned in place of the correct starting position for the first returned event. (Bug #29039732, Bug #93544)

- **Replication:** Replication could stop with a “record not found” error when the value `INDEX_SCAN,HASH_SCAN` (the default in MySQL 8.0) was set for the `slave_rows_search_algorithms` system variable, and an update event contained two updates to the same row in a table that did not have a unique key, meaning that the hash scan was used. In this situation, the second update was missed by the hash scan due to the row change. Now, after updating a row, the hash scan operation searches for the updated row in the hash map, and applies any further update.

  When the value `TABLE_SCAN,HASH_SCAN` is set for the `slave_rows_search_algorithms` system variable, so that the search cannot use an index, the “record not found” error can occur in the above situation whether or not the table has a unique key. Also, with this setting, when a hash scan is used on a table that has a unique key, in the case of an update event containing two row updates that are order-dependent, the updates might be applied out of order, causing replication to stop with a duplicate key error. To avoid these issues, the documentation has been updated to state that the value `TABLE_SCAN,HASH_SCAN` should not be used. (Bug #28846386)

- **Replication:** When binary logging is enabled on a replication slave, the combination of the `--replicate-same-server-id` and `--log-slave-updates` options on the slave can cause infinite loops in replication if the server is part of a circular replication topology. (In MySQL 8.0, binary logging is enabled by default, and slave update logging is the default when binary logging is enabled.) However, the use of global transaction identifiers (GTIDs) prevents this situation by skipping the execution of transactions that have already been applied. The restriction on this combination of options has therefore now been removed when `gtid_mode=ON` is set. With any other GTID mode, the server still does not start with this combination of options. As a safeguard against creating the problem situation after the server has started, you now cannot change the GTID mode to anything other than `ON` on a running server that has this combination of options set. Thanks to Facebook for the contribution. (Bug #28782370, Bug #92754)

- **Replication:** In query log events in the binary log, the thread ID used for the execution of `DROP TABLE` and `DELETE` statements was identified incorrectly or not at all. On a multi-threaded replication slave, where temporary tables were involved (which require the correct thread ID as they are session specific), this omission resulted in errors when using `mysqlbinlog` to replay the binary log for point-in-time recovery. The thread ID is now set correctly. (Bug #28642318, Bug #92398)

- **Replication:** When a trigger invoked an `INSERT` or `UPDATE` statement that set a column to its default value, and the `DEFAULT` expression of that column was non-deterministic, the expected warning was not raised when the trigger fired under statement-based replication. In addition, if the binary logging format was `MIXED`, the non-deterministic statements were logged in the format used for `STATEMENT` rather than that used for `ROW`.

  The statement that causes the trigger to fire checks at resolution time whether any of the triggered statements are non-deterministic. At this time, the triggered statements have been parsed, but not resolved, so the only check that can be performed is whether the triggered statements directly reference any non-deterministic operators. When the non-deterministic operator is used by a `DEFAULT` expression, the non-determinism is not visible until the triggered statement has been resolved, which happens when the trigger fires.

  This is fixed by adding an extra check when determining the logging format, where a statement is flagged as unsafe if any of its substatements can write to a table that has a column with a non-deterministic `DEFAULT` expression. Since it is not yet known at this point whether the `DEFAULT` expression will be used by the substatement, this check flags the statement as unsafe even if the

substatement provides an explicit value for the column with a non-deterministic `DEFAULT` expression. (Bug #28297486)

- **Replication:** When a slave server logs master status and connection information to a table (`master_info_repository=TABLE`), which is the default in MySQL 8.0, the `mysql.slave_master_info` table was not being updated on shutdown if the server was in super read only mode (`super_read_only=ON`). No error was written to the error log at this time, but replication failed after server startup because the master log file and master log position information was out of date. The thread that updates the master info log at shutdown is now excluded from read-only checks like other replication threads are, so it can update the table even if the server is in super read only mode. Error handling for a slave that is shutting down has also been improved so that any failure to write to the slave status logs results in an error in the error log. (Bug #27675107, Bug #89987)

- **Replication:** If a replication slave tried to connect to the master using an incorrect user name, host, or port, the original error message specifying the reason for the connection failure was overwritten with a generic message. The issue has now been corrected in the output from the `SHOW SLAVE STATUS` statement and in the Performance Schema table `replication_connection_status`. (Bug #26580064)

- **Group Replication:** When message fragmentation is in use for large Group Replication messages (which is available, and the default, from MySQL 8.0.16), if a fragmented message sent by the group member with the highest node identifier in XCom was partially delivered, and one or more members then left the group prior to the delivery of the remaining message fragments, reassembly of the message caused Group Replication to stop working. The loss of members meant that the node identifier of the original sender was no longer valid in the new view of the group. To correct this issue, reassembly of fragmented messages now uses the delivery information from the last fragment that was delivered, which reflects the new situation after the view change, rather than from the first fragment that was delivered, which reflects the old situation before the view change. (Bug #29716639)

- **Group Replication:** The error message that is issued for a discrepancy between the number of group members and the auto-increment interval incorrectly referred to the `group_replication_auto_increment_increment` system variable, instead of the `auto_increment_increment` system variable. The value of `auto_increment_increment` is changed to the value specified by `group_replication_auto_increment_increment` when Group Replication starts, but only if `auto_increment_increment` and `auto_increment_offset` have their default values, and from MySQL 8.0, only in multi-primary mode. The value of `auto_increment_increment` was always the value that was checked for the error message, and it has now been corrected to give the accurate system variable name. (Bug #29542425)

- **Group Replication:** Group Replication cannot be started following a MySQL Server upgrade that uses the `MINIMAL` option (`--upgrade=MINIMAL`), which does not upgrade system tables on which this feature depends. Previously, in this situation, the server waited indefinitely for Group Replication to start. This is now handled correctly by unblocking the waiting thread and issuing the expected error `ER_GRP_RPL_START_GRP_RPL_FAILED`. (Bug #29423358, Bug #94515)

- **Group Replication:** A change in the processing of suspicions by a member that is leaving the group meant that recovery failed because due to a circular dependency between the recovery failure and the view change notification. Now, if an error makes recovery impossible, the Group Communication System handles this as follows:

  1. The member leaves the group

  2. the view change is applied

3. The recovery thread is terminated.

(Bug #29417365, Bug #29628909)

- **Group Replication:** In Group Replication, joining members could wrongly identify themselves as incompatible with an existing replication group even if there were members at the same version already in the group, because they checked against all other members, including the member at the highest version. Joining members also included their own version in the compatibility check. Now, joining members only compare themselves with the existing group member at the lowest version, and do not count their own version. (Bug #29390946, Bug #94429)

- **Group Replication:** Functions for configuring an online group sometimes did not return an error when an issue arose during execution. These functions also now check whether the Group Replication plugin is stopping before they start to initialize. (Bug #28978767, Bug #93372)

- **Group Replication:** The XCom group communication engine did not handle out of memory errors in an appropriate way. If memory could not be allocated to make a copy of the payload for a message, an error was logged but the message was still sent, with a null payload. The Group Communication System (GCS) on the receiving member discarded the message as empty, and the XCom instance on the receiving member accepted this action and did not retry, resulting in the message effectively being skipped. This caused the GTID set on the receiving member to diverge from the group, leading to replication errors. XCom now terminates gracefully if it experiences an out of memory error, so that this situation cannot occur. (Bug #28702320)

- **macOS:** For macOS installations performed using DMG packages, `launchd` operation was problematic:

  - Previously, `SHUTDOWN` caused a restart if MySQL was configured to start at boot time via the preference pane. This also affected the `mysqladmin shutdown` command. Server shutdown initiated in these ways now works correctly.

  - Previously, `RESTART` did not work. It now works correctly.

  - Previously, unexpected server exits did not cause automatic restart if the server was not configured to start at boot time. Exits with a nonzero exit status now cause a restart regardless of boot-time startup configuration.

  (Bug #29789857)

- **JSON:** `MAX()` and `MIN()` used on expressions returning `JSON` data sometimes compared these values as strings rather than `JSON` values, which caused unexpected results; this was particularly evident when the `JSON` values were numbers.

  This was due to the fact that `GROUP BY` when using temporary tables with indexes did not compare `JSON` values correctly. (Bug #28947381)

- **JSON:** `JSON_TABLE()` returned the error `Unknown database ''` when executed from a stored function.

  The root cause of this issue was that, when merging tables from a select that used `JSON_TABLE()`, MySQL checked only for derived tables. This caused the result table returned by `JSON_TABLE()` to be noted as a regular table, so that when attempting to execute the query, the server failed to open it. Now MySQL checks whether the table to be added is not an internal table, that is, not a derived table, a `JSON_TABLE()` result table, or a reference to a recursive common table expression. (Bug #92976, Bug #28851656)

- The `WITH ADMIN` option for `GRANT` statements sometimes was not handled properly. (Bug #29900772)

- Some foreign key error messages could differ depending on whether the user had the `GRANT OPTION` privilege. (Bug #29868844)
- During upgrade operations, upgrades of the help tables failed if autocommit was disabled. (Bug #29865428, Bug #95620)
- Fetching dynamically allocated dictionary objects into a vector during upgrade while operating with a small `table_open_cache` size caused data dictionary tables to be reopened, triggering a garbage collection mechanism that incorrectly freed collected objects. Subsequent attempts to access the freed objects caused a segmentation fault. (Bug #29823053)
- For upgrades from MySQL 5.7 to 8.0, the upgrade process did not close the `innodb_*_stats_backup57.ibd` files in the `mysql` system schema before removing them, resulting in errors for subsequent file system operations. (Bug #29791350)
- `mysqld --initialize` would fail if the file system was mounted at the data directory mount point and a `lost+found` file or directory was present. The `lost+found` file or directory is now ignored during data directory initialization. (Bug #29780434)
- MySQL upgrades did not assign the `AUDIT_ADMIN` privilege to accounts that had the `SUPER` privilege. (Bug #29770732)
- The `REGEXP_REPLACE()` function did not handle empty strings correctly in all cases. (Bug #29763554)
- An overly strict assertion could be raised during sorting of stored program local objects. (Bug #29759547, Bug #95062)
- The `group_replication_get_communication_protocol()` function, which is used to query the group's communication protocol version, failed if any group members were in RECOVERING state, which was an unnecessary restriction. The function can now be used provided that the member where it runs is in ONLINE state, and is in contact with a majority of the group. (Bug #29754967, Bug #95306)
- For some arguments to `REPEAT()`, maximum length calculations were not always handled correctly. (Bug #29739778)
- `UPDATE` statements for tables with `CHECK` constraints could fail to enforce the constraints. (Bug #29706621, Bug #95189)
- For installation from RPM or Debian packages, if the `mysql_upgrade_info` file in the data directory is found to exist but is owned by `root`, it is now changed to the same owner as the data directory. The correct SELinux file context is also set. (Bug #29704041)
- Installing from RPM packages could result in an error log with incorrect permissions. (Bug #29702462)
- A replication group member that was unable to contact a majority of the group before the `group_replication_unreachable_majority_timeout` setting was reached, and then exhausted the number of auto-rejoin attempts specified by the `group_replication_autorejoin_tries` system variable without successfully rejoining, was sometimes unable to carry out the action specified by the `group_replication_exit_state_action` system variable. The member tries to carry out the action using the current session, which sometimes failed. The member now ensures that the session is initialized before using it to connect to the server to carry out the exit state action. (Bug #29698754, Bug #95151)
- When performing a upgrade, server-side SQL statements that update data dictionary tables used a non-default collation, causing an upgrade failure. (Bug #29697670, Bug #95144)
- Starting the server with `--skip-grant-tables` and invoking `ROLES_GRAPHML()` could lead to a server exit. (Bug #29681975)

- `SELECT DISTINCT` when using fixed-length keys did not deduplicate records properly in all cases. (Bug #29628699)
- Client programs that used asynchronous C API functions could access freed memory after occurrence of a fatal error such as connection failure or receipt of a too-large packet. (Bug #29596244)
- The `ST_AsWKB()` function could fail to perform proper error checking. (Bug #29594287)
- For failed data directory initialization, the server could produce a message indicating that the directory could be removed, even if the server itself did not create it. The message now indicates that only files in the directory created by the server can be removed. (Bug #29594082, Bug #94880)
- In MySQL 8.0, the `keyring_encypted_file` plugin could not read a file created by a MySQL 5.7 `keyring_encypted_file` plugin. (Bug #29588345)
- In builds with Undefined Behavior Sanitizer enabled, runtime errors occurred in `my_strtoll10_mb2()` and `val_decimal()`. (Bug #29585648, Bug #29594951)
- Some diagnostics produced by the server while checking grant table structure at startup were written as errors rather than warnings. (Bug #29558993)
- The `comp_err` utility could read uninitialized data. (Bug #29550442)
- Enabling audit log encryption could cause a server exit. (Bug #29549327)
- RPM package obsoletes were updated to enable successful upgrades from MariaDB to MySQL on EL8. (Bug #29549127, Bug #29623146, Bug #29623201, Bug #29659212)
- In a join of the form `LEFT JOIN t ON column WHERE t.x=0 IS NOT TRUE`, MySQL evaluated `IS NOT TRUE` as `FALSE` for a `NULL` argument in a null-complemented row, and rejected it. This caused an invalid conversion of the left join to an inner join, so that there were missing rows in the result. The same issue affected joins of the same form but having `IS NOT FALSE` in the `WHERE` condition. (Bug #29540230)
- If the `read_only` system variable was enabled, its value improperly was applied against updates to the character set and collation data dictionary tables. (Bug #29533590, Bug #94769)
- The functions `STATEMENT_DIGEST()` and `STATEMENT_DIGEST_TEXT()` used `character_set_client` while parsing the function's argument instead of the character set of the argument. (Bug #29526571)
- The server now tracks data dictionary upgrades to ensure that `INFORMATION_SCHEMA` views are recreated when data dictionary tables are changed. (Bug #29513265)
- Execution of `STATEMENT_DIGEST()` or `STATEMENT_DIGEST_TEXT()` could have unintended side effects on data in the `performance_schema.events_statements_summary_by_digest` table. (Bug #29512067)
- The implementation for `CREATE TABLE` did not provide the table a clearly defined starting cursor when no seeks or writes were performed. (Bug #29511318)
- When a table is `const` optimized, any predicate terms in the join's `ON` condition should also be evaluated on the tables marked as `const` to make sure that the row qualifies for the `ON` condition.

  The optimizer failed to mark a table as `const` when it was handled as having an impossible `ON` condition, in which case the resulting row from this table is a `NULL` extended row.

  This fix sets the `const_table` flag for the table and marks the row as a `NULL` row. It also changes `EXPLAIN` output for a table optimized as having an impossible ON condition, where the resulting `NULL`

values from this row are now propagated into the rows column values, as with other `const` rows. (Bug #29493830)

- A query containing an `IN` subquery could return superfluous rows when `optimizer_switch` did not include `firstmatch=on`. (Bug #29493026)
- RPM package detection of whether the operating system is EL6 or EL7 failed on some systems. (Bug #29492896)
- SQL layer validation of keys stored in storage-engine-private data fields in the data dictionary was disabled to permit storage engines to add new keys as required. Storage engines are now responsible for key validation. (Bug #29491593, Bug #94667)
- Checks for `NULL` returned an incorrect result for some expressions comparing rows. (Bug #29491083)
- When optimizing a table join which included `ORDER BY` and `LIMIT`, the optimizer modified the select limit on the first table by taking the fanout of the entire join into consideration. The fanout was calculated using the `filter_effect` for each table, which can be set to `COND_FILTER_STALE` (-1) to indicate an unknown value. This value was not actually checked for by the optimizer, which could result in a negative fanout, leading to an unexpected rows value of 0 in the `EXPLAIN` output for the query.

  Now `COND_FILTER_STALE` (`filter_effect` unknown) is specifically checked for, and, if it is found, the fanout is also assumed to be unknown, and the number of rows to be selected from the first table is thus not modified by the fanout for the entire join. (Bug #29487181, Bug #29531472)
- An implicit assumption was made for the `LEAST()` and `GREATEST()` functions that arguments to either of these were all signed or all unsigned values. (Bug #29467577)
- Some arguments for the `BIT_OR()` function were not always handled correctly. (Bug #29459549)
- For debug builds, mishandling of the `user_attributes` column of the `mysql.user` system table could cause a server exit. (Bug #29451897)
- MySQL Cluster-specific Debian debug packages had an incorrect path to the debug symbols. (Bug #29446947)
- Compilation failure occurred when building with Protobuf 3.7.0. (Bug #29436791, Bug #94543)
- On Debian and Ubuntu, MySQL packages did not enable `mysql.service` after upgrades from native MySQL packages. (Bug #29435592)
- The server did not properly close shared-memory connections when an error occurred, which could result in unexpected server behavior. (Bug #29435426)
- The internal `JOIN_CACHE::join_record()` method attempted to leave all tables on which it operated in the same state with regard to the `has_row` and `null_row` flags by saving the state upon entry and restoring the appropriate flags when on completion.

  The issue addressed here arose after these flags had been restored, due to the fact that `restore_last_record()` was also called to restore any records that had been processed, which could lead to setting back a `null_row` flag that had just been restored based on the saved state. This is fixed by calling `restore_last_record()` before the saved states are restored, rather than after, as well as making sure that `reset_null_row()` is also called if the saved state indicates that the `null_row` flag was not set at the beginning. (Bug #29435133)
- Some `ALTER TABLE` statements using `ADD COLUMN` did not perform as expected. (Bug #29428288)

  References: This issue is a regression of: Bug #28333657.

- An incorrect error message was reported during upgrade from MySQL 5.7 to MySQL 8.0 when a non-natively partitioned table was encountered. The error message referenced the `-DWITH_PARTITION_STORAGE_ENGINE` `CMake` build option, which is not supported in MySQL 8.0. (Bug #29426632, Bug #94518)
- At startup, the server did not properly check the length of user and host names in persisted variables. (Bug #29420141)
- The parser could leak memory for certain multiple-statement queries. (Bug #29419820)
- Installing MySQL on EL8 from RPM packages caused a conflict with the installed MeCab library. (Bug #29413115)
- On FreeBSD, stack traces dumped for fatal signals did not work correctly. (Bug #29408039)
- With the `derived_merge` switch disabled in the value of the `optimizer_switch` system variable, information retrieved for base tables from `INFORMATION_SCHEMA.TABLES` displayed inappropriate information in the `TABLE_COMMENT` column. (Bug #29406053, Bug #94468)
- In certain cases, use of `ORDER BY` in a subselect did not produce the expected result. (Bug #29402943)
- `ROLLUP` queries with wildcards (and probably also certain other wildcard queries) could cause assertion errors. (Bug #29396628)
- A condition inside an `IF` function having a constant evaluating to `TRUE` was not always handled correctly. (Bug #29394833)
- A code change in MySQL 8.0.13 related to full-text search caused a segmentation fault. (Bug #29393105)
- `mysqlpump` did not check whether the argument to `--result-file` was opened correctly and exited unexpectedly if an error occurred. (Bug #29389828)
- In queries without joins, batch mode was not enabled until after initialization was complete, so that `OFFSET` rows were read outside of batch mode. This negated any performance benefit to be had from batch mode. (Bug #29373972)
- When renaming a functional index, the server did not check for a possible duplicate index name. (Bug #29360763)
- The system variable service did not check the input buffer size for some operations. (Bug #29343505)
- For `mysql` system schema dumps, `mysqlpump` dumped certain tables by row rather than as account-management statements. (Bug #29343073)
- Creation of a functional index on a `UNION` subquery led to an assertion in debug builds. (Bug #29342245)
- Non-numeric arguments to `AVG()` were not always handled correctly. (Bug #29321764)
- Result sets with a very large number of columns could cause client programs to use all available memory. Now the client library allocates no more than `max_allowed_packet` bytes for result set metadata, returning an out-of-memory error if this limit is exceeded. If this error occurs, it is fatal and the client should disconnect. (Bug #29316814)
- A call to `mysql_shutdown()` could cause the client to exit when the server and client were from different MySQL series, due to return packet payload misinterpretation. (Bug #29315393)
- Client programs could fail while reading result set metadata if communicating with the server using the pre-MySQL 4.1 protocol. (Bug #29304864)

- `HANDLER` statements did not always work correctly with tables having generated columns. (Bug #29300049)
- Session-tracking information in the client/server protocol could be mishandled. (Bug #29297652)
- Client programs that used the `libmysqlclient` C client library could exit upon receipt of an `OK` packet containing malformed session-tracking information. (Bug #29297620, Bug #29630735)
- With the `PAD_CHAR_TO_FULL_LENGTH` SQL mode enabled, password changes failed, with no warning or error reported. (Bug #29287785)
- Initialization code of loadable functions for the `keyring_aws` plugin did not properly check whether the plugin had been initialized. (Bug #29278153)
- Scalar subqueries were not always identified and handled correctly when compared with aggregate or window functions. (Bug #29276063)
- Ubuntu packages did not disable binary logging for initialization scripts. (Bug #29263771)
- The internal collation map could become corrupted by use of invalid collation names. (Bug #29258979)
- Single-table `UPDATE` and `DELETE` statements that used indexed expressions could cause the server to exit when executed as prepared statements. (Bug #29257254)
- The `audit_log` plugin did not log `UNINSTALL PLUGIN audit_log` statements. (Bug #29248047)
- MySQL now uses `open(O_TMPFILE)` whenever applicable when creating a temporary file that is immediately unlinked. This is more efficient than previously and avoids the small possibility of a race condition. Thanks to Daniel Black for the contribution. (Bug #29215177, Bug #93937)
- `audit_log` filtering operations could leak memory. (Bug #29201747)
- The `GRANT OPTION` privilege could be mishandled in some cases. (Bug #29179334)
- Previously, `LIKE` comparisons could be incorrect if either of the `_` or `%` SQL wildcard characters was used as the `ESCAPE` character. These characters can now be used. (Bug #29175461, Bug #93811)
- The `component_sys_variable` service could fail to read certain system variables, causing a server exit. (Bug #29162657)
- Multiple-table `UPDATE` statements could search for incorrect generated-column values when determining which row to update. (Bug #29154379)
- If the server was started with Performance Schema parameters that caused an out-of-memory condition, startup failed. (Bug #29140212, Bug #93726)
- For consistency and compliance with standard SQL, the `ISNULL()` function is now printed as `IS NULL` in warnings generated by `EXPLAIN` statements. (Bug #29119455)
- Some supplemental Unicode characters could incorrectly be flagged with a warning message as invalid. (Bug #29110613, Bug #93626)
- For upgrades from MySQL 5.7 to 8.0, the `root` user was not given all dynamic privileges. (Bug #29043233)
- When using subpartitioning, table serialized dictionary information (SDI) was not stored in any tablespace file. (Bug #29020745, Bug #93499)
- Privileges for dropping some Performance Schema tables were checked incorrectly. (Bug #29010031)

- `mysqldump` failed to wrap `SET NAMES utf8mb4` and `SET character_set_client = utf8mb4` statements within version-specific comments, which could cause compatibility problems. (Bug #29007506, Bug #93450)

- For a unique index created on one or more columns defined as `NULL`, the optimizer failed to handle joins using that index as `eq_ref`.

  As part of this fix, late `NULL` filtering is now also performed to avoid treating comparisons with `NULL` as potential equality matches. (Bug #28965762, Bug #29337233)

- For debug builds, an assertion could be raised by string comparisons for which the expected and actual collation differed. (Bug #28960901)

- The server did not handle correctly a subquery which, prior to conversion to a semijoin, contained a merged derived table or common table expression which in turn had a `WHERE` clause with an outer reference against an indexed column. (Bug #28955216)

- A query that employed a derived table which included an `ORDER BY` was not always handled correctly. (Bug #28942965)

- When clients were terminated for inactivity exceeding the `wait_timeout` value, the message written to the error log was unclear. Now `ER_NET_WAIT_ERROR` is written, which is more specific about the cause of the problem. Thanks to Mattias Jonsson for the contribution. (Bug #28940167, Bug #93240)

- Concurrent execution of `FLUSH PRIVILEGES` along with `CREATE USER` or `ALTER USER` could cause the server to stall. (Bug #28937018, Bug #93085)

- `CASE` statement comparisons that relied on index prefix values could produce incorrect results. (Bug #28934315, Bug #93215)

- When adjusting the query cost after sort elimination to compensate for sorts that had not been considered in the original cost estimate, and the estimates could sometimes be too low, or even negative.

  The sort cost is added to the total cost in `Optimize_table_order::consider_plan()`, and subtracted from the total cost in `JOIN::optimize()` if it is found that sorting is not needed.

  If `consider_plan()` finds that sorting is not needed, it does not add the sort cost to the total cost, but still records a sort cost in `JOIN::sort_cost()`, and this is set to the sort cost of the candidate plan considered previously. Later, `JOIN::optimize()` saw that the `JOIN` object had an associated sort cost, and subtracted that cost from the total cost, since it also sees that no sort is needed. Since the sort cost came from a candidate plan that was not the same as the chosen plan, the estimate was incorrect.

  The fix is to make `consider_plan()` set `JOIN::sort_cost` to 0 if no sort cost is added to the total cost, so that the mistaken adjustment of the cost in `JOIN::optimize()` is not performed. (Bug #28884359)

- Some GIS code failed to compile under Visual Studio 2017 15.5.6. (Bug #28861188)

  References: This issue is a regression of: Bug #28842878.

- Stored generated column values and indexes on virtual generated columns were not correctly updated after columns on which these generated columns depended were swapped using `ALTER TABLE` with `RENAME COLUMN` or `CHANGE COLUMN`.

  Renaming of base columns for generated columns, generated defaults, and functional indexes is now prohibited unless the same `ALTER TABLE` statement satisfies one of these conditions:

  - The statement removes the generated column, generated default, or functional index.
  - The statement updates the dependent expression in question. This enables supporting existing scenarios in which generation expressions are updated to follow base column renaming.

  Restrictions on dropping columns on which generated columns, generated defaults, or functional indexes depend were relaxed in a similar way. (Bug #28772251, Bug #92727)

- Base columns were not excluded from index-only access by a generated column. (Bug #28652733)

  References: See also: Bug #29664369. This issue is a regression of: Bug #23169112.

- `SET PERSIST_ONLY` did not properly handle the `version_tokens_session` system variable. (Bug #28542569)
- For debug builds, improper error checking for `CREATE TABLE` statements could cause an assertion to be raised. (Bug #28490361, Bug #91976)
- MySQL 5.7 supported foreign keys on `InnoDB` tables with a parent key for which part of the referenced columns were hidden, but MySQL 8.0 did not. MySQL 8.0 now supports this capability. (Bug #28480149, Bug #91952)
- `INSERT ... ON DUPLICATE KEY UPDATE` did not consider privileges granted by active roles. (Bug #28395115)
- For unloaded components, component options specified at startup with a `--loose-` prefix were not processed if the component was later loaded at runtime. (Bug #28341329)
- Long passwords solicited interactively by client programs could fail to be null-terminated. (Bug #28121400)
- When building MySQL, `CMake` begins the process of downloading `Boost` by creating a zero-length tarball in the destination directory, which is removed when the download is complete. If the download was interrupted or timed out, the presence of this file prevented `CMake` from attempting to perform the download the time it was run. Now the zero-length tarball, if present, is removed before the download is started. (Bug #28089173)
- A thread pool group could be blocked when a thread process tick time exceeded the maximum permitted value. The tick time now uses a larger data type to permit larger values. (Bug #28072609)
- Privileges were not checked correctly for `ALTER USER ... IDENTIFIED WITH ... BY`. (Bug #27923149, Bug #29882299)
- MySQL does not support OpenSSL session tickets, but did not set the `SSL_OP_NO_TICKET` flag to inform OpenSSL of that. The flag is now set. (Bug #27655493)
- The `audit_null` plugin did not properly check for a null event record. (Bug #27638290)
- `UpdateXML()` did not always free memory properly in certain cases. (Bug #27312862)
- Empty values in the `name` column of the `mysql.plugin` system table caused the server to exit during startup. (Bug #27302459)
- The server did not properly check privileges for `CACHE INDEX` statements. (Bug #26173827)
- With the `thread_pool` plugin enabled, the Performance Schema `status_by_thread` table contained no data. (Bug #25933891)

- The `GRANT OPTION` privilege was treated as related to database operations. (Bug #25203933, Bug #34159579)

- `REPAIR TABLE` for `ARCHIVE` tables could result in a server exit. (Bug #23304911)

- If an `INSTALL PLUGIN` statement contained invalid UTF-8 characters in the shared library name, it caused the server to hang (or to raise an assertion in debug builds). (Bug #14653594, Bug #23080148, Bug #27167197)

- Logging to the `mysql.slow_log` system table could fail when values were to large for table columns. Now logging proceeds on a best-effort basis, writing what information can be provided. Otherwise, the row is discarded and a message is written to the error log. (Bug #11748692, Bug #37132)

- A query using `WHERE date_column LIKE 'year_value'` failed with error 1525 `Incorrect DATE value` on Windows platforms. (Bug #95780, Bug #29904751)

  References: This issue is a regression of: Bug #29368521.

- A fix for a previous issue caused `YEAR` values to be treated as unsigned, but this did not allow for possible negative values arising during calculations involving subtraction, which could lead to data truncation errors. (Bug #95045, Bug #29668676)

  References: This issue is a regression of: Bug #92209, Bug #28562930.

- When working with derived tables with an aggregation which had zero input rows, the results of the aggregate functions were not properly copied into the temporary table. This caused incorrect results in cases where the derived table was evaluated multiple times, such as when performing a lateral join. (Bug #94721, Bug #29514504)

- A window without a frame specification inheriting from a window with an `ORDER BY` yielded an incorrect result. (Bug #94251, Bug #29328529)

- The results returned by the functions `REGEXP_REPLACE()` and `REGEXP_SUBSTR()` used `UTF-16` rather than the character set and collation of the expression searched for matches. (Bug #94203, Bug #29308212)

- A `UNION ALL` query with `SUM(constant)` was processed very slowly compared to the same query using `SUM(column)` instead. (Bug #93922, Bug #29227464)

- `JSON_OBJECTAGG()` is non-deterministic in the presence of duplicate keys unless the window has ordering on the key, which is expected behavior, but a key-value pair that was no longer in the window frame still appeared in the result. (Bug #93822, Bug #29175262)

- `LIMIT` was applied before `HAVING` in queries with subselects. (Bug #93214, Bug #28934388)

  References: This issue is a regression of: Bug #25466100.

- A query involving `GROUP BY` on a `TIMESTAMP` column resulted in a duplicate entry for key (`ER_DUP_ENTRY`) error. This problem arose when `TIMESTAMP` values were inserted into a table using a given setting for the time zone and these values were later fetched after the time zone setting had been changed, such that at least some of the inserted `TIMESTAMP` values occurred during the hour that the time changed from standard to daylight time (DST) in the new time zone, during which time the same `TIMESTAMP` value can exist twice. Now, when the server would otherwise return the error `DUPLICATE ENTRY FOR KEY 'group_key'`, if the grouping involves a `TIMESTAMP` column, it instead raises

the error `Grouping on temporal is non-deterministic for time zones having DST. Please consider switching to UTC for this query.`

In addition, it is suggested to set `explicit_defaults_for_timestamp` to `ON` as well as one or more of `MODE_NO_ZERO_IN_DATE`, `MODE_NO_ZERO_DATE`, or `MODE_INVALID_DATES` as part of the server SQL mode to help avoid this issue. (Bug #90398, Bug #27970159)

# Changes in MySQL 8.0.16 (2019-04-25)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.


- Account Management Notes
- C API Notes
- Character Set Support
- Compilation Notes
- Configuration Notes
- Deprecation and Removal Notes
- Installation Notes
- Packaging Notes
- Parser Notes
- Performance Schema Notes
- Plugin Notes
- Security Notes
- Spatial Data Support
- SQL Syntax Notes
- sys Schema Notes
- Test Suite Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Account Management Notes

- Previously, users who had the `DROP ROLE` privilege could use the `DROP ROLE` statement to drop locked or unlocked accounts. Now, users who have the `DROP ROLE` privilege can use `DROP ROLE` only to drop accounts that are locked (unlocked accounts are presumably user accounts used to log in to the server and not just as roles). Users who have the `CREATE USER` privilege can use `DROP ROLE` to drop accounts that are locked or unlocked. (Bug #28953158, Bug #93263)

- Several changes have been made to MySQL account-management capabilities:

- MySQL now incorporates the concept of user account categories, with system and regular users distinguished according to whether they have the new `SYSTEM_USER` privilege:

  - System users are users who possess the `SYSTEM_USER` privilege. A system user can perform operations on both system and regular accounts.

  - Regular users are ordinary users who do not possess the `SYSTEM_USER` privilege. A regular user can perform operations on regular accounts, but not system accounts.

  If a user has the appropriate privileges to perform a given operation on regular accounts, `SYSTEM_USER` enables the user to also perform the operation on system accounts. `SYSTEM_USER` does not imply any other privilege, so the ability to perform a given account operation remains predicated on possession of any other required privileges. For example, if a user can grant the `SELECT` and `UPDATE` privileges to regular accounts, then with `SYSTEM_USER` the user can also grant `SELECT` and `UPDATE` to system accounts.

  The distinction between system and regular accounts enables better control over certain account administration issues by protecting accounts that have the `SYSTEM_USER` privilege from accounts that do not have the privilege. For example, the `CREATE USER` privilege enables not only creation of new accounts, but modification and removal of existing accounts. Without the system user concept, a user who has the `CREATE USER` privilege can modify or drop any existing account, including the `root` account. The concept of system user enables restricting modifications to the `root` account (itself a system account) so they can be made only by system users. Regular users with the `CREATE USER` privilege can still modify or drop existing accounts, but only regular accounts.

  Other operational implications of the `SYSTEM_USER` privilege:

  - A session that has the `SYSTEM_USER` privilege can be killed only by users who have the `SYSTEM_USER` privilege, in addition to any other required privileges.

  - An account that has the `SYSTEM_USER` privilege can be specified as the `DEFINER` for a stored object only by users who have the `SYSTEM_USER` privilege, in addition to any other required privileges.

  - A role that has the `SYSTEM_USER` privilege cannot be listed in the value of the `mandatory_roles` system variable.

  For more information, see Account Categories.

- Previously, it was not possible to grant privileges that apply globally except for certain schemas. This is now possible if the new `partial_revokes` system variable is enabled. For example, the following statements enable an account to select from or insert into any table except those in the `mysql` system schema:

```
SET PERSIST partial_revokes = ON;
GRANT SELECT, INSERT ON *.* TO u1;
REVOKE SELECT, INSERT ON mysql.* FROM u1;
```

  The server records partial revokes by adding a `Restrictions` attribute to the `User_attributes` column of the `mysql.user` system table. `SHOW GRANTS` includes `REVOKE` statements in its output to indicate partial revokes.

  **Note**

  Enabling `partial_revokes` causes MySQL to treat unescaped `_` and `%` SQL wildcard characters in schema names as literal characters, just as if they

> had been escaped as `\_` and `\%`. Because this changes the interpretation of privilege assignments, it may be advisable to avoid unescaped wildcard characters in schema privilege assignments for MySQL installations where `partial_revokes` may be enabled.

For more information, see Privilege Restriction Using Partial Revokes.

- The `GRANT` statement has a new `AS user [WITH ROLE]` clause that specifies additional information about the privilege context to use for statement execution. This syntax is visible at the SQL level, although its primary purpose is to enable uniform replication across all nodes of grantor privilege restrictions imposed by partial revokes, by causing those restrictions to appear in the binary log.

  For more information, see GRANT Statement.

(WL #12098, WL #12364, WL #12820)

## C API Notes

- The MySQL C API now supports asynchronous functions for nonblocking communication with the MySQL server:
  - `mysql_real_connect_nonblocking()`
  - `mysql_real_query_nonblocking()`
  - `mysql_store_result_nonblocking()`
  - `mysql_next_result_nonblocking()`
  - `mysql_fetch_row_nonblocking()`
  - `mysql_free_result_nonblocking()`

  Each function is the asynchronous counterpart to a synchronous function that has the same name without a `_nonblocking` suffix. The synchronous functions block if reads from or writes to the server connection must wait. The asynchronous functions enable an application to check whether work on the server connection is ready to proceed. If not, the application can perform other work before checking again later. See C API Asynchronous Interface. (WL #11381)

## Character Set Support

- MySQL now supports a new Chinese collation, `utf8mb4_zh_0900_as_cs`, for the `utf8mb4` Unicode character set. `utf8mb4_zh_0900_as_cs` is the first Chinese language-specific collation available for Unicode in MySQL. This collation is accent sensitive and case sensitive. Its characteristics are similar to `utf8mb4_0900_as_cs` except that language-specific rules take precedence where applicable. For more information, see Unicode Character Sets. (WL #11825)

## Compilation Notes

- `CMake` now causes the build process to link with the `llvm lld` linker for Clang if it is available and not explicitly disabled. To disable use of this linker, specify the `-DUSE_LD_LLD=OFF` option. (Bug #29264211)
- Builds on EL6 and EL7 now try to use the compiler in `devtoolset-8` rather than `devtoolset-7`. (Bug #29198846)
- The minimum version of the Boost library for server builds is now 1.69.0. (Bug #29114233)

- The configuration-time check for Visual Studio 2017 was not specific enough. The check for MySQL compilation now requires at least Visual Studio update 15.8, which is version number 1915. (Bug #28970895)
- MySQL now can be compiled using C++14. The following minimum version requirements apply for compiler support:
  - GCC 5.3 (Linux)
  - Clang 4.0 (FreeBSD)
  - XCode 9 (macOS)
  - Developer Studio 12.6 (Solaris)
  - Visual Studio 2017 (Windows)

  (WL #12424)

## Configuration Notes

- MySQL configuration now requires a minimum `CMake` version of 3.4.3. This requires the use of `cmake3` rather than `cmake` on some Red Hat and Oracle Linux platforms. (Bug #29246216)
- The `WITH_LZMA` `CMake` option was removed. (Bug #29153932, Bug #93755)
- The `EXCLUDE_FROM_ALL` option is now used in `CMake` configuration as appropriate so that libraries are built only if they are actually used by any executable. (Bug #29052599)
- The new `WITH_JEMALLOC` `CMake` option indicates whether to link with `-ljemalloc`. If enabled, built-in `malloc()`, `calloc()`, `realloc()`, and `free()` routines are disabled. The default is `OFF`. (Bug #29027974)
- The new `WITH_LSAN` `CMake` option indicates whether to run LeakSanitizer, without AddressSanitizer. The default is `OFF`. (Bug #28936574)
- The new `WITH_ROUTER` `CMake` option indicates whether to build MySQL Router. The default is `ON`. (Bug #28759234)
- MySQL Server now supports a `--validate-config` option that enables the startup configuration to be checked for problems without running the server in normal operational mode. For more information, see Server Configuration Validation. (WL #12360)

## Deprecation and Removal Notes

- The `TempTable` storage engine now always uses `InnoDB` to manage internal temporary tables on disk, and the choice of storage engine employed for this task is no longer user-configurable. The `internal_tmp_disk_storage_engine` system variable has been removed. (Bug #91377, Bug #28234637, WL #11974)

  References: See also: Bug #28081038, Bug #82556, Bug #27408352.

## Installation Notes

- Previously, after installation of a new version of MySQL, the MySQL server automatically upgraded the data dictionary tables at the next startup, after which the DBA was expected to invoke `mysql_upgrade`

manually to upgrade the system tables in the `mysql` schema, as well as objects in other schemas such as the `sys` schema and user schemas.

The server now performs the tasks previously handled by `mysql_upgrade`. After installation of a new MySQL version, the server now automatically performs all necessary upgrade tasks at the next startup and is not dependent on the DBA invoking `mysql_upgrade`. In addition, the server updates the contents of the help tables (something `mysql_upgrade` did not do). A new `--upgrade` server option provides control over how the server performs automatic data dictionary and server upgrade operations. For more information, see Upgrading MySQL.

This change to the upgrade procedure results in some deprecations:

- `mysql_upgrade` is deprecated because it is no longer necessary.
- The `--no-dd-upgrade` server option is deprecated because the `--upgrade` option supersedes it.

`mysql_upgrade` and the `--no-dd-upgrade` option will be removed in a future MySQL version. (Bug #28146052, Bug #28162609, Bug #91205, Bug #29185739, Bug #27740692, Bug #28547424, Bug #91961, WL #12413, WL #12918)

## Packaging Notes

- The Docker image for MySQL Cluster 8.0 is now available for download. (Bug #30010921, Bug #96084)
- Binary packages that include `curl` rather than linking to the system `curl` library now use `curl` 7.64.0. (Bug #29357198)
- The Henry Spencer regex library (`extra/regex`) is no longer used by MySQL 8.0 and is no longer present in source distributions. (Bug #29192306)
- RPM packages now have dependencies on `libtirpc` and `rpcgen` because newer versions of `glibc` do not include Sun RPC. (Bug #28995257)
- The `support-files/magic` file was removed from the MySQL source tree. Most MySQL file formats are covered by operating system file type capabilities. (Bug #18335080, Bug #71898)
- MySQL now provides “Minimal Install” Linux generic binary download packages for MySQL Server and the Test Suite. Minimal install packages exclude debug binaries and are stripped of debug symbols, making them significantly smaller than the regular Linux generic binary packages. Downloads are available at https://dev.mysql.com/downloads/.
- The Docker image for MySQL Enterprise Edition is now available for download from My Oracle Support.

## Parser Notes

- The parser no longer accepts the undocumented and nonstandard `= alias_name` syntax for specifying table aliases. (Bug #29205289)
- The parser is now permits several additional nonreserved keywords to be used as labels within stored programs that previously were restricted from such use: `ACCOUNT`, `ALWAYS`, `BACKUP`, `CLOSE`, `FORMAT`, `GROUP_REPLICATION`, `HOST`, `INVISIBLE`, `OPEN`, `OPTIONS`, `OWNER`, `PARSER`, `PORT`, `REMOVE`, `RESTORE`, `ROLE`, `SECONDARY`, `SECONDARY_ENGINE`, `SECONDARY_LOAD`, `SECONDARY_UNLOAD`, `SECURITY`, `SERVER`, `SOCKET`, `SONAME`, `UPGRADE`, `VISIBLE`, `WRAPPER`. (Bug #29033659)
- The parser accepted ODBC escape syntax for outer joins (`{ OJ outer_join }`), but also accepted identifiers other than `OJ`. The parser now accepts only `OJ`.

**Note**

`OJ` now is a nonreserved keyword.

(Bug #22320942)

## Performance Schema Notes

- The new Performance Schema `keyring_keys` table exposes metadata for keys in the MySQL Keyring. Key metadata includes key IDs, key owners, and backend key IDs. The `keyring_keys` table does not expose any sensitive keyring data such as key contents. See The keyring_keys table. (WL #11543)

## Plugin Notes

- MySQL now includes a `ddl_rewriter` plugin that modifies `CREATE TABLE` statements received by the server before it parses and executes them. The plugin removes `ENCRYPTION`, `DATA DIRECTORY`, and `INDEX DIRECTORY` clauses, which may be helpful when restoring tables from SQL dump files created from databases that are encrypted or that have their tables stored outside the data directory. For example, the plugin may enable restoring such dump files into an unencrypted instance or in an environment where the paths outside the data directory are not accessible. When installed, `ddl_rewriter` exposes the Performance Schema `memory/rewriter/ddl_rewriter` instrument for tracking plugin memory use. For more information, see The ddl_rewriter Plugin. (WL #12668)

## Security Notes

- Previously, if the grant tables were corrupted, the MySQL server wrote a message to the error log but continued as if the `--skip-grant-tables` option had been specified. This resulted in the server operating in an unexpected state unless `--skip-grant-tables` had in fact been specified. Now, the server stops after writing a message to the error log unless started with `--skip-grant-tables`. (Starting the server with that option enables you to connect to perform diagnostic operations.) (Bug #29394501, Bug #94394, WL #12971)

- The OpenSSL libraries bundled with MySQL on some platforms (Windows, macOS, and Generic Linux) have been upgraded to version 1.0.2r. On all other platforms, MySQL uses the system installed OpenSSL. Issues fixed in the new OpenSSL version are described at http://www.openssl.org/news/vulnerabilities.html. (Bug #28988091)

- Granting roles to anonymous users is no longer supported because such roles caused problematic behavior. (Bug #28910120)

- OpenSSL 1.1.1 supports the TLS v1.3 protocol for encrypted connections, and MySQL now supports TLS v1.3 as well, if both the server and client are compiled using OpenSSL 1.1.1 or higher:

  - Some TLSv1.3 ciphersuites are enabled by default. The `tls_ciphersuites` system variable enables explicitly specifying which TLSv1.3 ciphersuites the server permits.

  - The `--tls-ciphersuites` client option specifies which TLSv1.3 ciphersuites a client permits. This option applies to these programs: `mysql`, `mysqladmin`, `mysqlbinlog`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlpump`, `mysqlshow`, `mysqlslap`, `mysqltest`, `mysql_secure_installation`, and `mysql_upgrade`.

  - The `mysql_options()` C API function has a new `MYSQL_OPT_TLS_CIPHERSUITES` option that specifies from within the client library which TLSv1.3 ciphersuites a client program permits.

  For more information, see Encrypted Connection TLS Protocols and Ciphers.

**Note**

Group Replication does not support TLSv1.3.

(WL #12361)

- The TLS context the server uses for new connections now is reconfigurable at runtime. This capability may be useful, for example, to avoid restarting a MySQL server that has been running so long that its SSL certificate has expired. Dynamic SSL reconfigurability is based on these changes:

  - The system variables that define the TLS context are dynamic and can be changed at runtime: `ssl_ca`, `ssl_capath`, `ssl_cert`, `ssl_cipher`, `ssl_crl`, `ssl_crlpath`, `ssl_key`, `tls_ciphersuites`. `tls_version`.

  - The `ALTER INSTANCE` statement supports a `RELOAD TLS` action that reconfigures the TLS context from the current values of the system variables that define the context.

  - These status variables reflect the TLS context the server uses for new connections: `Current_tls_ca`, `Current_tls_capath`, `Current_tls_cert`, `Current_tls_cipher`, `Current_tls_ciphersuites`, `Current_tls_crl`, `Current_tls_crlpath`, `Current_tls_key`, `Current_tls_version`.

    `ALTER INSTANCE RELOAD TLS` updates those status variables from the corresponding system variable values when it reconfigures the TLS context.

  For more information, see Server-Side Runtime Configuration and Monitoring for Encrypted Connections.

  Thanks to Facebook for submitting code for a similar feature (although the code was not used). (WL #11541)

  References: See also: Bug #27980097.

- In MySQL 8.0, the default authentication plugin was changed from `mysql_native_password` to `caching_sha2_password`. Because `caching_sha2_password` provides a superset of the capabilities of the `sha256_password` authentication plugin, `sha256_password` is now deprecated and will be removed in a future MySQL version. MySQL accounts that authenticate using `sha256_password` should be migrated to use `caching_sha2_password` instead. (WL #12694)

## Spatial Data Support

- The `ST_Length()` function now takes an optional second argument enabling the unit for the return value to be specified. Permitted units are those listed in the new `INFORMATION_SCHEMA ST_UNITS_OF_MEASURE` table. See LineString and MultiLineString Property Functions, and The INFORMATION_SCHEMA ST_UNITS_OF_MEASURE Table. (WL #12657)

## SQL Syntax Notes

- **Incompatible Change:** In MySQL 5.7, specifying a `FOREIGN KEY` definition for an `InnoDB` table without a `CONSTRAINT` *`symbol`* clause, or specifying the `CONSTRAINT` keyword without a `symbol`, causes `InnoDB` to use a generated constraint name. That behavior changed in MySQL 8.0, with `InnoDB` using the `FOREIGN KEY` *`index_name`* value instead of a generated name. Because constraint names must be unique per schema (database), the change caused errors due to foreign key index names that were not unique per schema. To avoid such errors, the new constraint naming behavior has been reverted, and `InnoDB` once again uses a generated constraint name.

For consistency with InnoDB, the NDB storage engine now uses a generated constraint name if the CONSTRAINT *symbol* clause is not specified, or the CONSTRAINT keyword is specified without a symbol. In NDB releases based on MySQL 5.7 and earlier MySQL 8.0 releases, NDB used the FOREIGN KEY *index_name* value.

The changes described above may introduce incompatibilities for applications that depend on the previous foreign key constraint naming behavior. (Bug #29173134)

- Previously, MySQL permitted a limited form of CHECK constraint syntax, but parsed and ignored it. MySQL now implements the core features of table and column CHECK constraints, for all storage engines. Constraints are defined using CREATE TABLE and ALTER TABLE statements. The new INFORMATION_SCHEMA.CHECK_CONSTRAINTS table provides information about CHECK constraints defined on tables. For more information, see CHECK Constraints. (Bug #11744849, Bug #3464, Bug #3465, Bug #11746042, Bug #22759, WL #929)

## sys Schema Notes

- MySQL now includes built-in SQL functions that format or retrieve Performance Schema data, and that may be used as equivalents for existing sys schema stored functions:
  - FORMAT_BYTES(): Converts a byte count to a value with units. Similar to sys.format_bytes().
  - FORMAT_PICO_TIME(): Converts a time in picoseconds to a value with units. Similar to sys.format_time().
  - PS_THREAD_ID(): Returns the Performance Schema thread ID for a given thread. Similar to sys.ps_thread_id() invoked with a non-NULL argument.
  - PS_CURRENT_THREAD_ID(): Returns the Performance Schema thread ID for the current thread. Shortcut for sys.ps_thread_id() invoked with a NULL argument.

  The built-in functions can be invoked in any schema and require no qualifier, unlike the sys functions, which require either a sys. schema qualifier or that sys be the current schema.

  The built-in functions supersede the corresponding sys functions, which now are deprecated and will be removed in a future MySQL version. Applications that use the sys functions should be adjusted to use the built-in functions instead, keeping in mind some minor differences between the sys functions and the built-in functions. See Performance Schema Functions. (WL #7803)

## Test Suite Notes

- mysql-test-run.pl now supports the MTR_UNIQUE_IDS_DIR environment variable, which may be set to specify a unique-IDs directory to be used as the common location for all chroot environments by multiple simultaneous mysql-test-run.pl instances. This enables those instances to avoid conflicts when reserving port numbers. Thanks to Facebook for the contribution. (Bug #29221085, Bug #93950)
- The my_safe_process program was renamed to mysqltest_safe_process and now is installed with other binaries such as mysqltest rather than with test suite files. (Bug #29198969)
- These changes to the all_persisted_variables test were implemented:
  - It limits the number of hardcoded values in the test output by putting them into local variables. New patches that are then rebased on top that add new system variables do not need to change as many lines of the original test case, making it easier on the rebase process.

- It removes entries for bugs that were fixed and modifies the queries to include the system variables that were not tested due to open bugs.

Thanks to the Facebook team for the contribution. (Bug #29013375, Bug #93478)

## X Plugin Notes

- X Plugin previously returned a `StmtExecuteOk` message to the client after query cleanup had ended and the session had been deactivated. The message is now returned as soon as the result is known and before query cleanup, which gives a noticeable improvement in performance. (Bug #28997370)
- X Plugin logged the system message "`X Plugin ready for connections`" when user connections were not available because preparation of I/O interfaces had failed. (Bug #28906360)
- Some items in the X Plugin code were not instrumented for the Performance Schema by default. (Bug #28898155)
- X Protocol now supports the `COM_RESET_CONNECTION` utility command to reset the session state without re-authenticating or closing the connection. (Bug #28732455)
- X Plugin produced compilation warnings when the MySQL Server source code was built using the Clang 8 compiler. (Bug #28732158)

## Functionality Added or Changed

- **InnoDB:** When the amount of memory occupied by the TempTable storage engine exceeds the limit defined by the `temptable_max_ram` variable, the TempTable storage engine allocates space for internal in-memory temporary tables as memory-mapped temporary files. This behavior is now controlled by the `temptable_use_mmap` variable, which can be disabled to have the TempTable storage engine use `InnoDB` on-disk internal temporary tables instead. For more information, see Internal Temporary Table Use in MySQL. (Bug #28944457)
- **InnoDB:** `undo` and `purge` subsystem counters were added for monitoring background activities associated with undo log truncation. For counter names and descriptions, query the `INFORMATION_SCHEMA.INNODB_METRICS` table.

  ```
  SELECT NAME, SUBSYSTEM, COMMENT FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME LIKE '%truncate%';
  ```

  For information about enabling counters and querying counter data, see InnoDB INFORMATION_SCHEMA Metrics Table. (Bug #28813526, WL #12600)
- **InnoDB:** The new `innodb_spin_wait_pause_multiplier` variable provides greater control over the duration of spin-lock polling delays that occur when a thread waits to acquire a mutex or rw-lock. Delays can now be tuned more finely to account for differences in PAUSE instruction duration on different processor architectures. For more information, see Configuring Spin Lock Polling. (WL #12616)
- **InnoDB:** An internal service interface was added to support tracking of modified pages. (WL #11082)
- **InnoDB:** The `InnoDB` data-at-rest encryption feature now supports encryption of the `mysql` system tablespace. The `mysql` system tablespace contains the `mysql` system database and the MySQL data dictionary tables. (WL #12063)
- **Group Replication:** The group communication engine (XCom) includes a cache for messages (and their metadata) that are exchanged between the group members as a part of the consensus protocol. Among other functions, the message cache is used for recovery by members that return to the group after a period where they were unable to communicate with the other group members.

Previously, the size limit for the message cache was fixed at 1GB of memory, and the maximum number of messages in the cache was also fixed. Now that the `group_replication_member_expel_timeout` system variable (introduced in MySQL 8.0.13) can be set to allow up to an hour for members to return to the group rather than being expelled, having a fixed 1GB limit on the size of the cache could cause such nodes to be unable to recover the messages they missed on re-establishing communication.

For this reason, beginning with this release, the XCom message cache has no fixed limit on the number of messages it can contain, and is bounded only by the limit set for the amount of memory it can use. The cache size limit can be set using the new `group_replication_message_cache_size` system variable, which has a default and minimum setting of 1GB, as used in previous MySQL Server versions. If the cache size limit is reached, XCom removes the oldest entries that have been decided and delivered. The cache size limit can be increased or reduced dynamically at runtime. If you reduce the cache size limit, XCom removes the oldest entries that have been decided and delivered until the current size is below the limit. The Group Replication Group Communication System (GCS) issues a warning when a message likely to be needed for recovery by a currently unreachable member is removed from the message cache. (Bug #26482507, WL #11615)

- **Group Replication:** Large messages sent between Group Replication group members can now be split into multiple messages when they exceed a user-defined threshold size. Sending an abnormally large message can result in some group members being reported as failed and expelled from the group. This is because the single thread used by the group communication engine (XCom) is occupied processing the message for too long, so some of the group members might report the receiver as failed.

  A new system variable `group_replication_communication_max_message_size` specifies the maximum message size for Group Replication communications. Messages greater than this size are automatically split into fragments that are sent separately and reassembled by the recipients. Message delivery for a fragmented message is considered complete when all the fragments of the message have been received and reassembled by all the group members. Fragmentation is applied by default, and can be switched off by specifying a zero value for the system variable.

  Because older MySQL Server releases do not support message fragmentation, in order to ensure backward compatibility, Group Replication now has the concept of a communication protocol for the group. The communication protocol version is set to accommodate the oldest MySQL Server version that you want the group to support. A MySQL server at version X can only join and reach `ONLINE` status in a replication group if the group's communication protocol version is less than or equal to X.

  You can inspect the communication protocol in use by a group by using the new `group_replication_get_communication_protocol()` function, which returns the oldest MySQL Server version that the group supports. When a new member joins a replication group, it checks the communication protocol version that is announced by the existing members of the group. If the joining member supports that version, it joins the group and uses the communication protocol that the group has announced, even if the member supports additional communication capabilities. If the joining member does not support the communication protocol version, it is expelled from the group.

  If you need to change the communication protocol version of a group so that members at earlier releases can join, use the new `group_replication_set_communication_protocol()` function to specify the MySQL Server version of the oldest member that you want to allow. This makes the group fall back to a compatible communication protocol version if possible. If you upgrade all the members of a replication group to a new MySQL Server release, the group's communication protocol version is not automatically upgraded to match. If you no longer need to support members from earlier releases, use the `group_replication_set_communication_protocol()` function to set the communication protocol version to the new MySQL Server version to which you have upgraded the members. (Bug

#26438884, Bug #23240361, Bug #28474580, Bug #91830, Bug #28642504, Bug #26941977, Bug #29240931, WL #11610)

- **Group Replication:** A new system variable `group_replication_autorejoin_tries` lets you specify the number of tries that a member makes to automatically rejoin the group if it is expelled, or if it is unable to contact a majority of the group before the `group_replication_unreachable_majority_timeout` setting is reached. `0`, the default, means that the member does not try to rejoin, and proceeds to the action specified by the `group_replication_exit_state_action` system variable.

  Activate auto-rejoin if you can tolerate the possibility of stale reads and want to minimize the need for manual intervention, especially where transient network issues fairly often result in the expulsion of members. If you specify a number of tries, when the member's expulsion or unreachable majority timeout is reached, it makes an attempt to rejoin (using the current plugin option values), then continues to make further auto-rejoin attempts up to the specified number of tries. After an unsuccessful auto-rejoin attempt, the member waits 5 minutes before the next try. During the auto-rejoin procedure, the member remains in super read only mode and displays an `ERROR` state on its view of the replication group. The member can be stopped manually at any time by using a `STOP GROUP_REPLICATION` statement or shutting down the server. If the specified number of tries is exhausted without the member rejoining or being stopped, the member proceeds to the action specified by `group_replication_exit_state_action`. (Bug #25673350, Bug #84784, Bug #28732174, WL #11284)

- **Group Replication:** The default for `group_replication_exit_state_action` has been changed from `ABORT_SERVER` to `READ_ONLY`. (WL #12659)

- Some `InnoDB` memory allocation functions that previously were evaluated at runtime now are evaluated at compile time, resulting in performance improvements. (Bug #29370811, Bug #94380)

- The semijoin optimizations for `IN` subqueries have been extended to work with `EXISTS` subqueries as well; these can now be handled with the same semijoin strategies as `IN` subqueries, including first-match, materialization, duplicate weedout and loose index scan.

  In addition, the optimizer decorrelates trivially-correlated equality predicates in the `WHERE` condition attached to the subquery, so that they can be treated similarly to expressions in `IN` subqueries. The decorrelation is now also performed for `IN` subqueries as well as `EXISTS` subqueries.

  All hints and optimizer switches applicable to `IN` subqueries which are transformed into semijoin operations are also applicable to transformed `EXISTS` subqueries. All limitations on such optimization of `IN` subqueries also apply to transformed `EXISTS` subqueries, so that, for example, aggregate `EXISTS` subqueries cannot be transformed.

  For more information, see Optimizing IN and EXISTS Subquery Predicates with Semijoin Transformations. (Bug #28805105, Bug #28857990, WL #4389)

- For consistency with the SQL standard and other RDBMS, table aliases are now supported in single-table as well as multi-table `DELETE` statements. (Bug #27455809)

- `WHERE` conditions making comparisons between constants and column values in which the constant value is out of range or of the wrong type with respect to the column type are now handled during optimization rather than during execution. For example, given a table `t` with a column `c` whose type is `TINYINT UNSIGNED`, the condition in the query `SELECT * FROM t WHERE c < 256` can be folded to `SELECT * FROM t WHERE TRUE` because 256 is out of range for a column of this type.

Comparisons with `NULL` columns can also be optimized; if the column `c` is nullable, the same query can be optimized as `SELECT * FROM t WHERE c IS NOT NULL`.

The comparisons that can be treated in this manner are `>`, `>=`, `<`, `<=`, `=`, `<>`/`!=`, and `<=>`. (`BETWEEN` and `IN` are not currently supported.) Types for which comparisons can be folded based on range and type include integer, floating-point, and fixed-point numeric types. `BIT` is not supported by this optimization, nor are columns of date and time types.

For more information, see Constant-Folding Optimization. (Bug #90100, Bug #25484743, Bug #29048682, Bug #27703371, WL #11935)

References: See also: Bug #28172538, Bug #29699347.

- Added an experimental tree format for `EXPLAIN` output, which prints the generated iterator tree, and is intended to help users understand how execution was actually set up. `EXPLAIN FORMAT=TREE` is currently unsupported in production and both its syntax and output are subject to change in subsequent versions of MySQL. (WL #12074)

- When binary log and relay log encryption is in use on a MySQL server (`binlog_encryption=ON`), you can now rotate the binary log master key at any time while the server is running by issuing `ALTER INSTANCE ROTATE BINLOG MASTER KEY`. You can do this on a regular basis to comply with your organization's security policy, and also if you suspect that the current or any of the previous binary log master keys might have been compromised.

  When you rotate the binary log master key, the new master key is used to encrypt the file passwords for the new binary log and relay log files, and subsequent files until the key is changed again. The file passwords for existing encrypted binary log files and relay log files on the server are also re-encrypted in turn using the new binary log master key, starting with the most recent files. Any unencrypted files are skipped. Finally, all binary log encryption keys that no longer apply to any retained binary log files or relay log files are cleaned up from the keyring. (WL #12080)

- Table encryption can now be managed globally by defining and enforcing encryption defaults. The `default_table_encryption` variable defines an encryption default for newly created schemas and general tablespaces. The encryption default for a schema can also be defined using the `DEFAULT ENCRYPTION` clause when creating a schema. By default, a table inherits the encryption of the schema or general tablespace it is created in.

  Encryption defaults are enforced by enabling the `table_encryption_privilege_check` variable. The privilege check occurs when creating or altering a schema or general tablespace with an encryption setting that differs from the `default_table_encryption` setting, or when creating or altering a table with an encryption setting that differs from the default schema encryption.

  The `TABLE_ENCRYPTION_ADMIN` privilege permits overriding default encryption settings when `table_encryption_privilege_check` is enabled.

  For more information, see Defining an Encryption Default for Schemas and General Tablespaces. (WL #12261)

- When insertions, deletions, or updates are made to partitioned tables, the binary log now records information about the partition and (if any) the subpartition in which the row event took place. A new row event is created for a modification that takes place in a different partition or subpartition, even if the table involved is the same. So if a transaction involves three partitions or subpartitions, three row events are generated. For an update event, the partition information is recorded for both the “before” image and the “after” image. The partition information is displayed if you specify the `-v` or `--verbose` option when viewing the binary log using `mysqlbinlog`. Partition information is only recorded when row-based logging is in use (`binlog_format=ROW`). (WL #12168)

## Bugs Fixed

- **NDB Cluster:** `EXPLAIN` of a query executed with table access type `eq_ref` could also show a condition being pushed down, even when condition pushdown was not supported for the query. Now the access type is checked before `NDB` checks the condition for a possible pushdown optimization.

  This fix does not affect handling of tables which are part of a pushed join, for which `NDB` continues to support pushed conditions as before. (Bug #27429615)

  References: See also: Bug #27397802, Bug #27808758, Bug #90301.

- **InnoDB:** Undo tablespaces remained unencrypted after enabling undo tablespace encryption at startup. (Bug #29477795)

- **InnoDB:** Problematic macros introduced with undo tablespace DDL support in MySQL 8.0.14 were revised. (Bug #29324132, Bug #94243)

- **InnoDB:** Static thread local variables defined at the wrong scope were not released at thread exit. (Bug #29305186)

- **InnoDB:** The `performance_schema.data_locks` `LOCK_DATA` column only showed the secondary index values of the locked record for a lock placed on a unique secondary index, which was not sufficient to ensure the uniqueness of identified records. The clustered index column values of the locked record are now appended. (Bug #29296645)

- **InnoDB:** An incorrect count of transactions using a rollback segment for recovery of an XA transaction prevented an undo tablespace truncation operation from proceeding and left the purge thread busy checking for the undo tablespace to become empty. (Bug #29273194)

  References: This issue is a regression of: Bug #29273194.

- **InnoDB:** Invalid assertions were raised at startup after a failure to retrieve the space ID of a compressed file-per-table tablespace. The invalid assertion code was removed. (Bug #29221385, Bug #93760)

- **InnoDB:** Optimized `InnoDB` internal temporary tables did not support in-place `UPDATE` operations, which caused the number of delete-marked records to increase continuously. The large number of delete-marked records could cause longer than expected query execution times. (Bug #29207450)

- **InnoDB:** The `std::sort` function in the Contention-Aware Transaction Scheduling (CATS) algorithm was replaced by the `std::stable_sort` function to preserve the original FIFO order for transactions of equal weight. (Bug #29058967)

- **InnoDB:** Write-ahead did not work as expected due to an incorrectly initialized variable.

  Thanks to Yuhui Wang for the contribution. (Bug #29028838, Bug #93442)

- **InnoDB:** The base column information for a generated column was not stored. (Bug #29021730)

- **InnoDB:** An implicit lock check on secondary indexes needlessly compared columns using collation rules. (Bug #29010725)

- **InnoDB:** Assertion code related to the `innodb_flush_method` `O_DIRECT_NO_FSYNC` setting was no longer valid due to a recent modification to that setting. Assertion code was revised. (Bug #29007731)

  References: See also: Bug #27309336.

- **InnoDB:** When starting the server with undo log encryption enabled, the master key for newly created undo tablespaces was generated without a server UUID. Undo tablespaces should use the DefaultMasterKey if the server UUID is not yet generated. (Bug #29006275)

- **InnoDB:** Data dictionary code did not check for a returned data dictionary object, which could potentially cause the server to exit due to a null pointer access. (Bug #28977444, Bug #93362)
- **InnoDB:** An undo tablespace file was left behind by a failed `CREATE UNDO TABLESPACE` operation. (Bug #28966457)
- **InnoDB:** A `CREATE UNDO TABLESPACE` statement failed on Windows due to an invalid character in the file name. The failure resulted in a hang condition due to a missing `OS_FILE_ON_ERROR_NO_EXIT` attribute in the call that creates the undo tablespace file. (Bug #28955676)
- **InnoDB:** Modifying the value of the `innodb_undo_log_encrypt` variable was not a blocking operation, which could lead to the modification being reverted by a background thread after the operation appeared to have been completed successfully. (Bug #28952870)
- **InnoDB:** An invalid debug assertion was removed from the `temptable::Handler::primary_key_is_clustered` function. (Bug #28949332)
- **InnoDB:** An `ALTER TABLE ... EXCHANGE PARTITION` operation did not properly update column `table_id` values in the data dictionary. (Bug #28927005)
- **InnoDB:** Memory leaks discovered in the `innochecksum` utility were removed. (Bug #28917614, Bug #93164)
- **InnoDB:** A DDL operation that followed a failed attempt to create an index on a virtual column resulted in an assertion failure. (Bug #28825718)
- **InnoDB:** A performance regression was observed for partial update operations on compressed BLOBs less than or equal to 128KB in size. (Bug #28784301)
- **InnoDB:** Running aggregated queries raised `Valgrind` warnings. (Bug #28711717)
- **InnoDB:** A `CHECK TABLE` operation raised an assertion failure. A pointer to a local call stack variable was not set back to null before a function exit. (Bug #28525110)
- **InnoDB:** DDL log functions were modified to handle `ER_TOO_MANY_CONCURRENT_TRXS` errors. (Bug #28523127, Bug #92071)
- **InnoDB:** The purge thread failed to free LOB data pages. (Bug #28510599)
- **InnoDB:** Some DDL log table transactions were not rolled back prior to DDL log recovery. (Bug #28494969)
- **InnoDB:** A function invoked during `SHOW CREATE TRIGGER` processing that retrieves the table name did not perform the expected lowercase conversion. (Bug #28351038)
- **InnoDB:** The `INFORMATION_SCHEMA.INNODB_FOREIGN TYPE` column reported incorrect values. (Bug #28315651, Bug #91577)
- **InnoDB:** A Linux AIO handler function failed to check if completed I/O events succeeded. Thanks to Wei Zhao for the contribution. (Bug #27850600, Bug #90402)
- **InnoDB:** An assertion failure was raised in a check that determines if a transaction holds an implicit lock on a secondary index. A transaction that does not change the columns of a secondary index that includes virtual columns could be incorrectly determined to hold an implicit lock. (Bug #27491839)
- **InnoDB:** A function called by a `CREATE TABLE` thread attempted to access a table object after it was freed by a background thread.

  Thanks to Yan Huang for the patch. (Bug #27373959, Bug #89126)

- **InnoDB:** Two sessions concurrently executing an `INSERT ... ON DUPLICATE KEY UPDATE` operation generated a deadlock. During partial rollback of a tuple, another session could update it. The fix for this bug reverts fixes for Bug #11758237, Bug #17604730, and Bug #20040791. (Bug #25966845)

- **InnoDB:** When the method used to access a joined table was `const`, `InnoDB` attempted to unlock the matching row multiple times. (Bug #20939184)

- **InnoDB:** The `INDEX_LENGTH` value in `INFORMATION_SCHEMA.TABLES` was not updated when adding an index. (Bug #19811005)

- **Partitioning:** Some partitioning DDL statements were improperly rejected due to name validation checks which used the wrong table identifier. (Bug #29317007)

- **Partitioning:** While rolling back `ALTER TABLE ... COALESCE PARTITION`, the server sometimes attempted to lock and close partitions which had been dropped as a result of this statement. (Bug #28517446)

- **Partitioning:** An `AUTO_INCREMENT` key added to a partitioned table by an `ALTER TABLE` statement using `ALGORITHM=INPLACE` restarted on each partition. (Bug #92241, Bug #28573894)

- **Replication:** If the `WAIT_FOR_EXECUTED_GTID_SET()` function was used with a timeout value including a fractional part (for example, `1.5`), an error in the casting logic meant that the timeout was rounded down to the nearest whole second, and to zero for values less than 1 second (for example, `0.1`). The casting logic has now been corrected so that the timeout value is applied as originally specified with no rounding.

  Our thanks to Dirkjan Bussink for the contribution. (Bug #29324564, Bug #94247)

- **Replication:** The consistency level `AFTER` for the system variable `group_replication_consistency` did not include the consistency guarantees provided by `BEFORE_ON_PRIMARY_FAILOVER`. These consistency guarantees, which were already implicitly present with the `BEFORE` and `BEFORE_AND_AFTER` consistency levels, are now provided with `AFTER`. (Bug #29315752, Bug #94213)

- **Replication:** On Debian-based platforms (such as Ubuntu), if the hostname resolved to 127.0.1.1 - which is the default on these platforms - it was not possible to create a cluster using the default settings. Now, in such situations a proper validation of the instance is performed before creating a cluster and adding instances to it. (Bug #29246110)

- **Replication:** In a blocked group, if you set an invalid value for `group_replication_force_members` and then issued `STOP GROUP_REPLICATION`, the server could stop unexpectedly. (Bug #29119961)

- **Replication:** MySQL Server's behavior when the `mysql.gtid_executed` table cannot be accessed has been refactored to provide appropriate error responses and actions. The MySQL server now ensures that writes are permitted to the `mysql.gtid_executed` table when the server is in read only or super read only mode, so that the binary log file can still be rotated in these modes. If the `mysql.gtid_executed` table cannot be accessed for writes, and the binary log file is rotated for any reason other than reaching the maximum file size (`max_binlog_size`), the current binary log file continues to be used. An error message is returned to the client that requested the rotation, and a warning is logged on the server. If the `mysql.gtid_executed` table cannot be accessed for writes and `max_binlog_size` is reached, the server responds according to its `binlog_error_action` setting. If `IGNORE_ERROR` is set, an error is logged on the server and binary logging is halted, or if `ABORT_SERVER` is set, the server shuts down. (Bug #29111514)

- **Replication:** When issuing `STOP GROUP_REPLICATION` while the member was trying to assess whether or not it had lost majority, the server could stop unexpectedly. (Bug #29053128)

- **Replication:** When a `RESET MASTER TO` statement is used to specify the starting index number for binary log files, the maximum number that you can specify has been reduced from the maximum integer value to 2000000000. If the maximum integer value was specified, the server was not able to start up because no further binary log files could be created. The server also previously experienced a segmentation fault in that situation. (Bug #28980788, Bug #28995220)
- **Replication:** On a replication slave with GTIDs in use and binary logging disabled, an assertion was raised in debug mode when a DDL statement was filtered out by a table filter. (Bug #28965972)
- **Replication:** Two issues with the deserialization of statement based replication events in the binary log were corrected. (Bug #28889181, Bug #29028491)
- **Replication:** If an applier thread was stopped while it was in the process of opening a table, no error was set, which could result in a segmentation fault or assertion depending on the build type. Error handling is now correctly activated in this situation. (Bug #28864557)
- **Replication:** With GTIDs in use on the server, the master info log on a replication slave was being synchronized every time the master skipped a transaction using the auto-skip function. The process ends with a dummy heartbeat which is sent to the slave and caused a forced flush to the log, and this could have a large cumulative impact on the write load on the slave. The same issue could occur in a circular replication topology with events that originated from the same server and were therefore ignored, which were also handled by the slave with a forced flush to the log. The slave handling code has now been changed to remove the forced flush for heartbeat events and for ignored events received through circular replication, so that the master info log is only synchronized when appropriate (for example, when a `CHANGE MASTER` statement is issued, or the binary log is rotated). (Bug #28815555, Bug #85158)
- **Replication:** When an `ALTER TABLE` statement is used with a `DEFAULT` clause to specify an expression default value for a new column, and the expression default value refers to a nondeterministic function, the statement is unsafe for statement-based replication. Previously, such statements were also evaluated in terms of GTID consistency, which was not the appropriate check as the statements do not impact GTID consistency. Now, these statements are evaluated only for binary logging and are handled depending on the binary logging format in use. When `binlog_format` is set to `STATEMENT`, the statement is logged but a warning message is written to the error log. When `binlog_format` is set to `MIXED` or `ROW`, the statement is not executed and an error message is written to the error log. (Bug #28799939)
- **Replication:** In a replication group configured in single-primary mode (`group_replication_single_primary_mode=ON`, which is the default), if severe network delays affected the group, it was possible for the primary and the secondaries to reach different decisions on a transaction, which could lead to divergence in the `gtid_executed` sets on the members. The issue has now been fixed. (Bug #28768550, Bug #28966455, Bug #92690)
- **Replication:** When you use the `group_replication_force_members` system variable to force a new configuration for a group, the group communication engine (XCom) now checks that you have not included any group members that are currently unreachable. If any are found, the reconfiguration is disallowed and an error is returned. (Bug #28678845)
- **Replication:** `GRANT` statements that were written to the binary log were logged incorrectly in some cases, which could result in a `GRANT` statement that executed successfully on the master causing an error on the replication slave. (Bug #28643405, Bug #29155451, Bug #93750)
- **Replication:** If a storage engine has the capability to log in `STATEMENT` format but not in `ROW` format, when `binlog_format` is set to `STATEMENT`, an unsafe SQL statement should be logged and a warning message should be written to the error log. However, such statements were instead not executed and an error message was written to the error log, which is the correct behavior when `binlog_format` is set to `MIXED` or `ROW`. The issue has now been corrected so that unsafe statements are logged with a warning as expected when `binlog_format` is set to `STATEMENT`. (Bug #28429993, Bug #73936)

- **Replication:** It is possible for a replication group member to go offline briefly, then attempt to rejoin the replication group again before the group has detected its failure and been reconfigured to remove the member. Previously, in this situation, the rejoining member could participate in XCom's consensus protocol if it received and processed messages intended for its pre-crash incarnation. This could cause XCom to deliver different values for the same consensus round, because the rejoining member could make a different decision before and after failure. To prevent this situation, a rejoining member now ignores messages intended for its pre-crash incarnation. (Bug #27383487)

- **Replication:** On overloaded servers there was a possibility that when a member joined the group, the `VIEW_CHANGE_LOG_EVENT` event which marks that point was not logged in the correct place. This could lead to errors in the data transfer to the newly joining server and data divergence. Now, the `VIEW_CHANGE_LOG_EVENT` event is logged in the correct place in the binary log. In addition, warnings are logged about the delay in logging the event. (Bug #93347, Bug #28971594)

- **Replication:** When a member joined a group on server start, if the join process failed, for example because the server was incompatible with the group, there was a possibility that the offline member could still see another member as being online. Now, in such a situation the information shown in the `performance_schema.replication_group_members` table is restricted to the local member when it is `OFFLINE`. (Bug #92110, Bug #28533993)

- **Replication:** In the event of the recovery channel failing, unprocessed relay logs were being erased. (Bug #90671, Bug #27940732)

- **Group Replication:** `START GROUP REPLICATION` did not function properly when the port specified for the member's local address was busy. (Bug #29347285)

- **Group Replication:** When a host name was specified in the IP address whitelist for Group Replication (`group_replication_ip_whitelist`),IPv6 addresses were used for name resolution and whitelist comparison when an IPv4 address was also available. An IPv4 address should always be preferred for Group Replication connections. Now, if the host name resolves to an IPv4 address, any IPv6 addresses are not considered for comparison to the whitelist. (Bug #28841543)

- **Group Replication:** Previously, relay logs could not be rotated manually for the Group Replication `group_replication_applier` channel using the `FLUSH RELAY LOGS` statement. Due to this restriction, when encryption was enabled for binary log files and relay log files (`binlog_encryption=ON`), as available beginning with MySQL 8.0.14, the relay log file in use on that channel could not be rotated immediately if encryption was disabled again. The restriction had a similar impact on binary log master key rotation, as available from MySQL 8.0.16. The restriction has now been removed, and the `FLUSH RELAY LOGS` statement and corresponding internal requests now operate on the `group_replication_applier` channel as for any other channel, with the exception that if the request is received while a transaction is being applied, the request is performed after the transaction ends. The requester must wait while the transaction is completed and the rotation takes place. This behavior prevents transactions from being split, which is not permitted for Group Replication. (Bug #28684376)

- **Group Replication:** A replication group member could trigger a local view after being expelled from the group due to a loss of majority. This resulted in a message incorrectly stating that the member had resumed regular operation after the expulsion. Group Replication now checks before delivering a local view that the member has not been expelled. (Bug #27349236)

- **Group Replication:** If an invalid value was specified for `group_replication_communication_debug_options`, the Group Communication System set its corresponding internal variable to `GCS_DEBUG_NONE`, and the server returned the invalid value for a `SHOW VARIABLES` query. The value of this system variable is now checked during server initialization, and if an invalid value was specified, an error message is logged and Group Replication does not start automatically. (Bug #26729404)

- **macOS:** `CMake` 3.12.4 or higher (which forces `UseModernBuildSystem = NO`) is now required on macOS if building with Xcode rather than Makefiles. (Bug #28893131)
- **Microsoft Windows:** Validity testing for the `named_pipe_full_access_group` system variable did not account for `NULL` values. (Bug #29256690)
- **Microsoft Windows:** When multiple instances of `mysqld` were started with the `--no-monitor` option on the same host for same user, the `SHUTDOWN` command shut down the wrong server process. This fix creates a unique shutdown event name for use with `--no-monitor` by appending the process ID of the process. (Bug #28723675)
- **JSON:** The JSON path parser now propagates errors in the same way as most other components of the MySQL server, returning true on error and false on success. (Bug #28851426)
- **JSON:** Removed an unneeded type lookup in `Json_wrapper::get_datetime()`. (Bug #28851324)
- The `authentication_ldap_simple` plugin could enforce authentication incorrectly. (Bug #29637712)
- RPM package obsoletes were updated to enable successful upgrades from MariaDB to MySQL on EL8. (Bug #29413354)
- It was possible for the result of an outer join to contain a non-`NULL` row where a `NULL` extended row was expected. (Bug #29402481)

  References: This issue is a regression of: Bug #27808758.
- `SET PASSWORD FOR ...` could not be executed as a prepared statement. (Bug #29387041, Bug #94416)
- Builds under Visual Studio could fail while building MySQL Router libraries. (Bug #29382197)

  References: This issue is a regression of: Bug #29361890.
- Imported foreign keys did not work if defined before the referenced table. (Bug #29379078, Bug #94400)

  References: This issue is a regression of: Bug #28493257.
- When an error was raised while evaluating the `ESCAPE` clause of a `LIKE` expression at resolve time, the error status was not propagated to the caller. (Bug #29368521)
- Tablespace files for the `innodb_table_stats_backup` and `innodb_index_stats_backup` metadata backup tables were not removed after an in-place upgrade from MySQL 5.7 to MySQL 8.0. (Bug #29365552)
- While flattening a subquery, if a predicate which was always false was present, the MySQL Optimizer did not perform any kind of transformation, resulting in a subquery that was not prepared, and which asserted later when it was executed. To resolve this issue, when such a predicate is present in a subquery, the subquery's query expression is now unlinked from the query block. (Bug #29356132)
- Upgrade from MySQL 8.0.11, 8.0.12, or 8.0.13 to MySQL 8.0.14 or 8.0.15 failed if an event, routine, or trigger was defined with the `ALLOW_INVALID_DATES` SQL mode. The SQL mode identifier in the data dictionary was changed in MySQL 8.0.14, causing a migration failure. (Bug #29350955)
- RPM builds ignored the `WITH_SSL` configuration setting. (Bug #29347534)
- Length metadata for the `TO_SECONDS()` function was not always calculated correctly. (Bug #29321387)
- Conditions using windowing functions removed due to being always true or false were not always handled correctly. (Bug #29320484)

- `SET ROLE` statements could leak memory. (Bug #29304583)
- Upgrading from MySQL 5.7 to MySQL 8.0 on Windows failed with “Error 197 from SE while migrating tablespaces”. The error was due to an access share violation that occurred when attempting to open a tablespace file. (Bug #29292860)

  References: This issue is a regression of: Bug #28642608.
- The `COMPILATION_COMMENT_SERVER` value could be incorrect in RPM packages. (Bug #29284651)
- `CREATE TABLE IF NOT EXISTS` failed even if the table already exists if the new table definition had no primary key and the `sql_require_primary_key` system variable was enabled. (Bug #29283055, Bug #94134)
- A delete from a partitioned table from which all partitions had been pruned away was not always handled correctly. (Bug #29280186)
- The `CMake` check for the GNU `gold` linker could fail with Clang. (Bug #29278244)
- The data dictionary version was incremented for the MySQL 8.0.16 release. (Bug #29278241)
- An argument of the wrong type to a function used with `DISTINCT` was not always handled correctly. (Bug #29277571)
- Length metadata for the `QUOTE()` function was not always calculated correctly. (Bug #29276074)
- When evaluating `GREATEST()` or `LEAST()`, MySQL checked for correct signedness of the return value before checking it for `NULL`. (Bug #29275835)
- When strict SQL mode was not in effect, the values of some string functions returning `NULL` to indicate a result greater than `max_allowed_packet` were handled inconsistently behaviour, which could result in incorrectly sorted output and possibly other misbehaviors. (Bug #29272683)

  References: See also: Bug #97301, Bug #29133127.
- Logic that checks if upgrade is supported to a particular server version was inverted to check for server versions from which upgrade is not supported. (Bug #29270297)
- Event creation could store an incorrect repetition interval. (Bug #29269819, Bug #94085)
- A WHERE condition containing a view reference or an item created by a transformation was not always handled correctly. (Bug #29268867, Bug #29268698, Bug #28723669, Bug #29244238)
- `mysql_ssl_rsa_setup` failed to compile using GCC 9. (Bug #29245251)
- If `CMake` finds a `libtirpc` library that is too old to work with MySQL, it tries to use Sun RPC from `glibc` instead. (Bug #29240701)
- The server could fail to write slow queries to the slow query log due to incorrect calculation of query execution time. (Bug #29232684, Bug #93963)
- MySQL regular expression functions using positions employed internal indexes based on 16-bit chunks, rather than on codepoint positions. (Bug #29231490)
- On Windows, the MySQL MSI installer could fail to correctly detect whether the Visual Studio 2015 Redistributable was installed. (Bug #29227209)
- SDI `JSON` files did not include the `m_hidden` field of `Index_impl` objects. This made it hard to use the SDI `JSON` to recreate the `CREATE` statement for the table because `InnoDB` adds a number of hidden

indexes. The SDI `JSON` now includes the `m_hidden` field. This changes SDI format, so the SDI version number was increased to the current server version number. (Bug #29210646, Bug #93914)

- The position hint for the last row in a range frame was updated to be one row past the actual last row in the frame. (Bug #29201831)
- In a column definition, multiple constraint definitions were not accepted when the first was a `CHECK` constraint. (Bug #29191994)
- Error log information buffered during startup could be buffered too long if the server was performing an upgrade. (Bug #29189532)
- For nullable columns, if we find an expression that is always true except when the column is `NULL`, the expression is folded to *`column`* `IS NOT NULL`. When such an expression was nested, this caused `NULL` rows to be selected in error. To prevent this from happening, such an expressions when nested is now instead folded to `IF(`*`column`*` IS NULL, NULL, TRUE)`. (Bug #29179604)
- `PERIOD_ADD()` did not handle values greater than 32 bits in length for the *`period`* argument correctly on Windows platforms. (Bug #29175262)
- When `sql_auto_is_null` is enabled, a `WHERE` clause of the form `WHERE` *`auto_increment_col`* `IS NULL` is rewritten as `WHERE` *`auto_increment_col`* `= LAST_INSERT_ID()`. This transformation was only performed once per auto-incremented value, which made it difficult to know in advance whether the transformation would be performed. Now the transformation is performed unconditionally whenever `sql_auto_is_null` is enabled.

  In addition, the value returned by `LAST_INSERT_ID()` is now treated as unsigned, fixing a failure to match an auto-incremented value outside the range of a signed `BIGINT`. (Bug #29171668)
- On Debian and Ubuntu, installation operations in noninteractive mode ignored the `root` password, resulting in the `auth_socket` authentication plugin being installed by default. (Bug #29165407)
- The `harness_plugin_eventlog` declaration caused compilation errors in some build environments. (Bug #29160214)
- The logic for truncating or extending a decimal constant to the desired number of fractional digits during constant folding was lacking. Extra trailing zeros in the fraction could trigger an attempt to widen the faction, since the decision to widen was based on the number of non-zero fractional digits, rather than on total number of fractional digits, leading to an assert (in debug builds) in the internal function `widen_fraction()`. This issue is fixed by identifying where it is possible merely to truncate excess trailing zeroes. In this case, it is not necessary to adjust the comparison operator, and the constant can be replaced by one having fewer trailing zeroes instead. (Bug #29155439)
- On Windows, the internal function `get_mysql_time_from_str_no_warn()` did not always perform proper error checking. (Bug #29155126)

  References: See also: Bug #29175262.
- Under certain conditions, `RENAME TABLE` statements that renamed the same table multiple times could raise an assertion or cause a server exit. (Bug #29140407)
- For debug builds, starting the server with `--event-scheduler=DISABLED` could result in an assertion being raised for certain events. (Bug #29140298, Bug #93719)
- In debug builds, When strict SQL mode was not in effect, the `CONCAT()` and `CONCAT_WS()` functions raised an assertion if the result was longer than `max_allowed_packet`. (Bug #29133127)
- An out-of-range fractional part could produce incorrect timestamps in `SET SESSION timestamp` statements. (Bug #29120569, Bug #93600)

- The `mysql_service_component_sys_variable` service could access component system variables but not server or plugin system variables. (Bug #29113463)
- `ALTER TABLE ... CONVERT TO CHARACTER SET` could produce a memory-access error. (Bug #29058369, Bug #93603)
- The `mysql.tablespaces.name` column limit was 259 bytes, which was less than required for permitted identifier lengths. The column limit was raised to 268 bytes. (Bug #29053560, Bug #93587)
- For debug builds, for spatial computations that raised an assertion, partition-handling code ignored the error, resulting in a server exit. (Bug #29047811)
- A zero length LOB that was stored externally caused an assertion failure. (Bug #29047795)
- Handling of the `COMPILE_DEFINITIONS` and `COMPILE_FLAGS` `CMake` options was adjusted to avoid cross-compiling failures. (Bug #29041100)
- The server could try to read the first diagnostics area message even when the `max_error_count` system variable was set to zero, resulting in a memory-access error. (Bug #29031684)
- When comparing `DATE` values with constant strings, MySQL first tries to convert the string to a `DATE` and then to perform the comparison. When the conversion failed, MySQL executed the comparison treating the `DATE` as a string, which could lead to unpredictable behavior. Now in such cases, if the conversion of the string to a `DATE` fails, the comparison fails with `ER_WRONG_VALUE`. (Bug #29025656)

  References: See also: Bug #95466, Bug #29812087.
- With the `--users` option, `mysqlpump` wrote `CREATE USER` and `GRANT` statements to the output, but too late to apply to the other objects created by the dump. Consequently, restoring the dump file created the user accounts too late to apply to other objects created by the file. `mysqlpump` now writes user accounts to the dump file before other objects. (Bug #29023216)
- For syntax errors at the position of the `WITH` keyword, the parser error message identified the problem at the incorrect location. (Bug #29022263)
- MySQL builds using recent versions of the International Components for Unicode (ICU) now return `ER_REGEXP_ILLEGAL_ARGUMENT` rather than the generic error `ER_REGEXP_ERROR` for malformed regular expressions. (Bug #29016798)
- On Windows, the LDAP library could not be found if the file system was case sensitive. (Bug #29016220)
- Values selected from the `TABLE_COMMENT` column of the `INFORMATION_SCHEMA.TABLES` table could be truncated. (Bug #29014272, Bug #93451)
- Data layer memory leaks related to unfreed schemas were discovered in ASAN and Valgrind builds. (Bug #29008688)
- A `GROUP BY` query with `ROLLUP` incorrectly raised `ER_WRONG_FIELD_WITH_GROUP` when used with an `ORDER BY` clause that contained an expression.

  The same issue also led to incorrect ordering of the result from a `GROUP BY` query with ROLLUP when ordering on a function. (Bug #29006668, Bug #29054096)
- The parser accepted multiple `COLLATE` clauses in generated column definitions. It now accepts a single `COLLATE` clause. (Bug #28997518)
- `INSERT ... ON DUPLICATE KEY UPDATE` could perform incorrect updates. (Bug #28995498, Bug #93410)

References: This issue is a regression of: Bug #26188578.

- A damaged `mysql.user` table could cause a server exit. (Bug #28986737)
- Reading rows from the Performance Schema `data_locks` or `data_lock_waits` table could return an unbounded number of rows during a scan, resulting in a server exit due to excessive memory allocation. This situation now produces an error. (Bug #28977428, Bug #87748)
- Calculating the cost of materialization for a `LATERAL` join on an empty table led to an assertion. Now in such cases, the query cost is reported as zero. (Bug #28976533)
- For debug builds, an invalid `utf8` character in the comment string for a column type in `CREATE` or `ALTER` DDL statements raised an assertion. (Bug #28972424, Bug #93321)
- The `CMake` check for `tirpc` headers now falls back to using `pkgconfig`, to enable finding the headers on more Linux platforms. This requires that `pkgconfig` be installed. (Bug #28970313, Bug #93341, Bug #28997093)
- `mysqld` could undergo an unplanned shutdown when a component in the `WHERE` clause of a subquery was non-deterministic, in this particular case due to referencing a user variable that was also set in the same statement. This issue is fixed by ignoring predicates having non-deterministic components.

  In addition, since non-deterministic subqueries as a general rule should be evaluated per outer row, these should not be targets for materialization strategy. This is prevented by not decorrelating the non-deterministic predicates. When possible, the same strategy selection for non-semijoin and non-deterministic subqueries is also now enforced. (Bug #28970261)
- `ALTER TABLE` statements to change a table storage engine could hang when user-level locks and explicit table locks had been acquired earlier. (Bug #28966941)
- `COUNT()` with `LIMIT ... OFFSET` returned a different result from that obtained with other aggregate functions used with this clause. (Bug #28961843)
- Extracting the value of the `MYSQL_HOME` environment variable could change the value of the variable stored following `MYSQL_HOME` in the environment. (Bug #28960613)
- `Item_subselect::walk_body()` now walks `FROM` clauses. (Bug #28955358)
- An issue was uncovered in query plan for a query with the following two `IN` subqueries:

  - A subquery with a derived table containing an outer reference to the topmost query
  - A subquery with a derived table, which was not outer-correlated, was semijoined, and was not lateral.

  For the second subquery: calculation of the map of dependencies neglected to exclude irrelevant lateral tables, and wrongly included dependencies of the derived table in the first subquery. (Bug #28954838)
- An error generated while evaluating a constant expression in certain `GROUP BY` queries was not checked for in a timely fashion. (Bug #28949452)
- When comparing a `DATE` or `DATETIME` value with a string MySQL first tries to convert the string to the same type (`DATE` or `DATETIME`) and then to compare the two as values of that type. If this conversion fails, MySQL raises an `Incorrect date value` warning and then falls back to comparing the values as strings, which is the expected behavior. In some cases, even though the converted string did not reflect a valid `DATE` or `DATETIME` value, no warning was issued so and it was compared with the `DATE` or `DATETIME` value as a value of that type. Now in such cases, the converted string is always checked

for validity as a `DATE` or `DATETIME` before being compared, unless the `ALLOW_INVALID_DATES` server SQL mode has been set explicitly. (Bug #28940878)

- Long client host names could cause unexpected server behavior. (Bug #28936359)
- Substitution of a large number of parameters in a single prepared statement could lead to excessive and unnecessary reallocation of memory. This is fixed by having `String::replace()` use exponential buffer growth as `String::append()` does. (Bug #28929977)
- The data dictionary cache could become out of sync with data dictionary contents due to failure to check the result of a transaction-related operation. (Bug #28923782, Bug #93196)
- Determination of the number of online CPUs available to the `mysqld` process is now more accurate. Thanks to Daniel Black for the contribution. (Bug #28907677, Bug #93144)
- The `INFORMATION_SCHEMA.TABLES TABLE_COMMENT` column reported the wrong error message due to an invalid view. (Bug #28901919)
- After upgrading, the `INFORMATION_SCHEMA.TABLES` view reported “View '*`view_name`*' references invalid table(s) or column(s) or function(s) or definer/invoker of view lack rights to use them” for several `sys` schema views. The upgrade process did not update the referencing view state. (Bug #28901821)
- Creating histograms for large tables (millions of rows) having many distinct column values took an excessive amount of time.

  The fix make this process marginally slower for data sets with few distinct values and consumes slightly more memory than previously, but for large data sets with many distinct values it is significantly faster. (Bug #28888936)
- Upgrade from MySQL 5.7 to MySQL 8.0 failed for instances with a an `InnoDB` 4K page size due a key length error that should be suppressed during upgrade. (Bug #28884503)
- An incorrectly configured keyring plugin could cause a server exit. (Bug #28876033)
- Two issues were found in cases involving use of the `COUNT()` function:

  - When the argument passed to `COUNT()` was a nullable expression, it was possible for the function to return an incorrect result.
  - A query using `COUNT()` together with an `EXISTS` subquery returned an incorrect result.

  (Bug #28857990, Bug #29240516)
- Commands for invoking `ccache` during MySQL builds were generated incorrectly. (Bug #28841612, Bug #92927)
- Previously, for `INSERT` statements that listed insert column names multiple times, only the first duplicated name was reported. Now, all duplicated names are reported. (Bug #28836669)
- When using `ROLLUP`, the `GROUP BY` may yield `NULL` in some rows. Expressions referencing any of these rows could lose the `NULL`, producing an incorrect result. (Bug #28836345)
- After a failed attempt to open a table, attempts to discover the table from the storage engine did not check for an error return, which could cause the server to hang if an error occurred. (Bug #28828450)
- The server could exit when trying to drop a user who had been granted a particular role. (Bug #28817441)
- MySQL Router compilation failed if MySQL was configured with `-DWITH_MYSQLX=0`. (Bug #28811356)

- A function called during data dictionary upgrade temporarily modified the `avoid_temporal_upgrade` parameter so that a check for old temporal types is always performed. Because multiple threads could call the function concurrently, a check was added to ensure that the parameter value could be safely modified during data dictionary upgrade. (Bug #28805429)
- An `EXISTS` subquery is converted to a semijoin operation. The optimizer chooses a materialization lookup strategy for this semijoin, but because the subquery is not correlated with the outer query block, there were no keys to use for the lookup, which caused the parent query to fail. To solve this issue, we use two equal constant items as keys, to ensure that the materialized query gets the constant as a key (and so that the materialized table consists of at most one row). (Bug #28805105)
- The `Last_query_cost` status variable now shows the cost of queries which contain multiple blocks such as subqueries or unions. (Previously, this variable was set only for simple queries consisting of a single query block.) (Bug #28786951)
- The message displayed by `mysqld` when a fatal signal occurred has been simplified and made more informative. It also does not display the calculation of memory usage, which had become out of date and inaccurate. (Bug #28773322, Bug #92731)
- The `skip_name_resolve` system variable could be persisted using `SET PERSIST_ONLY` to enable it, but not to disable it. (Bug #28749668)
- Removed obsolete and unnecessary condition pushdown handling in iterators except where required by single table update and delete queries. (Bug #28745859)
- `IS NOT NULL` predicates are added as part of early filtering of `NULL` for ref access performed by the Optimizer. For queries having a star-join topology, redundant duplicates of these predicates were added, which made evaluation of affected `WHERE` conditions less efficient. (Bug #28727717)
- `CMake` could generate an incorrect order of system includes when MySQL was configured to use the bundled version of a system library. (Bug #28727631, Bug #92615)
- Some numeric operations involving double to long long conversions could return different results on Windows and Linux. (Bug #28706832)
- An attempt to access a null pointer could occur during prepared statement execution. (Bug #28692136)
- `FROM_UNIXTIME()` returned an out-of-range value if passed an argument that, when rounded up, exceeded the epoch value. Now it returns `NULL` as for other out-of-range values. (Bug #28671811, Bug #92501)
- A replicated DDL operation could result in a 'duplicate entry on primary key' error during recovery on servers started with the `slave_parallel_workers` system variable. (Bug #28670843)
- Empty host names in accounts could cause the server to misbehave. (Bug #28653104)
- The MySQL 5.7 to MySQL 8.0 upgrade process did not check for the existence of tablespace files, which could result in an inconsistent data dictionary. (Bug #28642608)
- Type casting of `ENUM` behaved differently in subqueries than not in subqueries. (Bug #28547906, Bug #92173)
- Statements that caused truncation of floating-point values could be executed as prepared statements even when the server SQL mode included `STRICT_TRANS_TABLES`. In addition, if the binary logging mode was `MIXED`, such statements were propagated to the slave, where they caused errors. (Bug #28546855)
- Reducing the value of `thread_cache_size` at runtime did not reduce the size of the thread cache. (Bug #28508923, Bug #92024)

- Some error messages still referred to the `PASSWORD()` function, which has been removed. (Bug #28498714)
- The `Aborted_connects` status variable was not incremented for unsuccessful connection attempts, if connections were managed by the `thread_pool` plugin. (Bug #28490126)
- `mysqladmin shutdown` did not wait for `mysqld` to shut down. (Bug #28466137, Bug #91803)

  References: This issue is a regression of: Bug #25364806.
- Specifying `CURRENT_USER` as the user in `GRANT` statements could fail. (Bug #28454014)
- Repeated invocations of stored procedures which executed queries undergoing short-circuit evaluation were not always handled correctly. (Bug #28379655)
- Keyring migration should require only read access to the source keyring, but failed unless the user had write access. (Bug #28339014)
- If a user performing a keyring migration did not have write access to the keyring file, the migration failed but reported success in its final error log message. (Bug #28330922)
- During `FLUSH STATUS` execution, the Performance Schema unnecessarily aggregated session status to global status, causing double counts for some status variables. (Bug #28291258, Bug #91541)
- `mysql_secure_installation` no longer attempts to read a password from the `.mysql_secret` file. This was created by `mysql_install_db`, a program that has been removed. (Bug #28235716, Bug #91270)
- Some status variable values could temporarily increase before returning to their original value. (Bug #27839644, Bug #90351)
- Client programs did not exit if `--ssl-fips-mode` was given but FIPS mode could not be set to the specified option value. (Bug #27809371)
- Executing `ALTER INSTANCE ROTATE INNODB MASTER KEY` and migrating keys from the `keyring_file` plugin to the `keyring_encrypted_file` plugin could make encrypted tables unusable. (Bug #27760952)
- When upgrading from MySQL Community to Commercial, the `root` password prompt was shown even when the data directory existed. (Bug #27741998)
- In the client/server protocol, malformed packets for prepared statements could go undetected and cause an assertion to be raised. (Bug #27627731)
- Changes were made in session connect and disconnect handling for threads in order to speed up query throughput when using the X Plugin. Previously, a session (`Srv_session`) was attached to and detached from the current thread with every command; now, this thread switch takes place only when disconnecting the current session, or when the current thread and the thread being attached to belong to different plugins. (Bug #27463277)
- The `PERSIST` and `PERSIST_ONLY` keywords were reserved by mistake. They are now nonreserved keywords. (Bug #25220656)
- Installing and uninstalling a plugin concurrently with client connection activity could cause a server exit. (Bug #22980441)
- The `LOAD_FILE()` function could fail for files for which `stat()` should be considered only advisory, such as files under `/proc`. (Bug #18394503, Bug #72027)

- A query using an `ORDER BY` clause failed silently when the table being queried had an implicit full-text index and the sort buffer was of insufficient size to contain the sorted keys. (Bug #93241, Bug #28940361)
- A loadable function returning a string value now sets an explicit return type. Depending on the arguments passed to the function, this is one of `VARCHAR`, `MEDIUMBLOB`, or `LONGBLOB`. (Bug #92890, Bug #28828169)
- Made a comparison in the internal method `Item_result::item_cmp_type()` more efficient. Our thanks to Daniel Black for the contribution. (Bug #92784, Bug #28796107)
- A windowing function employed in an arithmetic expression produced an incorrect result when the query containing it used `DISTINCT`. (Bug #92503, Bug #28672483)
- Some queries involving complex joins leaked file handles. (Bug #90902, Bug #28039829)
- Fedora packaging now supports Fedora 30.
- Ubuntu 14.04 and SLES 11 are EOL, and no longer supported.

# Changes in MySQL 8.0.15 (2019-02-01)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

## Bugs Fixed

- **InnoDB:** After a checkpoint operation persisted modifications to data dictionary metadata, there was potential for new metadata changes to be lost under certain circumstances. (Bug #29120297)
- **Group Replication:** Group Replication was unable to function in MySQL 8.0.14 if IPv6 support was disabled at the operating system level, even if the replication group did not use any IPv6 addresses. (Bug #29249542, Bug #94004)
- If MySQL was running on the host system and within Docker, it was not possible to update or remove MySQL on the host system. (Bug #28244773, Bug #91405)

# Changes in MySQL 8.0.14 (2019-01-21)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

- Account Management Notes
- Audit Log Notes
- Compilation Notes
- Component Notes
- Configuration Notes
- Deprecation and Removal Notes
- SQL Function and Operator Notes
- Logging Notes

- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Pluggable Authentication
- Security Notes
- Spatial Data Support
- SQL Syntax Notes
- Thread Pool Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Account Management Notes

- Previously, each MySQL user account was permitted to have a single password. MySQL now permits an account to have dual passwords, designated as primary and secondary passwords. This capability enables phased password changes to be performed seamlessly in complex multiple-server systems, without downtime. To support dual-password capability, the `ALTER USER` and `SET PASSWORD` statements now have a `RETAIN CURRENT PASSWORD` clause that saves the current password as the secondary password when you assign an account a new primary password. `ALTER USER` also has a `DISCARD OLD PASSWORD` clause to discard a secondary password that is no longer needed. See Password Management.

  **Important**

  The implementation of dual-password capability involves a change to the structure of the `mysql.user` system table. If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate this system database change. Until this is done, *password changes are not possible.*

  (WL #11540)

## Audit Log Notes

- The audit API now enables applications to add their own message events to the audit log using the new `audit_api_message_emit` component, which includes an `audit_api_message_emit_udf()` loadable function. See The Audit Message Component. (WL #11542)

## Compilation Notes

- The minimum version of the Boost library for server builds is now 1.68.0. (Bug #28478497)

## Component Notes

- A new `host_application_signal` component service is available to enable components to deliver signals to the host application. For example, the service enables replication components to send a shutdown signal to the server. (WL #12003)

## Configuration Notes

- Previously, the `COMPILATION_COMMENT` `CMake` option was used by the server (for example, to set the `version_comment` system variable) and by other programs. However, when the value contained the word “server”, it was inappropriate for use by other programs. The server now uses the new `COMPILATION_COMMENT_SERVER` option. Other programs continue to use `COMPILATION_COMMENT`. (Bug #28888510)

- The content of the `.gitignore` file has been cleaned up. Much of this file was inherited from its `.bzrignore` predecessor and was not relevant. One implication of this cleanup is that in-source builds are disallowed. (Bug #28341794, Bug #91626)

- MySQL Server now permits a TCP/IP port to be configured specifically for administrative connections. This provides an alternative to the single administrative connection that is permitted on the network interfaces used for ordinary connections even when `max_connections` connections are already established. The administrative network interface has these characteristics:

  - The interface is enabled only if the `admin_address` system variable is set at startup to indicate the IP address for it. If `admin_address` is not set, the server maintains no administrative interface.

  - The `admin_port` system variable specifies the interface TCP/IP port number (default 33062).

  - There is no limit on the number of administrative connections, but connections are permitted only for users who have the `SERVICE_CONNECTION_ADMIN` privilege.

  - The `create_admin_listener_thread` system variable enables DBAs to choose at startup whether the administrative interface has its own separate thread. The default is `OFF`; that is, the manager thread for ordinary connections on the main interface also handles connections for the administrative interface.

  Thanks to Facebook for suggesting the idea (and for contributing code, although it was not used). (Bug #27847672, Bug #90395, WL #12138)

## Deprecation and Removal Notes

- The deprecated `resolveip` and `resolve_stack_dump` utilities have been removed and are no longer included in MySQL distributions. `nslookup`, `host`, or `dig` can be used instead of `resolveip`. Stack traces from official MySQL builds are always symbolized, so there is no need to use `resolve_stack_dump`. (WL #12619, WL #12620)

## SQL Function and Operator Notes

- The `JSON_ARRAYAGG()` and `JSON_OBJECTAGG()` aggregate functions now can be used as window functions, if an `OVER` clause is present. See Aggregate Function Descriptions. (WL #11574)

## Logging Notes

- A new system variable, `log_slow_extra`, if enabled, causes the server to write additional fields to slow query log lines that provide information about slow statements. In addition, `SET` lines written to the log to indicate statement timestamps now use the time from the beginning of statement execution, rather than the time at the end of execution. See The Slow Query Log. Thanks to Facebook for the contribution on which this feature is based. (Bug #27535580, Bug #89637, WL #12393)

- Binary log files and relay log files can now be encrypted, helping to protect these files and the potentially sensitive data contained in them from being misused by outside attackers, and also from unauthorized viewing by users of the operating system where they are stored.

You enable encryption on a MySQL server by setting the new `binlog_encryption` system variable to `ON`. `OFF` is the default. The system variable sets encryption on for binary log files and relay log files. When you first start the server with encryption enabled, a new binary log encryption key is generated before the binary log and relay logs are initialized. This key is used to encrypt a file password for each binary log file (if the server has binary logging enabled) and relay log file (if the server has replication channels), and further keys generated from the file passwords are used to encrypt the data in the files.

If you activate encryption while the server is running, a new binary log encryption key is generated at that time, and the binary log files and relay log files are rotated so that the new and subsequent files are encrypted. If you deactivate encryption by changing the `binlog_encryption` system variable to `OFF`, the binary log file and relay log files are rotated immediately and all subsequent logging is unencrypted. Previously encrypted files are not automatically decrypted, but the server is still able to read them. (The `SHOW BINARY LOGS` statement now shows whether each binary log file is encrypted or unencrypted.) `SUPER` privileges or the new `BINLOG_ENCRYPTION_ADMIN` privilege are required to activate or deactivate encryption while the server is running.

The encryption algorithm used for the files, the AES (Advanced Encryption Standard) cipher algorithm, is built in to MySQL Server and cannot be configured. The binary log encryption keys used to encrypt the file passwords for the log files are 256-bit keys that are generated specifically for each MySQL server instance using MySQL Server's built-in keyring service. The binary log encryption key that is currently in use on the server is called the binary log master key.

The new `binlog_rotate_encryption_master_key_at_startup` system variable controls whether the binary log master key is automatically rotated when the server is restarted. If this system variable is set to `ON`, a new binary log encryption key is generated and used as the new binary log master key whenever the server is restarted. If it is set to `OFF`, which is the default, the existing binary log master key is used again after the restart.

Note that when encryption is active for a MySQL server instance, only the data at rest that is written to the binary log files and relay log files is encrypted. The data in motion in the replication event stream, which is sent to MySQL clients including `mysqlbinlog`, is always in unencrypted format, so it must be protected in transit by the use of connection encryption. The data in use that is held in the binary log transaction and statement caches during a transaction, and any data that exceeds the space available in those caches and is therefore stored in a temporary file on disk, is also in unencrypted format. The temporary files and caches are deleted when the thread that handles the transaction ends. (WL #10957)

- Server logging behavior has changed with respect to error log messages generated prior to processing startup options that specify logging configuration. Previously, the server generated messages with the default timestamp, format, and verbosity level; buffered them; and then flushed them after the error log configuration became known. Because these early messages used the default logging configuration, they could differ from what is specified by the startup options.

  Now, the server buffers log events rather than formatted log messages. This enables it to retroactively apply configuration settings to those events after the settings are known, with the result that flushed messages use the configured settings, not the defaults. For more information, see Error Log Output Format. (WL #11875)

## Optimizer Notes

- Previously, derived tables and common table expressions could not contain outer references. Outer references are now permitted. (WL #461)

## Packaging Notes

- Ubuntu 18.10 and Fedora 29 install OpenSSL 1.1.1 by default, but OpenSSL 1.1.1 is not fully supported by MySQL. To install MySQL, the OpenSSL 1.0.2 compatibility package must be installed. (Bug #28981868)

## Performance Schema Notes

- The Performance Schema statement event tables (`events_statements_current`, `events_statements_history`, and `events_statements_history_long`) now have a `STATEMENT_ID` column that indicates the query ID maintained by the server at the SQL level. Column values are unique for the server instance because they are generated using a global counter that is incremented atomically. (WL #12165)

## Pluggable Authentication

- If the LDAP port number is configured as 636 or 3269, the plugin now uses LDAPS (LDAP over SSL) instead of LDAP. The port number is settable using the `authentication_ldap_sasl_server_port` or `authentication_ldap_simple_server_port` system variable. (LDAPS differs from `startTLS`.) (Bug #28743563)

- Previously, for LDAP authentication with proxying, LDAP authentication plugins used the first group name returned by the LDAP server as the MySQL proxied user account name. The authentication string for a MySQL account now can specify a list of groups to match, in preference order, and can optionally map the matching group name to a specified MySQL proxied user name. See LDAP Pluggable Authentication. (WL #12005)

## Security Notes

- The OpenSSL libraries bundled with MySQL on some platforms (Windows, macOS, and Generic Linux) have been upgraded to version 1.0.2q. On all other platforms, MySQL uses the system installed OpenSSL. Issues fixed in the new OpenSSL version are described at http://www.openssl.org/news/vulnerabilities.html. (Bug #28988091)

- To affect subsequent server restarts, the `SET PERSIST` and `SET PERSIST_ONLY` statements enable system variables to be persisted to the `mysqld-auto.cnf` option file in the data directory. However, some system variables cannot be persisted (for example, because they involve sensitive data). Consequently, they cannot be set at runtime from within a session conducted by a remote administrator, and thus require the administrator to log into the server host and directly modify a `my.cnf` option file.

  MySQL now permits users to perform runtime administration of many previously nonpersistible system variables, so that they can be persisted under certain restrictive conditions. To enable this capability, designate an SSL certificate X.509 Subject value that signifies the ability to persist these restricted system variables, and set the new `persist_only_admin_x509_subject` system variable to that Subject value. Users who connect to the server using an encrypted connection and supply an SSL certificate with the designated Subject value then can use `SET PERSIST_ONLY` to persist persist-restricted system variables. For more information, see Nonpersistible and Persist-Restricted System Variables. (WL #12086)

- For most system variables, setting the session value requires no special privileges and can be done by any user to affect the current session. For some system variables, setting the session value can have effects outside the current session and thus is a restricted operation that can be done only by users who have a special privilege. Previously, either `SYSTEM_VARIABLES_ADMIN` or `SUPER` qualified as such a privilege, but both privileges also permit operations other than setting session variables. The new

SESSION_VARIABLES_ADMIN privilege makes it possible to grant users *only* the ability to set restricted session variables without also enabling other operations.

Any operation permitted by SESSION_VARIABLES_ADMIN is also permitted by SYSTEM_VARIABLES_ADMIN or SUPER, so any user who already has one of the latter privileges effectively has SESSION_VARIABLES_ADMIN by implication and need not be granted SESSION_VARIABLES_ADMIN explicitly. However, if a user has been granted SYSTEM_VARIABLES_ADMIN or SUPER only for the purpose of enabling that user to modify restricted session system variables, an administrator can reduce the user's privilege footprint by revoking SYSTEM_VARIABLES_ADMIN and SUPER, and granting SESSION_VARIABLES_ADMIN instead. For instructions, see System Variable Privileges.

These previously restricted session variables required SYSTEM_VARIABLES_ADMIN or SUPER but now can also be set with SESSION_VARIABLES_ADMIN:

```
binlog_format
binlog_row_image
binlog_row_value_options
binlog_rows_query_log_events
debug
debug_sync
default_collation_for_utf8mb4
explicit_defaults_for_timestamp
gtid_next
histogram_generation_max_mem_size
original_commit_timestamp
sql_log_bin
sql_log_off
sql_require_primary_key
```

These previously unrestricted session variables now are restricted and setting them requires at least SESSION_VARIABLES_ADMIN (they can also be set by users who have SYSTEM_VARIABLES_ADMIN or SUPER):

```
auto_increment_increment
auto_increment_offset
binlog_direct_non_transactional_updates
bulk_insert_buffer_size
character_set_filesystem
character_set_database
collation_database
pseudo_slave_mode
pseudo_thread_id
rbr_exec_mode
transaction_write_set_extraction
```

(WL #12217)

## Spatial Data Support

- The ST_Distance() function now takes an optional third argument enabling the unit for the return value to be specified. Permitted units are those listed in the new INFORMATION_SCHEMA ST_UNITS_OF_MEASURE table. See Spatial Relation Functions That Use Object Shapes, and The INFORMATION_SCHEMA ST_UNITS_OF_MEASURE Table. (WL #11000)

## SQL Syntax Notes

- A derived table now may be preceded by the LATERAL keyword to specify that it is permitted to refer to (depend on) columns of preceding tables in the same FROM clause. A derived table specified with LATERAL can occur only in a FROM clause, either in a list of tables separated with commas or in a join

specification (`JOIN`, `INNER JOIN`, `CROSS JOIN`, `LEFT [OUTER] JOIN`, or `RIGHT [OUTER] JOIN`). Lateral derived tables make possible certain SQL operations that cannot be done with nonlateral derived tables or that require less-efficient workarounds. See Lateral Derived Tables.

**Note**

`LATERAL` now is a reserved word and cannot be used as an identifier without identifier quoting.

(WL #8652)

## Thread Pool Notes

- The `INFORMATION_SCHEMA` tables that accompany the thread pool plugin have been migrated to be available as Performance Schema tables. The `INFORMATION_SCHEMA` tables now are deprecated and will be removed in a future MySQL version. Applications should transition away from the old tables to the new tables. For example, if an application uses this query:

  ```
  SELECT * FROM INFORMATION_SCHEMA.TP_THREAD_STATE;
  ```

  The application should use this query instead:

  ```
  SELECT * FROM performance_schema.tp_thread_state;
  ```

  For more information, see Performance Schema Thread Pool Tables. (WL #11547)

## X Plugin Notes

- X Plugin now includes the 5-digit SQLSTATE error code in its error handling class. Previously, the SQLSTATE error code was returned to clients for SQL errors, but only the MySQL-specific error number was exposed. (Bug #28735058)
- When querying collections of documents, if boolean values were used as arguments for a placeholder in an SQL query, unexpected results were returned. A new translation specialization has now been added for boolean values so that they are handled correctly in this situation. (Bug #28227037)
- X Protocol now always converts retrieved data to the `utf8mb4` character set (using the `utf8mb4_general_ci` collation) before returning it. (Bug #28180155)
- X Protocol now supports SQL prepare functionality. (WL #9270)

## Functionality Added or Changed

- **InnoDB:** Disabling the `innodb_buffer_pool_in_core_file` variable reduces the size of core files by excluding `InnoDB` buffer pool pages. To use this variable, the `core_file` variable must be enabled and the operating system must support the `MADV_DONTDUMP` non-POSIX extension to `madvise()`, which is supported in Linux 3.4 and later. For more information, see Excluding Buffer Pool Pages from Core Files.

  Thanks to Facebook for the contribution. (Bug #27724476, Bug #90144)

- **InnoDB:** By default, undo logs reside in two undo tablespaces that are created when the MySQL instance is initialized.

  Additional undo tablespaces can be created in a chosen location at runtime using `CREATE UNDO TABLESPACE` syntax.

  ```
  CREATE UNDO TABLESPACE tablespace_name ADD DATAFILE 'file_name.ibu';
  ```

Undo tablespaces created using `CREATE UNDO TABLESPACE` syntax can be dropped at runtime using `DROP UNDO TABLESPACE` syntax.

```
DROP UNDO TABLESPACE tablespace_name;
```

`ALTER UNDO TABLESPACE` syntax can be used to mark an undo tablespace as active or inactive.

```
ALTER UNDO TABLESPACE tablespace_name SET {ACTIVE|INACTIVE};
```

A `STATE` column that shows the state of a tablespace was added to the `INFORMATION_SCHEMA.INNODB_TABLESPACES` table. An undo tablespace must be in an `empty` state before it can be dropped.

The previously deprecated `innodb_undo_tablespaces` variable is no longer configurable and will be removed in a future MySQL version.

For more information, see Undo Tablespaces. (WL #9508)

- **InnoDB:** `InnoDB` now supports parallel clustered index reads, which can improve `CHECK TABLE` performance. This feature does not apply to secondary index scans. The `innodb_parallel_read_threads` session variable must be set to a value greater than 1 for parallel clustered index reads to occur. The default value is 4. The actual number of threads used to perform a parallel clustered index read is determined by the `innodb_parallel_read_threads` setting or the number of index subtrees to scan, whichever is smaller. (WL #11720)

- **InnoDB:** The `ADD DATAFILE` clause of the `CREATE TABLESPACE` statement is now optional, which permits users without the `FILE` privilege to create tablespaces. A `CREATE TABLESPACE` statement executed without an `ADD DATAFILE` clause implicitly creates a tablespace data file with a unique file name. (WL #12236)

- **InnoDB:** When the server is started with `--innodb-dedicated-server`, the size and number of log files are now determined by the buffer pool which has been set by this option. Previously, log file size was set according to the amount of memory detected on the server, and the number of log files was not set automatically. See Enabling Automatic InnoDB Configuration for a Dedicated MySQL Server. (WL #12300)

- **Replication:** Two new session variables have been added for internal use by replication. `original_server_version` and `immediate_server_version` support cross-version replication by transmitting the MySQL server release numbers associated with a transaction through the replication topology. `original_server_version` holds the MySQL Server release number of the server where a transaction was originally committed (for example, `80014` for a MySQL 8.0.14 server instance). `immediate_server_version` holds the MySQL Server release number of the server that is the immediate master in a replication topology. If either of those servers, or another intervening server in the replication topology, is at an older release that does not support these session system variables, their values are set to 0.

  With this information the slave can correctly process data originating from a master at an older release, by recognizing where syntax changes or semantic changes have occurred between the releases involved and handling these appropriately. The information can also be used in a Group Replication environment where one or more members of the replication group is at a newer release than the others. The value of the variable can be viewed in the binary log for each transaction (as part of the `Gtid_log_event`, or `Anonymous_gtid_log_event` if GTIDs are not in use on the server), and could be helpful in debugging cross-version replication issues. (WL #11879)

- **Replication:** When running a group in single-primary mode, in the event of a new primary being elected while there were transactions held in the backlog to be applied, there was a chance

that a read operation against the new primary could return a stale value. Now, you can use the `group_replication_consistency` variable to control how a group behaves in this situation. When `group_replication_consistency` is set to `EVENTUAL`, a new primary responds to read requests even when there is a backlog which has not yet been applied, which matches the previous behavior and comes with the risk that a client could read old values while any backlog is being applied. Writes to the new primary fail during this period because it is has `super_read_only` mode enabled. When `group_replication_consistency` is set to `BEFORE_ON_PRIMARY_FAILOVER`, any new read or write queries against a newly elected primary that is applying backlog from the old primary are held until the backlog is applied. This ensures that clients always read the newest value which they have written, but also means that clients might have to wait until the backlog has been applied before they can read from the new primary. (WL #11123)

References: See also: Bug #26004894.

- **Group Replication:** The Group Communication System (GCS) and group communication engine (XCom, a Paxos variant) for Group Replication now provide full support for IPv6, so replication group members can use IPv6 addresses as an alternative to IPv4 addresses for internal group communications. The localhost address for IPv6, and the private subnetwork addresses for IPv6 (unique-local addresses and link-local unicast addresses), are added to the automatic whitelist for Group Replication for use if no manual whitelist is specified.

  If all members of a replication group are at a MySQL server version that supports the use of IPv6 addresses for Group Replication, the group can contain a mix of members using IPv6 addresses and members using IPv4 addresses. Joining members must provide whitelisted IP addresses or host names that match the protocols offered by the seed members for connection, but the joining member's main identifying address or host name (`group_replication_local_address`) can use either protocol. If a member uses a host name that resolves to both an IPv4 and an IPv6 address, the IPv4 address is always used for Group Replication connections.

  If any or all existing members of a replication group use an older MySQL Server version without support for the use of IPv6 addresses for Group Replication, joining members must present an IPv4 address for group communications in the `group_replication_local_address` option. When every group member has been upgraded, the group can be migrated to IPv6 addresses. (Bug #26088469, Bug #27757729, Bug #90217, WL #11926)

- **Group Replication:** MySQL Group Replication can now communicate using a dedicated input channel as an alternative to using a TCP socket. The new input channel uses shared memory for communication between the Group Replication logic and the local instance of the underlying group communication engine (XCom, a Paxos variant).

  Previously, communication with the local XCom instance always took place using a TCP socket, namely the network address that is specified by the `group_replication_local_address` system variable for each group member. This incurred overheads that were unnecessary for local communication, such as memory copying through the network protocol stack and data serialization. A TCP socket (`group_replication_local_address`) is still required for each group member to communicate with remote XCom instances. The Group Communication System (GCS) component of Group Replication now selects the most appropriate communication method for each Group Replication task, either the input channel or TCP. For example, the process of joining a group requires communication with a remote XCom instance, so TCP must be used. However, the process of removing a member from a group only requires communication with the local XCom instance, so the input channel is used. The input channel is selected wherever possible to minimize the overheads associated with communication using networking mechanisms. (WL #9850)

- **Microsoft Windows:** The access control granted to clients on the named pipe created by the MySQL server now is set to the minimum necessary for successful communication on Windows. Newer MySQL

client software can open named pipe connections without any additional configuration. If older client software cannot be upgraded immediately, the new `named_pipe_full_access_group` server system variable can be used to give a Windows group the necessary permissions to open a named pipe connection. Membership in the full-access group should be restricted and temporary. (WL #12445)

- The minimal server RPM is mostly used for Docker images. For better Docker compatibility, the `log-error` line has been removed from the `rpm-docker` configuration file. This way, logging goes to `stdout`/`stderr`, enabling use of Docker's own interface. (Bug #28692675)

- Error messages relating to creating and dropping foreign keys were improved to be more specific and informative. (Bug #28526309, Bug #92087)

- The error message for `ALTER TABLE` statements that attempted character set conversion but failed was improved to indicate which column produced the error. (Bug #27546306, Bug #88738)

- Previously, for command options that take a numeric value, the value could be given with a suffix of `K`, `M`, or `G` to indicate a multiplier of 1024, $1024^2$ or $1024^3$. Now a suffix can also be `T`, `P`, and `E` to indicate a multiplier of $1024^4$, $1024^5$ or $1024^6$. Thanks to Daniel Black for the patch. (Bug #27306931, Bug #89017)

- Resource group locking was revised to improve scalability and performance. (Bug #27148580)

- The startup option `--binlog-row-event-max-size` now has a corresponding system variable `binlog_row_event_max_size`. The startup option and system variable set a soft limit on the maximum size of a row-based binary log event, with a default setting of 8192 bytes. Where possible, rows stored in the binary log are grouped into events with a size not exceeding the value of this setting. If an event cannot be split, the maximum size can be exceeded.

  The `binlog_row_event_max_size` global system variable is read-only and can be set only at server startup. Its value can therefore only be modified by using the `PERSIST_ONLY` keyword or the `@@persist_only` qualifier with the `SET` statement. The addition of a system variable means that this setting can be viewed using the Performance Schema tables or a `SHOW VARIABLES` or `SELECT` statement. (Bug #19985377, Bug #74728, WL #12385)

- `ALTER TABLE` now can be used to change a column character set in place (without a table rebuild), when these conditions apply:

  - The column data type is `CHAR`, `VARCHAR`, a `TEXT` type, or `ENUM`.
  - The character set change is from `utf8mb3` to `utf8mb4`, or any character set to `binary`.
  - There is no index on the column.

  (WL #11605)

- The new `-DFORCE_INSOURCE_BUILD` `CMake` option defines whether to force an in-source build. Out-of-source builds are recommended, as they permit multiple builds from the same source, and cleanup can be performed quickly by removing the build directory. To force an in-source build, invoke `CMake` with `-DFORCE_INSOURCE_BUILD=ON`.

## Bugs Fixed

- **Important Change:** Importing a dump from a MySQL 5.7 server to a server running MySQL 8.0 often failed with `ER_WRONG_VALUE_FOR_VAR` when an SQL mode not supported by the 8.0 server was used. This could happen frequently due to the fact that `NO_AUTO_CREATE_USER` is enabled by default in MySQL 5.7 but not supported in MySQL 8.0.

The behavior of the server in such circumstances now depends on the setting of the `pseudo_slave_mode` system variable. If this is false, the server rejects the mode setting with `ER_UNSUPPORTED_SQL_MODE`. If `pseudo_slave_mode` is true, the server ignores the unsupported mode and gives a warning. Note that `mysqlbinlog` sets `pseudo_slave_mode` to true prior to executing any SQL. (Bug #90337, Bug #27828236)

- **InnoDB:** Global and backup metadata locks were not released after the background purge thread truncated undo logs. (Bug #29215254, Bug #93901)
- **InnoDB:** MySQL would not start on Solaris X86. The static thread-local 'tables' variable in the TempTable storage engine was not properly initialized. (Bug #28987365)
- **InnoDB:** Latching logic used during deadlock detection was simplified. (Bug #28904966)
- **InnoDB:** An invalid record offset for an old version of a clustered index record raised a debug assertion. (Bug #28825617)

  References: This issue is a regression of: Bug #25540277.
- **InnoDB:** The minimum DML delay imposed when the length of the history list exceeds `innodb_max_purge_lag` was decreased from 5000 microseconds to 5 microseconds. (Bug #28813453)
- **InnoDB:** An incorrect lock order caused a deadlock when one thread attempted to drop a table while another created an encrypted tablespace. (Bug #28774259)
- **InnoDB:** `ALTER TABLESPACE` failed to ignore unsupported tablespace attributes. (Bug #28656611)
- **InnoDB:** Implicit to explicit lock conversion logic was simplified and optimized. (Bug #28637472)
- **InnoDB:** A fragment page allocation failure raised an assertion. (Bug #28615893)
- **InnoDB:** Incorrectly placed debug points caused flushed LOB pages to be considered corrupt. (Bug #28607368)
- **InnoDB:** The `TempTable` storage engine incorrectly created temporary files in the system temporary directory instead of the directory defined by the `tmpdir` variable. (Bug #28598943)
- **InnoDB:** Attempting to drop a table with a name similar to that of a full-text search auxiliary table caused an assertion failure. (Bug #28577083)
- **InnoDB:** A function called by an `UPDATE` query did not account for virtual columns. (Bug #28560650)
- **InnoDB:** An incorrect key was defined for the buffer pool zip hash mutex. (Bug #28556539)
- **InnoDB:** Deadlock handling for background transactions that involve the `mysql.innodb_table_stats` and `mysql.innodb_index_stats` tables was modified. The tables were incorrectly included in an assertion that is triggered when internal tables are included in a deadlock cycle. (Bug #28523042, Bug #92069)
- **InnoDB:** Setting `innodb_spin_wait_delay` to a high value caused an assertion failure when attempting to shut down the server. To prevent this failure from occurring, the `innodb_spin_wait_delay` maximum value was reduced to 1000. (Bug #28489407, Bug #91973)
- **InnoDB:** An `ON DELETE CASCADE` operation on table with a foreign key constraint and an indexed virtual column caused the server to exit. (Bug #28470805)
- **InnoDB:** An incorrectly written DML log involving a virtual column value raised an assertion. (Bug #28448853)

- **InnoDB:** A `RENAME TABLE` operation failed when run on a table created outside of the MySQL data directory using the `DATA DIRECTORY` clause. (Bug #28341514)

- **InnoDB:** `ALTER TABLE ... EXCHANGE PARTITION` permitted partitions with different virtual column definitions to be exchanged, which resulted in an assertion when `InnoDB` later attempted to read from a nonexistent virtual column. (Bug #28235668)

- **InnoDB:** A counter was added for redo log write and flush requests that occur during transaction commit. The counter is used by the log writer thread to compute the average time between consecutive requests. When the average time is greater than 100 microseconds, log writer threads do not use spin delay and instead wait on request events with a 10 microsecond timeout limit.

  A log writer thread implementation issue that could cause a hang was also fixed. (Bug #28062382, Bug #28444247, Bug #28616442, Bug #90890)

- **InnoDB:** An assertion was raised when attempting to add rollback segments to newly added undo tablespace that was not fully initialized. (Bug #27914054)

- **InnoDB:** Foreign key constraints were ignored after a `RENAME TABLE` operation. (Bug #27453180, Bug #89441)

- **InnoDB:** Using the `O_DIRECT_NO_FSYNC` `innodb_flush_method` setting could cause the system to hang due to file system metadata becoming unsynchronized. To prevent this issue from occurring in `O_DIRECT_NO_FSYNC` mode, `InnoDB` now calls `fsync()` after creating a new file, after increasing file size, and after closing a file. The `fsync()` system call is still skipped after each write operation. (Bug #27309336)

- **InnoDB:** Specifying the `CREATE TABLE` or `ALTER TABLE` `ENCRYPTION` option with an empty string failed to raise an error and was interpreted as a default setting, which is `ENCRYPTION='N'`. Specifying an empty string is now treated as invalid and raises an error. (Bug #27177845)

- **InnoDB:** Partitioned table name delimiters (the `#P#` or `#SP#` part of a partitioned table name) were not converted to lowercase when moving tablespace data files from a MySQL instance on Windows to a MySQL instance on Linux where the `lower_case_table_names` variable was enabled. Failure to fully convert names to lowercase caused errors such as “'InnoDB error' from storage engine” when attempting to alter, rename, or optimize the tables. (Bug #26925260)

- **InnoDB:** An assertion was raised when attempting to write to a tablespace file greater than 4GB in size on a 64-bit Windows system. The failure was due to a narrowing cast. (Bug #26636815, Bug #87423)

- **InnoDB:** After attempting to create a table with a foreign key constraint that referenced a partitioned table, which is an unsupported operation, `SHOW ENGINE INNODB STATUS` output incorrectly reported a foreign key error indicating that the referenced table name could not be resolved. This error no longer appears, and the error message returned to the client now states that foreign keys are not yet supported in conjunction with partitioning. (Bug #25319071, Bug #84331)

- **InnoDB:** Misleading error messages were reported for unsupported foreign key operations, including creating a foreign key that referenced a partitioned table, and referencing a table that uses a storage engine that does not support foreign keys. The error messages are now more informative. (Bug #11747571, Bug #33027)

- **Partitioning:** Trying to perform an instant add column on a discarded tablespace led to an assert. An error is now returned in such cases. (Bug #28517843)

- **Partitioning:** Repeated `ALTER TABLE` statements on partitioned tables containing `BLOB` or `TEXT` columns were not always handled correctly. (Bug #28491099)

- **Partitioning:** `ALTER TABLE ... EXCHANGE PARTITION` did not work when the partitioned table had one or more partition definitions using the `DATA DIRECTORY` option. This fix supports partitioned tables using the `InnoDB` storage engine only. (Bug #19730200)

- **Replication:** Depending on the value of `group_replication_exit_state_action`, the behavior of members exiting a group was not consistent. To harmonize the behavior of members exiting the group regardless of the error scenario, now when a member with `group_replication_exit_state_action=READ_ONLY` exits the group unintentionally, the `super_read_only` mode that the member had when started is restored. This makes the behavior consistent with that of a member with `group_replication_exit_state_action=ABORT_SERVER`. (Bug #28971639, Bug #28526591)

- **Replication:** The metadata written to the binary log for `CREATE TABLE` statements includes character set information for the character columns in the table. Previously, when the `mysqlbinlog` option `--print-table-metadata` was specified, a default character set was printed for the table. This default character set was the character set that appeared most frequently in the table columns, and might not match the default character set that had been specified for the table. `mysqlbinlog` now prints the character set for each column individually. The columns are also printed on separate lines. (Bug #28774144)

- **Replication:** Character set information was not written to the binary log as part of the table metadata for `ENUM` and `SET` columns. This information is now added when `binlog_row_metadata=FULL` is set, which produces extended metadata. (For character columns, character set information is also added with `binlog_row_metadata=MINIMAL`.) (Bug #28706307)

- **Replication:** A patch to correct the handling of quotes for identifiers in `ROLLBACK TO SAVEPOINT` statements in the binary log was not correctly applied to subsequent MySQL versions. (Bug #28569645)

- **Replication:** Following a patch in MySQL 5.7.23, `LOAD DATA` statements stopped statement-based replication from a MySQL 5.7.22 master to a replication slave at a later release. The problem has now been fixed. (Bug #28541204, Bug #92132)

- **Replication:** In some circumstances, the `CHANGE MASTER TO` statement could not be used on a replication slave if the master info log had been changed from a table (`master_info_repository=TABLE`) into a file (`master_info_repository=FILE`). (Bug #28529558)

- **Replication:** `mysqlbinlog` incorrectly added statements to set the `sql_require_primary_key` system variable (which was introduced in MySQL 8.0.13) to `ON` for events involving DML SQL statements. The check carried out when the system variable is set to `ON` is only relevant for DDL SQL statements that create new tables or alter the structure of existing tables. (Bug #28524803)

- **Replication:** When the system variables `binlog_transaction_dependency_tracking` and `binlog_transaction_dependency_history_size` were set or read, the types of lock that were required could result in a deadlock scenario, because the same locks were also required for working with the active binary logs. A new lock type is now used instead for access to the transaction dependency tracking system variables, so that this deadlock cannot occur. (Bug #28511326, Bug #91941, Bug #28537209, Bug #92108)

- **Replication:** An assertion was raised in debug builds if an implicit commit was attempted when the GTID value for the next transaction had not yet been determined (`gtid_next=NOT_YET_DETERMINED`). The `gtid_next` system variable has this value immediately after the internal-use statement `BINLOG` has been issued by `mysqlbinlog` to execute a format description event. If a statement with an implicit commit was attempted next (such as a `CREATE TABLE` statement), the `gtid_next` setting did not transition to `AUTOMATIC` state, and was left in an unacceptable state. If autocommit was on, the error

ER_CANT_SET_GTID_NEXT_TO_ANONYMOUS_WHEN_GTID_MODE_IS_ON was also logged when the statement was attempted.

To fix this issue, the use of the BINLOG statement is now prevented during transactions if it would change the state of gtid_next. The error ER_VARIABLE_NOT_SETTABLE_IN_TRANSACTION is returned if this is attempted. Also, when GTIDs are in use and the value of gtid_next is NOT_YET_DETERMINED, the next statement must either explicitly set gtid_next to a valid value or leave the GTID state unaffected. Otherwise the error ER_CANT_SET_GTID_NEXT_TO_ANONYMOUS_WHEN_GTID_MODE_IS_ON is returned. (Bug #28490793, Bug #91980)

- **Replication:** The PURGE BINARY LOGS TO *'log_name'* statement failed for binary log files that had been moved to another location using mysqlbinlogmove. Such files are still listed in the binary log index file, but they are listed using an absolute path, rather than a path relative to the directory where the binary log files are normally stored. MySQL Server can now locate and purge moved binary log files successfully. (Bug #28284624)

- **Replication:** When binlog_format is set to MIXED, if a function contained DML statements that applied to a temporary table, and also a DROP TEMPORARY TABLE statement, the function call was not written to the binary log, which caused replication errors. The function call is now written to the binary log in mixed replication mode if the function contains DML statements that operate on a temporary table. (Bug #28258992)

- **Replication:** If autocommit was set to 0 for a replication slave or Group Replication group member where GTIDs were in use and super_read_only=ON was set, server shutdown was prevented by a transaction that did not complete. The transaction was attempting to save GTIDs to the mysql.gtid_executed table, but the update failed because super_read_only=ON was set. (With autocommit set to 1, the transaction would complete in this situation, and the mysql.gtid_executed table would instead be updated at server startup.) Now, the check for the super_read_only setting is skipped for this task, so the transaction is able to save the GTIDs to the mysql.gtid_executed table and complete regardless of the combination of super_read_only and autocommit settings. (Bug #28183718)

- **Replication:** An assertion was raised in debug builds if an XA ROLLBACK statement was issued for an unknown transaction identifier when the gtid_next value had been set manually. The server now does not attempt to update the GTID state if an XA ROLLBACK statement fails with an error. (Bug #27928837, Bug #90640)

- **Replication:** An assertion was raised in debug builds if a SELECT... FOR UPDATE statement was issued immediately after a transaction was committed or rolled back, and the transaction had been assigned a GTID manually using the gtid_next session system variable. After gtid_next has been used to set a GTID for a transaction, and the transaction has been committed or rolled back, another explicit SET GTID_NEXT statement must be issued before any other statement, otherwise the gtid_next value is left undefined. The SELECT... FOR UPDATE statement caused a GTID consistency violation in this situation because it acquired write locks, although it did not make any changes. SELECT... FOR UPDATE statements that acquire write locks now return an error in this situation. (Bug #27903848, Bug #90547)

- **Replication:** Under heavy loads, a race condition in binary log group commit could cause the server to stop unexpectedly. The tracking of transaction commits has been changed to prevent this situation. (Bug #27556117)

- **Replication:** The value returned by a SHOW SLAVE STATUS statement for the total combined size of all existing relay log files (Relay_Log_Space) could become much larger than the actual disk space used by the relay log files. The I/O thread did not lock the variable while it updated the value, so the SQL thread could automatically delete a relay log file and write a reduced value before the I/O thread finished

updating the value. The I/O thread then wrote its original size calculation, ignoring the SQL thread's update and so adding back the space for the deleted file. The `Relay_Log_Space` value is now locked during updates to prevent concurrent updates and ensure an accurate calculation. (Bug #26997096, Bug #87832)

- **Replication:** If the relay log index file was temporarily locked for viewing by a backup process for a replication slave, and MySQL Server also attempted to access the file at that time for rename or delete operations, the backup completed with warnings, but MySQL Server experienced an unexpected halt. MySQL Server now retries the file access operation a number of times in case this or a similar scenario is the explanation and the file becomes available again before long. (Bug #25839610)

- **Replication:** With `sync_binlog=1` set, if the binary log was rotated during a commit before the binary log end position was updated, replication stopped on the slave because the server attempted to use the old binary log end position with the new binary log file. The server now compares the binary log file name with the active binary log file when updating the binary log end position, so that the issue does not occur. (Bug #22252394, Bug #25524203, Bug #84752)

- **Replication:** When adding a new member to a group, if the certification information was too big to transmit, an event was generated that caused failures in all group members. To avoid this situation, now if the certification information is too large an error is generated which makes the joining member leave the group. (Bug #93130, Bug #91870, Bug #28900691, Bug #28443958)

- **Replication:** When you used `group_replication_switch_to_single_primary_mode()`, if a member which also had an asynchronous channel encountered an error, the asynchronous replication channel was not stopped correctly, and the server could stop unexpectedly. (Bug #91747, Bug #28382590)

- **Replication:** It was possible to use the group coordinator based functions which configure a group, such as `group_replication_switch_to_single_primary_mode`, while members were in the `UNREACHABLE` or `RECOVERING` state, and this caused the operation to wait until all members became `ONLINE`. This could result in the group coordinator operation never completing successfully. Now, if you invoke any of these functions on a group in this state, an error is returned. Ensure all members are `ONLINE` before attempting to configure the group using the functions. (Bug #91537, Bug #28284355)

- **Replication:** When a member joined a group that had a constant peak load, the member might not be able to move from the `RECOVERING` to the `ONLINE` state. The cause was that:

  - the member was waiting in a loop for the complete queue of transactions that arrived during recovery to be applied, while new transactions were still arriving.

  - even when the complete queue had been applied, the member was also checking that the applier was paused, which is unlikely to happen in a continuous peak workload.

  Now, when the recovery completion policy is waiting for transactions to be applied, the member first waits until one of the following conditions is fulfilled:

  - the transactions to apply fit within the flow control configuration. In other words, the transactions to be applied can be applied during the next flow control iteration;

  - no transactions are being queued or applied, in the case of an empty recovery queue.

  Then, the member waits for the currently queued transactions in the `group_replication_applier` channel to be applied, before the member state changes to `ONLINE`. (Bug #89582, Bug #27511404)

- **Group Replication:** The maximum timeout setting for the waiting period before expelling a suspect Group Replication group member has been reduced to 3600 seconds (one hour). Previously, the `group_replication_member_expel_timeout` system variable could be set to a value of up to

31536000 seconds. The new upper limit provides a more reasonable maximum for the removal of inactive members from the group. The default setting for the timeout is zero, meaning that inactive members are liable for expulsion immediately after the 5-second detection period ends. Specifying a timeout value is useful to avoid unnecessary expulsions on slower networks, or in the case of expected transient network failures or machine slowdowns. (Bug #28656750)

- **Group Replication:** On systems where the Group Communication System (GCS) used the `systemd-resolved` service for network name resolution, if the host name could not be resolved, GCS kept trying indefinitely. Now, if a retry message is returned from any name resolution service, GCS makes a limited number of retries, then concludes that the host name is unresolvable. (Bug #28177861)
- **Group Replication:** When a group was being reconfigured online, for example using `group_replication_switch_to_multi_primary_mode` or `group_replication_set_as_primary`, there was a chance that stopping a member could result in an unexpected stop. Now, when you issue `STOP GROUP_REPLICATION`, if the member is part of an online group that is being reconfigured, the group coordinator is informed that Group Replication is stopping. The member waits for the online configuration process to complete any ongoing actions, but any subsequent actions are cancelled. (Bug #92829, Bug #28807260)
- **Group Replication:** When stopping group replication, any channels with pending transactions could cause a deadlock. (Bug #92376, Bug #28636768, Bug #28365855)
- **Group Replication:** When `group_replication_exit_state_action` is set to `ABORT_SERVER`, the Group Replication plugin now uses the new component service added by WL#12003 to shut down MySQL. (Bug #91793, Bug #28401703)
- **Microsoft Windows:** MySQL Installer could fail after failure to remove an existing MySQL service. This is now treated as nonfatal so that installation operations can continue, but might require a system restart to permit service cleanup. (Bug #29016677, Bug #93048)
- **Microsoft Windows:** When multiple instances of `mysqld` were started with the `--no-monitor` option on the same host for same user, the `SHUTDOWN` command shut down the wrong server process. This fix creates a unique shutdown event name for use with `--no-monitor` by appending the process ID of the process. (Bug #28723675)
- **X DevAPI:** When using the X Protocol, a stored procedure called with a user variable as an `OUT` parameter did not set the variable's value. (Bug #91907, Bug #28458752)
- **JSON:** Iteration over JSON objects resulted in unnecessary allocation of strings. (Bug #28975640)
- **JSON:** Conversion of `JSON` values to text caused linear growth of the destination string, resulting in an unnecessarily high number of reallocations. Now this process uses exponential growth instead, to reduce the number of allocations required. (Bug #28949700)

  References: See also: Bug #103790, Bug #32919524.
- **JSON:** `YEAR` values were stored as opaque data in JSON; when JSON documents containing `YEAR` values were converted to text, the YEAR values were shown as base64-encoded strings. To resolve this issue, `YEAR` values are now stored as unsigned integers, which are shown as numbers when converted to text. An additional benefit of this fix is that less storage space is now required for `YEAR` values within JSON documents. (Bug #28947107)
- **JSON:** Hit an assert when attempting to execute `UPDATE` or `DELETE` on an `ARCHIVE` table containing a `JSON` column. (Bug #28923281)
- **JSON:** When trying to select from a `JSON` column of a `FEDERATED` table, the server returned `ER_INVALID_JSON_PATH_CHARSET` `Cannot create a JSON value from a string with CHARACTER SET 'binary'`.

In addition, neither `DELETE` or `UPDATE` had any effect on a `FEDERATED` table containing a `JSON` column. (Bug #28877215)

- **JSON:** A query of the form `SELECT jt.* FROM t1, JSON_TABLE(t1.c, '$[*]' COLUMNS (num INT PATH '$[0]')) AS jt` failed due a permissions error even though the user executing the query had the `SELECT` privilege on column `c`. (Bug #23254268)
- The code contributed by Facebook for the feature implemented by Bug#27855592 was updated. (Bug #28950397)

  References: See also: Bug #27855592.
- On SuSE Linux, spurious `EBUSY` return values from `pthread_mutex_destroy()` were not handled. (Bug #28948462)
- `mysqld_safe` and `mysqld_multi` were incorrectly included in client-only packages. (Bug #28942508)
- Mishandling of host cache locking could cause a server exit. (Bug #28936159)
- MySQL Enterprise Firewall did not work well if the `audit_log` plugin was installed. (Bug #28930885, Bug #93184)
- Corrections were made to enable successful builds under Visual Studio on Windows. (Bug #28892711, Bug #93077)
- The server permitted creation of databases with the same name as redo log files, which could result in unexpected server behavior. Such names are no longer permitted as database names. (Bug #28867993)
- `mysqld_multi` could fail to pass the correct `datadir` value to `mysqld`. (Bug #28866662, Bug #90801)
- A debug assertion that checks parameter schema names during MDL key creation for routines, events, and triggers to ensure that names are lowercase failed when encountering a schema name that included multi-byte character. (Bug #28864244)
- The format specifiers for some error messages were improved to avoid displaying incorrect numeric values. (Bug #28860795)
- For debug builds on Windows, unused memory leak checks were enabled and could slow down the shutdown process. These checks are now enabled only for specialized builds. (Bug #28857626)
- `mysql_upgrade` could fail to upgrade certain system tables if the `sql_require_primary_key` system variable was enabled. (Bug #28855207, Bug #92988)
- Builds configured with `-DWITH_LIBWRAP=ON` did not compile. (Bug #28853650, Bug #92983)
- For `InnoDB` tables, the values of stored or indexed virtual generated columns that depended on the `DEFAULT()` function were not correctly updated by `ALTER TABLE`, if the default for a column referenced in this function was changed by making column nullable. (Bug #28848265)
- Corrections were made to enable successful builds under Visual Studio on Windows with the `/permissive` flag turned on. (Bug #28842878, Bug #92943)
- Builds configured with `-DCMAKE_BUILD_TYPE=Release` did not compile. (Bug #28841366, Bug #92945)
- `ALTER TABLE` now can use the `INPLACE` algorithm when these conditions apply:
  - For `InnoDB` tables, statements that modify generated stored columns but do not change their type, expression, or nullability.

  - For non-InnoDB tables, statements that modify generated stored or virtual columns but do not change their type, expression, or nullability.

  An example of such a change is a change to the column comment. (Bug #28836543)

- Plugin system variables that had been persisted were not applied when the plugin was reinstalled. (Bug #28823972)

- EXPLAIN ... FOR CONNECTION could modify another connection's SQL mode. (Bug #28786981)

- Removal of Sun RPC and XDR from glibc into a separate libtirpc library caused problems with libasan on some platforms. (Bug #28785835, Bug #92762, Bug #28897799, Bug #93116)

- It was possible to hit an assert when comparing two ENUM values while processing a query of the form SELECT a FROM *table* WHERE b = *value* and there was an index on column b. (Bug #28769996)

- Concurrent read and write access to the offline_mode system variable could result in deadlock. (Bug #28761869)

- Triggers were loaded into memory in an incorrect order when upgrading from MySQL 5.7 to MySQL 8.0, causing an assertion failure. (Bug #28760011, Bug #92609)

- Joins involving the Performance Schema data_locks table could produce incorrect results. (Bug #28733170)

- Some multiply-nested subqueries involving the use of scalar subqueries were not handled correctly. (Bug #28723670)

- On Ubuntu, the installed /etc/mysql/mysql.conf.d/default-auth-override.cnf file was mistakenly created with executable mode. Thanks to Evgeniy Patlan for the correction contribution. (Bug #28714840, Bug #92587)

- A memory leak was caused by GET_LOCK() calls with a zero timeout that failed due to concurrent connections holding the same user-level lock. (Bug #28714367)

- Heap corruption and a server exit could occur when a server hosting a large number of tables was started and stopped repeatedly. (Bug #28705511, Bug #92572)

- MySQL Router was missing from MySQL Server MSI packages. (Bug #28685556)

- The example stored function GTID_SUBTRACT_UUID has been corrected in the code to match the documented version. (Bug #28670170)

- CAP_SYS_NICE capability is no longer enabled for mysqld by MySQL package installers for Linux. (This was done to facilitate use of resource groups thread priorities.) For Linux deployments that require access to thread priorities, consult the MySQL Reference Manual instructions for enabling CAP_SYS_NICE capability at Resource Group Restrictions. (Bug #28670160)

- The internal implementation of the <=> operator was simplified. (Bug #28660232)

- After a STOP GROUP_REPLICATION statement was issued to remove a server instance from a group, multiple instances of the error message "[GCS] Error pushing message into group communication engine" were logged on the server instance. The error is now ignored when a server is in the process of leaving a group or is no longer a member of a group. (Bug #28658228, Bug #92454)

- If the CHARACTER SET attribute of some column was implicit in JSON_TABLE(... COLUMNS ...), the resulting column used the global character_set_results as the default character set. The column

now uses the session `character_set_connection` and `collation_connection` values. (Bug #28643862)

- Adding a functional index on an expression that produced a row value raised an assertion; now it results in an error instead. (Bug #28643252)
- In debug builds, creating a trigger after setting `sql-mode` to `TIME_TRUNCATE_FRACTIONAL` caused an assertion failure. The SQL mode was not present in the `sql_mode` column of the `mysql.triggers` data dictionary table. (Bug #28642918)
- When using `--log-timestamps=SYSTEM`, ISO 8601 timestamps in log messages did not take account of daylight saving time. (Bug #28632725, Bug #32893161)
- The arguments for error `ER_IB_MSG_720` were computed incorrectly. (Bug #28629175)
- The server could exit at startup if the option for specifying a socket file was not specified correctly. (Bug #28609181)
- It was possible to create an inconsistent foreign key by adding a parent table with a different storage engine from the child table, then changing the parent table to the same storage engine as the child table. (Bug #28608460, Bug #92317)
- When a server is joining a replication group, it attempts to connect to the first seed member listed in its `group_replication_group_seeds` system variable. If the connection is refused, the joining member tries to connect to each of the other seed members in the list in order. Previously, If the joining member connected to a seed member but did not get added to the replication group as a result, the joining member did not make any further connection attempts. This situation could occur if the seed member failed after the connection was made, or if the seed member did not have the joining member's address in its whitelist and closed the connection, or if the seed member rejected the joining member's request to join the group. Now, if the joining member connects to a seed member but does not manage to join the group, the joining member continues to try the remaining seed members in the list in order. (Bug #28602835)
- Given certain patterns of allocations, copies with rebinds of the allocator, and deallocations, it was possible for `temptable::Allocator` to reuse a freed memory block. This led to failures in the test suite on Windows platforms. (Bug #28595557)
- Setting `time_zone` to a negative offset and `timestamp` to a low value triggered an assertion when altering routines and views. (Bug #28590623, Bug #92273)
- Persisting the `pid_file` system variable to `DEFAULT` could result in a value of `NULL` for subsequent server startups. (Bug #28589736)
- Incorrect privilege checking could produce an error for `SELECT ... FOR UPDATE` statements that executed successfully in MySQL 5.7. (Bug #28581664, Bug #92254)
- Attempting to rename the parent column of a foreign key with `ALTER TABLE` could fail. (Bug #28581468)
- Privileges for `RESET PERSIST` were not checked correctly. (Bug #28564239)
- An overflow occurred when calculating `AVG(YEAR(datetime_column))`. (Bug #28562930)
- After a server restart, path names of persisted system variables in the Performance Schema `variables_info` table could be calculated incorrectly. (Bug #28561584)
- A partitioned table name check raised an invalid assertion. (Bug #28556942)
- The `handler::create()` function could be called with an error in the condition list, which could prevent an error in the `handler::create()` function from being reported properly. (Bug #28556264)

- For `ALTER TABLE`, `ALGORITHM=INSTANT` was incorrectly rejected on tables created in a MySQL version prior to 8.0.12. (Bug #28554157, Bug #92194)
- mysqlpump did not free all allocated resources when it encountered an error, resulting in memory leaks. (Bug #28538971, Bug #92131)
- The `COLLATE` attribute was rejected for data types in the `COLUMNS` clause of the `JSON_TABLE()` function. (Bug #28538315)
- For debug builds, the server could exit when attempting to roll back `CREATE USER` statements. (Bug #28536312)
- Plugin variables with signed values were displayed incorrectly. (Bug #28534414, Bug #92107)
- Mishandling of deprecated system variables could cause output from queries on the Performance Schema `variables_by_thread` table to be incorrect. (Bug #28515475, Bug #92049)
- Data races discovered by Thread Sanitizer in `Event_queue::lock_data` and the `SAFE_MUTEX` implementation were fixed. (Bug #28510721, Bug #92041, Bug #28510691, Bug #92040)
- No `ER_NEED_REPREPARE` diagnostic was pushed to the diagnostics area when a reprepare failed for prepared statements. (Bug #28509306, Bug #92029)
- When a subquery contained a `UNION`, the count of the number of subquery columns was calculated incorrectly. (Bug #28499924)
- When evaluating an expression using `WITH ROLLUP`, we now write the result of the expression into a temporary table only when it has a temporary table column. (Bug #28493849, Bug #28523014)
- For debug builds, incorrect foreign key error checking for `ALTER TABLE` on a `TEMPORARY` table could result in a server exit. (Bug #28493257, Bug #91990)
- For some system variables, `SET PERSIST` persisted the default value rather than the specified value. (Bug #28466045)
- `SET RESOURCE GROUP` could not be executed as a prepared statement. (Bug #28448258, Bug #91876)
- Restored a call to `Item_field::fix_fields()` that was inadvertently removed during work done to implement window functions. (Bug #28431783)
- Data races reported by Thread Sanitizer during X Plugin startup and shutdown were corrected. (Bug #28407294)
- Creating a table with a partition description containing illegal utf8 characters raised an assertion. (Bug #28387488, Bug #91763)
- `mysqldump` output could include SQL mode values that have been removed. (Bug #28373001, Bug #91714)
- A potential lock order cycle was corrected. (Bug #28366531)
- On a GTID-enabled server, concurrent statements on the `INFORMATION_SCHEMA.COLUMNS` table could deadlock. (Bug #28293047, Bug #91548)
- `CREATE TABLE` statements for tables with the `utf32` table character set and literal strings in the table definition raised an assertion. (Bug #28275881)
- Internal functions were added to support updating the server version number upon the successful completion of a server upgrade. (Bug #28211486, Bug #91323)

- Comparing log file names as strings using the `memcmp()` function resulted in uninitialized memory read errors. The comparison now uses the `strncmp()` function. Thanks to Zsolt Parragi and Laurynas Biveinis for their contributions. (Bug #28178776, Bug #90238)
- The server mishandled stored program and resource group names that differed only in accents. (Bug #28122841)
- The optimizer skipped the second column in a composite index when executing an inner join with a `LIKE` clause against the second column. (Bug #28086754)
- `CREATE TABLE ... SELECT` could create date columns with “zero” date default values when it should have created them without a default value. (Bug #28022129)
- The transformation of `IN` subquery predicates into semijoins was not handled correctly for a very large number of tables. (Bug #28004674)
- The bitmap used for reading fields from the storage engine was not enabled correctly when performing the filesort which is added to the last of any temporary tables created for window functions. In the case where there was no need for a temporary table, the server added a filesort to the output from the select table, but the removed reference (the `WHERE` condition) was not added. Now in such cases, the reference is added to the select table when the first window function needs sorting and no temporary table was created before processing this window function. (Bug #27975193)
- After seeing a row in the range frame, if another row was later determined to be appearing before this range frame, the server continued to check for new rows. This led to the next frame calculation being done incorrectly. (Bug #27973860)
- Server mishandling of `SIGHUP` signals could result in a server exit. (Bug #27966483, Bug #90742)
- `DELETE WHERE a=constant` from a table with column `a` and partitioned by the value of a generated column `b` led to an assertion in debug builds. (Bug #27954073)
- `INFORMATION_SCHEMA` queries could cause a server exit when updating dynamic table statistics. (Bug #27898108)
- A metadata locking deadlock could occur when opening a foreign key parent table. (Bug #27859086)
- Improper memory handling by account management statements could result in server misbehavior. (Bug #27820277)
- In certain cases, window functions did not handle `ORDER BY` and `PARTITION BY` correctly. (Bug #27816506)
- The MySQL query optimizer identifies each predicate to be pushed down to a table as a table condition; as part of this process it checks to see whether a given predicate among the table conditions is already known to be true in virtue of the selected access path for the table, in which case the predicate can safely be eliminated.

  For example, when executing `SELECT * FROM t1 WHERE pk=1`, where `pk` is the primary key of table `t1`, the `ref` access method is selected. Since we know that this already returns only rows for which `pk=1`, further evaluation of this condition as a filter (`Using where`) should be eliminated.

  When optimizing a query which includes a `GROUP BY` or `ORDER BY`, a late optimizer check is performed to discover whether sorting can be skiped by using a sorted index instead. Since this is done after another index may have been chosen for accessing the table to be sorted, some predicates thought to be redundant (due to the previous access path selected) could be removed prematurely. To compensate for this, the following actions were performed:

- Reconstruction of a table condition containing predicates previously eliminated due to the access method already being selected.
- Performing a check to see whether any sorted index existed such that sorting could be avoided, possibly modifying the access plan.

Issues arose because the following actions also intended to remedy early predicate removal were not performed correctly:

- Whether the access plan was modified or not, any extra predicates added back in the reconstructed table condition mentioned previously became a permanent part of the table condition.
- When the access plan was changed to use another sorted index, no analysis was performed for the new index in order to remove predicates made obsolete by the new index.

A further problem existed for storage engines implementing condition pushdown, such as `NDBCLUSTER`: The conditions pushed down were generated from the table condition prior to analysis, such that, if the access path was later changed, the pushed condition did not contain the predicates already removed, making condition pushdown less efficient.

The root cause of this issue was that `part_of_refkey()` analysis was performed on table predicates before the access method for the table had been completely decided. This is fixed by removing such early analysis. (Bug #27808758, Bug #27814026)

- A windowing function that included an `ORDER BY` *`column`* clause failed with `Unknown field in window order by` even when the column was found in the table being queried. (Bug #27808099)
- Executing a prepared statement to do a multiple-row insert with large number of placeholders consumed excessive memory and could execute slowly. (Bug #27703912)
- On Windows, if the Visual C++ Redistributable for Visual Studio had been removed, MySQL uninstallation using the MSI installer failed. (Bug #27621546)
- The parser accepted invalid `SET` statement syntax in trigger definitions that could result in a server exit. (Bug #27595603)
- The server failed to start if the `keyring_encrypted_file` plugin keyring file was invalid. (Bug #27588064)
- Keyring migration failed with source and destination keyring plugins of `keyring_okv` and `keyring_encrypted_file`, respectively. (Bug #27493970)
- It was possible in debug builds for a windowing function using a signed integer to mishandle a frame that included `FOLLOWING`. (Bug #27452365)
- When executing a prepared statement with a procedure call with the `CURSOR_TYPE_READ_ONLY` flag set, the client library hung if the procedure performed a `SELECT` that returned an empty result set. (Bug #27443252, Bug #89214)
- Names of referenced columns of foreign keys were always shown in lowercase in `SHOW CREATE TABLE` output and the `INFORMATION_SCHEMA.KEY_COLUMN_USAGE` table. (Bug #27353767, Bug #88718)
- Loading and unloading the `audit_log` plugin while performing other concurrent activity could cause the server to become unresponsive. (Bug #27325622)

- The data dictionary properties interface (`dd::Properties`) and implementation was revised to provide a new method of defining valid keys for property objects. (Bug #27309072, Bug #89031, Bug #27309082, Bug #89032)
- Installing and uninstalling the `validate_password` component concurrently with `SET PASSWORD` could cause component failure. (Bug #27020979)
- Some typos in server source code were fixed. Thanks to Hyunwoo Park for the contribution. (Bug #26189673, Bug #86565)
- After column privileges were granted to a table, a `HANDLER READ` call asserted during privilege checking. (Bug #25987758)
- A check that ensures compatibility of referencing and referenced column types in a foreign key definition was moved from the storage engine layer to the SQL layer. In addition, a better error message is produced when columns are not compatible, and when a foreign key constraint references a table that does not exist. (Bug #25722927, Bug #28371394, Bug #91712, Bug #21308781, Bug #77467, Bug #11746132, Bug #23693)
- The parser performed some out-of-memory checks incorrectly. (Bug #25633994)
- A race condition between user-management statements and other statements that tried to access grant tables directly could result in deadlock and transaction rollback. (Bug #24481240)
- When the server was started with the `skip_name_resolve` system variable enabled, spurious warnings could be written to the error log about ignoring accounts with a host name part of `localhost`. (The accounts in fact were used and not ignored.) (Bug #23329861, Bug #81441)
- DML statements using `IGNORE` were not always handled correctly on tables having generated columns. (Bug #22990029)
- MySQL now removes trivial `WHERE` conditions arising from constant literal expressions during preparation, rather than at a later stage in optimization. This should result in improved plans for queries with outer joins containing trivial conditions, such as this one:

  ```
  SELECT * FROM t1 LEFT JOIN t2 ON condition_1 WHERE condition_2 OR 0 = 1
  ```

  After removing the redundant `OR 0 = 1` condition the optimizer can rewrite the query as an inner join, as shown here:

  ```
  SELECT * FROM t1 LEFT JOIN t2 WHERE condition_1 AND condition_2
  ```

  For more information, see What Is New in MySQL 8.0, and Outer Join Optimization. (Bug #16893426, Bug #28237111, Bug #28239008, Bug #28341790, WL #9571)

  References: See also: Bug #28197977, Bug #28240054.
- Updates for `BLOB` columns in `FEDERATED` tables did not work. (Bug #11748067, Bug #34997)
- Each of the functions REGEXP_REPLACE(), REGEXP_SUBSTR(), REGEXP_LIKE(), and REGEXP_INSTR() returned a DOUBLE instead of a value of the function's specified return type. (Bug #90039, Bug #27682225)
- A query employing a dynamic range and an index merge could use more memory than expected. (Bug #89953, Bug #27659490)
- Selecting from a table having a `CHAR` column with a `NO_PAD` collation yielded inconsistent results. (Bug #89753, Bug #27578340)

# Changes in MySQL 8.0.13 (2018-10-22)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.


- Account Management Notes
- Compilation Notes
- Configuration Notes
- Data Type Notes
- Deprecation and Removal Notes
- Error Handling
- INFORMATION_SCHEMA Notes
- Logging Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Pluggable Authentication
- Plugin Notes
- Security Notes
- Spatial Data Support
- SQL Syntax Notes
- XA Transaction Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Account Management Notes

- It is now possible to require that attempts to change an account password be verified by specifying the current password to be replaced. This enables DBAs to prevent users from changing a password without proving that they know the current password. It is possible to establish password-verification policy globally using the `password_require_current` system variable, as well as on a per-account basis using the `PASSWORD REQUIRE` option of the `CREATE USER` and `ALTER USER` statements. Together with existing password-management capabilities, the new capability of requiring verification provides DBAs more complete control over password management. For more information, see Password Management.

  **Important**

  The implementation of password-verification capability involves a change to the structure of the `mysql.user` system table. If you upgrade to this MySQL release

> from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate this system database change. Until this is done, *password changes are not possible.*

(WL #11544)

## Compilation Notes

- **Solaris:** MySQL now can be compiled on Solaris using `gcc`. (Bug #27802681)

## Configuration Notes

- The new `WITH_LTO` `CMake` option controls whether to enable link-time optimization. Currently, this is supported only by GCC 7 and 8. (Bug #28184537, Bug #28211382)
- The new `WITH_RAPIDJSON` `CMake` option controls whether to compile with the bundled or system RapidJSON library. (Bug #28024992, Bug #90867)
- The `CMAKE_BUILD_TYPE` `CMake` option now supports a `Release` build type, which is like the `RelWithDebInfo` build type but omits debugging information to reduce the build size. (Bug #27874068, Bug #32287863)
- The new `sql_require_primary_key` system variable makes it possible to have statements that create new tables or alter the structure of existing tables enforce the requirement that tables have a primary key. Enabling this variable helps avoid performance problems in row-based replication that can occur when tables have no primary key. Suppose that a table has no primary key and an update or delete modifies multiple rows. On the master server, this operation can be performed using a single table scan but, when replicated using row-based replication, results in a table scan for each row to be modified on the slave. With a primary key, these table scans do not occur. (Bug #17468242, Bug #69845, Bug #17005592, Bug #69223, WL #11807)
- To enable the server to listen on a set of addresses, the `bind_address` system variable now permits a list of comma-separated IP addresses or host names, not just a single address or name. For details, see Server System Variables. (WL #11652)

## Data Type Notes

- MySQL now supports use of expressions as default values in data type specifications. This includes the use of expressions as default values for the `BLOB`, `TEXT`, `GEOMETRY`, and `JSON` data types, which previously could not be assigned default values at all. For details, see Data Type Default Values. (WL #9418)

## Deprecation and Removal Notes

- **Important Change:** Setting user variables in statements other than `SET` is now deprecated due to issues that included those listed here:
  - The order of evaluation for expressions involving user variables was undefined.
  - The default result type of a variable is based on its type at the beginning of the statement, which could have unintended effects when a variable holding a value of one type at the beginning of a statement was assigned a new value of a different type in the same statement.
  - `HAVING`, `GROUP BY`, and `ORDER BY` clauses, when referring to a variable that was assigned a value in the select expression list, did not work as expected because the expression was evaluated on the client and so it was possible for stale column values from a previous row to be used.

Syntax such as `SELECT @var, @var:=@var+1` is still accepted in MySQL 8.0 for backward compatibility, but is subject to removal in a future MySQL version. (WL #12131)

- **InnoDB; Partitioning:** Support for placing table partitions in shared tablespaces was removed. Shared tablespaces include the system tablespace and general tablespaces. For information about identifying partitions in shared tablespaces and moving them to file-per-table tablespaces, see Preparing Your Installation for Upgrade. (WL #11571, WL #9286)
- **InnoDB:** Support for `TABLESPACE = innodb_file_per_table` and `TABLESPACE = innodb_temporary` clauses with `CREATE TEMPORARY TABLE` is deprecated and will be removed in a future MySQL version. (WL #12179)
- The `utf8mb3` character set is deprecated and will be removed in a future MySQL version. Please use `utf8mb4` instead. (WL #11109)
- Nested comments have never been supported (although they might be permitted under some conditions), but now are considered deprecated and will be removed in a future MySQL version. (WL #12099)
- The deprecated `metadata_locks_cache_size` and `metadata_locks_hash_instances` system variables were removed. (WL #12212)
- The `PAD_CHAR_TO_FULL_LENGTH` SQL mode has been deprecated and will be removed in a future MySQL version. (WL #12129)

## Error Handling

- The MySQL client library now returns better error messages for OpenSSL errors. Thanks to Facebook for the patch. (Bug #27855668, Bug #90418)
- Previously, the `ER_NO_REFERENCED_ROW_2` and `ER_ROW_IS_REFERENCED_2` error messages for foreign key operations were displayed and revealed information about parent tables, even when the user had no parent table access privileges. Error handling for this situation has been revised:
  - If the user does have table-level privileges for all parent tables, `ER_NO_REFERENCED_ROW_2` and `ER_ROW_IS_REFERENCED_2` are displayed, the same as before.
  - If the user does not have table-level privileges for all parent tables, more generic error messages are displayed instead (`ER_NO_REFERENCED_ROW` and `ER_ROW_IS_REFERENCED`).

  An exception is that, for stored programs defined to execute with `DEFINER` privileges, the user against which privileges are assessed is the user in the program `DEFINER` clause, not the invoking user. If that user has table-level parent table privileges, parent table information is still displayed. In this case, it is the responsibility of the stored program creator to hide the information by including appropriate condition handlers. (Bug #19477611, WL #8910)

## INFORMATION_SCHEMA Notes

- These new `INFORMATION_SCHEMA` tables are available as views on data dictionary tables:
  - `VIEW_ROUTINE_USAGE` provides information about stored functions used in view definitions.
  - `VIEW_TABLE_USAGE` provides information about tables and views used in view definitions.

  For more information, see The INFORMATION_SCHEMA VIEW_ROUTINE_USAGE Table, and The INFORMATION_SCHEMA VIEW_TABLE_USAGE Table. (WL #11864)

## Logging Notes

- **Incompatible Change:** The system variables have been removed that previously configured error logging to the system log (the Event Log on Windows, and `syslog` on Unix and Unix-like systems). Where appropriate, the removed system variables were replaced with new system variables managed by the `log_sink_syseventlog` error log component. The following table shows the old and new variable names.

| Old System Variable | New System Variable |
|---|---|
| `log_syslog_facility` | `syseventlog.facility` |
| `log_syslog_include_pid` | `syseventlog.include_pid` |
| `log_syslog_tag` | `syseventlog.tag` |
| `log_syslog` | None |

**Important**

Installations that used the old system variable names must update their configuration to use the new variable names. For more information, see Error Logging to the System Log.

(WL #11828)

References: See also: Bug #27534089.

- A new system variable, `log_error_suppression_list`, specifies which diagnostics should not be written to the error log when they occur with a severity of `WARNING` or `INFORMATION`. For example, if a particular type of warning occurs frequently but is not of interest (and thus may be considered undesirable “noise” in the error log), it now can be suppressed. (WL #11393)

- The code to handle rewriting of account-management statements was refactored to make it easier to maintain and extend. This work resulted in a few minor user-visible effects in the audit, general, and slow query logs:

  - Plaintext passwords are replaced by `<string>` rather than `'<string>'`.

  - Default clauses are not written unless specified by the user.

  (WL #12007)

- Previously, messages written to the error log by several internal server methods were logged using the `ER_LOG_PRINTF_MSG` error code. Each of these messages now is logged using a unique error code.

## Optimizer Notes

- The optimizer now supports a Skip Scan access method that enables range access to be used in previously inapplicable situations to improve query performance. For more information, see Skip Scan Range Access Method. Thanks to Facebook for the patch on which this access method is based. (Bug #26976512, Bug #88103, WL #11322)

- MySQL now supports creation of functional index key parts that index expression values rather than column values. Functional key parts enable indexing of values that cannot be indexed otherwise, such as `JSON` values. For details, see CREATE INDEX Statement. (WL #1075)

- Performance of `SELECT COUNT(*) FROM tbl_name` queries for `InnoDB` tables was improved for single-threaded workloads and when no extra clauses such as `WHERE` or `GROUP BY` are used. (WL #10398)

## Packaging Notes

- The `libevent` library bundled with MySQL was upgraded to version 2.1.8. (Bug #28207237, Bug #29041505, Bug #29055011)
- Binary packages that include `curl` rather than linking to the system `curl` library now use `curl` 7.60.0 rather than 7.45.0. (Bug #28043702)
- Debian packaging was updated to reflect the removal of yaSSL and that OpenSSL is the default SSL library for all builds. (Bug #28025599)
- Test plugins have been moved from server packages to test packages. (Bug #27860172)
- MySQL Router is now included in the MySQL server source and monolithic binary packages.

## Performance Schema Notes

- A new Performance Schema stage, `waiting for handler commit`, is available to detect threads going through transaction commit. Thanks to Facebook for the patch. (Bug #27855592, Bug #90417)

  References: See also: Bug #28950397.

## Pluggable Authentication

- **Microsoft Windows:** On Windows, MySQL Enterprise Edition distributions now bundle the Cyrus SASL library files `libsasl.dll` and `saslSCRAM.dll` so that the LDAP authentication plugins can use the `SCRAM-SHA-1` authentication method. (WL #11927)

## Plugin Notes

- Previously, MySQL plugins could be written in C or C++. MySQL header files used by plugins now contain C++ code, which means that plugins must be written in C++, not C. (Bug #87725, Bug #26781567)

## Security Notes

- MySQL Enterprise Edition now provides data masking and de-identification capabilities, implemented as a plugin library containing a plugin and a set of loadable functions. Data masking hides sensitive information by replacing real values with substitutes. MySQL Enterprise Data Masking and De-Identification functions enable masking existing data using several methods such as obfuscation (removing identifying characteristics), generation of formatted random data, and data replacement or substitution. For example:

```
mysql> SET @ssn = gen_rnd_ssn();
mysql> SET @masked_ssn1 = mask_ssn(@ssn);
mysql> SET @masked_ssn2 = mask_outer(mask_inner (@ssn,4,5,'A'), 3,0,'B');
mysql> SELECT @ssn, @masked_ssn1, @masked_ssn2;
+-------------+--------------+--------------+
| @ssn        | @masked_ssn1 | @masked_ssn2 |
+-------------+--------------+--------------+
| 980-31-2838 | XXX-XX-2838  | BBB-AA-2838  |
+-------------+--------------+--------------+
```

  For more information, see MySQL Enterprise Data Masking and De-Identification. (WL #7722)

## Spatial Data Support

- **Incompatible Change:** Previously, `ST_Area()` supported only geometry arguments having a Cartesian spatial reference system (SRS) and produced an error when invoked with geometry arguments that specified a geographic SRS. `ST_Area()` now supports geometry arguments having a geographic SRS and returns the geodetic area in square meters.

  **Note**

  If spatial data contains geometry values that now are interpreted differently by `ST_Area()`, existing queries using this function will return different results, compared to previous MySQL versions.

  (WL #10726)

- The parser for spatial reference system (SRS) definitions in `CREATE SPATIAL REFERENCE SYSTEM` statements now is stricter about rejecting invalid constructs.

  The OGC 01-009 WKT grammar does not make `AXIS` clauses mandatory in `GEOGCS` spatial reference system definitions. However, if there are no `AXIS` clauses, MySQL cannot determine whether a definition has axes in latitude-longitude order or longitude-latitude order. MySQL now enforces the nonstandard requirement that each `GEOGCS` definition must include two `AXIS` clauses. One must be `NORTH` or `SOUTH`, and the other `EAST` or `WEST`. The `AXIS` clause order determines whether the definition has axes in latitude-longitude order or longitude-latitude order. (Bug #28186073, Bug #28147723)

- Previously, `ST_Validate()` supported only geometry arguments having a Cartesian spatial reference system (SRS) and produced an error when invoked with geometry arguments that specified a geographic SRS. `ST_Validate()` now supports geometry arguments having a geographic SRS.

  **Note**

  If spatial data contains geometry values that now are interpreted differently by `ST_Validate()`, existing queries using this function will return different results, compared to previous MySQL versions.

  (WL #11087)

- MySQL now implements the `ST_Transform()` spatial function for use in converting geometry values from one spatial reference system (SRS) to another. Currently, it supports conversion between geographic SRSs. For details, see Spatial Operator Functions. (WL #8684)

## SQL Syntax Notes

- **Incompatible Change:** The deprecated `ASC` or `DESC` qualifiers for `GROUP BY` clauses have been removed. Queries that previously relied on `GROUP BY` sorting may produce results that differ from previous MySQL versions. To produce a given sort order, provide an `ORDER BY` clause.

  Queries and stored program definitions from MySQL 8.0.12 or lower that use `ASC` or `DESC` qualifiers for `GROUP BY` clauses should be amended. Otherwise, upgrading to MySQL 8.0.13 or higher may fail, as may replicating to MySQL 8.0.13 or higher slave servers. (WL #8693)

## XA Transaction Notes

- Previously, metadata locks for XA transactions in `PREPARED` state could be dropped when the client performing the transaction disconnected or the server restarted. This could lead to behavior such as one session being able to drop tables used by an XA transaction in progress for another session. Metadata

locks for XA transactions in `PREPARED` state now are maintained across client disconnects and server restarts, until an `XA COMMIT` or `XA ROLLBACK` is executed. (Bug #22710164, Bug #79940, WL #9335)

## X Plugin Notes

- **Important Change:** X Protocol now provides a connection pooling option, which can reduce overhead for applications that open many connections to MySQL server such as small web pages or a REST API endpoint. Use the new `mysqlx.getClient(connection, options)` operation, which returns a `Client` object. By using a `Client` object, executing an open session operation retrieves an existing and currently unused network connection from the pool, resets it, and uses it. Closing a session marks the underlying connection as unused and returns it to the `Client` object's connection pool. The connection pool is configured using a options data dictionary, which means that a deployed application can switch from using connection pooling or not by simply changing the connection string. (WL #11824)
- A mutex lock has been added to prevent conflict between multiple X Plugin client sessions when a large number of sessions are being opened and closed. (Bug #28637947)
- When an X Plugin client attempted to connect to a MySQL server specifying a database that did not exist on the server, the error message indicated that access was denied rather than that the database was not known. The correct error message is now returned. (Bug #28110957)
- Incorrect copying of an integer value by X Plugin caused an error relating to misaligned memory access. The issue is now fixed. (Bug #28070946, Bug #90983)
- Now that X Plugin is loaded and enabled by default, the default verbosity setting for the error log meant that no message was visible to indicate that X Plugin was available on the MySQL server. A message is now issued during system startup to confirm that X Plugin has been loaded. (Bug #27287340)
- X Protocol now provides the ability to broadcast information to interested clients without the requirement to request the information. In addition, changes related to Group Replication are also broadcast. This work also adds the `Mysqlx_notified_by_group_replication` and `Mysqlx_notice_global_sent` status variables. (WL #10655)

## Functionality Added or Changed

- **Important Change; NDB Cluster; NDB Client Programs:** Removed the deprecated `--ndb` option for `perror`. Use `ndb_perror` to obtain error message information from `NDB` error codes instead. (Bug #81705, Bug #23523957)

  References: See also: Bug #81704, Bug #23523926.

- **InnoDB:** The `innodb_fsync_threshold` system variable permits defining a threshold, in bytes, for flushing data to disk from the operating system cache. By default, when `InnoDB` creates a new data file, such as a new log file or tablespace file, the file is fully written to the operating system cache before it is flushed to disk, which can cause a large amount of disk write activity to occur at once.

  Specifying a threshold to force smaller, periodic flushes may be beneficial in cases where multiple MySQL instances use the same storage devices. For example, creating a new MySQL instance and its associated data files could cause large surges of disk write activity, impeding the performance of other MySQL instances that use the same storage devices. Configuring a threshold helps avoid such surges in write activity. (Bug #27724600)

- **InnoDB:** User-created temporary tables and internal temporary tables created by the optimizer are now stored in session temporary tablespaces that are allocated to a session from a pool of temporary tablespaces. When a session disconnects, its temporary tablespaces are truncated and released back to the pool. In previous releases, temporary tables were created in the global temporary tablespace (`ibtmp1`), which did not return disk space to the operating system after temporary tables were dropped.

The `innodb_temp_tablespaces_dir` variable defines the location where session temporary tablespaces are created. The default location is the `#innodb_temp` directory in the data directory.

The `INNODB_SESSION_TEMP_TABLESPACES` table provides metadata about session temporary tablespaces.

The global temporary tablespace (`ibtmp1`) now stores rollback segments for changes made to user-created temporary tables. (WL #11613)

- **InnoDB:** The `InnoDB` data-at-rest encryption feature now supports general tablespaces. Previously, only file-per-table tablespaces could be encrypted. To support encryption of general tablespaces, `CREATE TABLESPACE` and `ALTER TABLESPACE` syntax was extended to include an `ENCRYPTION` clause.

  The `INFORMATION_SCHEMA.INNODB_TABLESPACES` table now includes an `ENCRYPTION` column that indicates whether or not a tablespace is encrypted.

  The `stage/innodb/alter tablespace (encryption)` Performance Schema stage instrument was added to permit monitoring of general tablespace encryption operations. (WL #9286)

- **Replication:** The `group_replication_member_expel_timeout` option has been added to define the time period that the group should wait for a non-responding member before evicting the member from the group. This enables you to configure the eviction process when the connection to members is unreliable. (WL #11570)

- **Group Replication:** You can now inspect and configure the maximum number of consensus instances at any time for a group. This maximum is referred to as the event horizon for a group, and is the maximum number of consensus instances that the system can execute in parallel. This enables you to fine tune the performance of your Group Replication deployment.

  To inspect a group's event horizon value at runtime, issue this statement:

  ```
  SELECT group_replication_get_write_concurrency()
  ```

  To set the maximum number of write consensus instances, issue the statement shown here:

  ```
  SELECT group_replication_set_write_concurrency(instances);
  ```

  *`instances`* is the maximum number of instances used for consensus.

  See Using Group Replication Group Write Consensus for more information. (Bug #27260096, WL #11512)

- **Group Replication:** You can now make changes to the configuration of a group while it is running, without having to stop all members to make changes. This functionality relies on loadable functions

which are installed with this version of the plugin, and all members of a group must have these functions installed. To use the functions, connect to an online member and issue `SELECT function_name;`

Use the `group_replication_set_as_primary()` function to trigger the election of a specific member as the new primary in a single-primary group, overriding the usual election process. See Changing the Primary for more information.

In addition, you can configure the mode which a group is using while it is online, changing between single-primary mode and multi-primary mode. To change the mode of an online group, choose one of these options:

- Use `group_replication_switch_to_single_primary_mode()` to change a group running in multi-primary mode to single-primary mode.
- Use `group_replication_switch_to_multi_primary_mode()` to change a group running in single-primary mode to multi-primary mode.

See Changing the Group Mode for more information. (WL #10378)

- **Solaris:** On Solaris, MySQL can now be built with Developer Studio 12.6. (Bug #27055190, Bug #88316, Bug #28165246, Bug #91214)
- Data truncation testing was rewritten to avoid undefined behavior. (Bug #28255956, Bug #91445)
- Out-of-range checking for float values was improved. (Bug #28225635)
- The upgrade check that the server runs during the startup process now verifies that partitioned `InnoDB` tables do not use shared tablespaces. (Bug #28204431)
- Previously, file I/O performed in the I/O cache in the `mysys` library was not instrumented, affecting in particular file I/O statistics reported by the Performance Schema about the binary log index file. Now, this I/O is instrumented and Performance Schema statistics are accurate. Thanks to Yura Sorokin for the contribution. (Bug #27788907, Bug #90264)
- Performance for locating user account entries in the in-memory privilege structures has been improved. Thanks to Eric Herman for the contribution. (Bug #27772506, Bug #90244)
- If `mysqld --initialize` fails to complete but creates an unusable data directory, it now displays a message that the data directory is unusable and can safely be removed. (Bug #27675647)
- Instrumentation is now provided in the Performance Schema for transaction retries by individual applier threads on a single-threaded or multithreaded slave. Previously, the Performance Schema table `replication_applier_status_by_worker` displayed information about errors that stopped the applier thread, but not about transient errors that occurred before a transaction was eventually applied. With this information, you can identify transient errors that are causing replication lag on replication slaves or Group Replication group members.

  Eight new columns have been added to the Performance Schema `replication_applier_status_by_worker` table:

  - `LAST_APPLIED_TRANSACTION_RETRIES_COUNT` - The number of times the last applied transaction was retried by the worker after the first attempt. If the transaction was applied at the first attempt, this number is zero.
  - `LAST_APPLIED_TRANSACTION_LAST_TRANSIENT_ERROR_NUMBER` - The error number of the last transient error that caused the transaction to be retried.

- `LAST_APPLIED_TRANSACTION_LAST_TRANSIENT_ERROR_MESSAGE` - The message text for the last transient error that caused the transaction to be retried.
- `LAST_APPLIED_TRANSACTION_LAST_TRANSIENT_ERROR_TIMESTAMP` - The timestamp in `'YYYY-MM-DD hh:mm:ss[.fraction]'` format for the last transient error that caused the transaction to be retried.
- `APPLYING_TRANSACTION_RETRIES_COUNT` - The number of times the transaction that is currently being applied was retried until this moment. If the transaction was applied at the first attempt, this number is zero.
- `APPLYING_TRANSACTION_LAST_TRANSIENT_ERROR_NUMBER` - The error number of the last transient error that caused the current transaction to be retried.
- `APPLYING_TRANSACTION_LAST_TRANSIENT_ERROR_MESSAGE` - The message text for the last transient error that caused the current transaction to be retried.
- `APPLYING_TRANSACTION_LAST_TRANSIENT_ERROR_TIMESTAMP` - The timestamp in `'YYYY-MM-DD hh:mm:ss[.fraction]'` format for the last transient error that caused the current transaction to be retried.

The existing column `APPLYING_TRANSACTION_START_APPLY_TIMESTAMP` is no longer reset each time a transaction is retried. It now retains the timestamp from the worker's first attempt to apply the transaction. (WL #9131)

- Previously, `CREATE TEMPORARY TABLE` and `DROP TEMPORARY TABLE` statements were not supported inside transactions, procedures, functions, or triggers when using GTIDs (that is, when the `enforce_gtid_consistency` system variable is set to `ON`). It was possible to use these statements with GTIDs enabled, but only outside of any transaction, and only with `autocommit=1`.

  From MySQL 8.0.13, this restriction has been removed when `binlog_format` is set to `ROW` or `MIXED`. With row-based logging in use, `CREATE TEMPORARY TABLE` and `DROP TEMPORARY TABLE` statements can now be used inside transactions, procedures, functions, or triggers when GTIDs are enabled. When `binlog_format` is set to `STATEMENT`, the restriction remains. Because of this difference in behavior, some additional restrictions now apply to changing the `binlog_format` setting at runtime:

  - If a session has open temporary tables, the replication format cannot be changed for the session (`SET @@SESSION.binlog_format`).
  - If any replication channel has open temporary tables, the replication format cannot be changed globally (`SET @@GLOBAL.binlog_format` or `SET @@PERSIST.binlog_format`).
  - If any replication channel applier thread is currently running, the replication format cannot be changed globally (`SET @@GLOBAL.binlog_format` or `SET @@PERSIST.binlog_format`).

  Trying to switch the replication format in any of these cases (or attempting to set the current replication format) results in an error. You can, however, use `PERSIST_ONLY` (`SET @@PERSIST_ONLY.binlog_format`) to change the replication format at any time, because this action does not modify the runtime global system variable value, and takes effect only after a server restart.

  When `binlog_format` is set to `ROW` or `MIXED`, `CREATE TEMPORARY TABLE` and `DROP TEMPORARY TABLE` statements are not written to the binary log and are therefore not replicated to slaves. When they are used in transactions, if the removal of these statements from the transaction results in an empty transaction, the transaction is not written to the binary log. If a transaction involving these statements is

rolled back, a warning message is issued stating that the creation or dropping of the temporary tables could not be rolled back. (WL #8872)

- The MySQL Server code for binary logging has been refactored to create new internal interfaces for accessing events in a binary log and relay log. The new interfaces separate the write and read processes for binary logging into input and output streams, and decouple the process of capturing and retrieving binary log events from the process of writing them to files. A logical binary log file is used to wrapper the storage layer operations.

  The new internal interfaces make it possible for MySQL Server to use alternative storage methods for binary log events besides a standard binary log or relay log file, including a binary log cache or memory buffer. For example, Group Replication uses the new interfaces to serialize events directly to memory buffers and transaction messages in order to co-ordinate transactions in the group. `mysqlbinlog` also now uses them to read binary log events from standard input.

  The following existing error messages are now marked as obsolete due to the new internal interfaces:

  - ER_BINLOG_CANT_OPEN_LOG
  - ER_BINLOG_CANT_CREATE_CACHE_FOR_LOG
  - ER_BINLOG_ERROR_GETTING_NEXT_LOG_FROM_INDEX
  - ER_RPL_RECOVERY_ERROR_FREEING_IO_CACHE
  - ER_GRP_RPL_REINIT_OF_INTERNAL_CACHE_FOR_READ_FAILED
  - ER_GRP_RPL_APPENDING_DATA_TO_INTERNAL_CACHE_FAILED
  - ER_GRP_RPL_REINIT_OF_INTERNAL_CACHE_FOR_WRITE_FAILED
  - ER_GRP_RPL_FAILED_TO_CREATE_COMMIT_CACHE
  - ER_GRP_RPL_REINIT_OF_COMMIT_CACHE_FOR_WRITE_FAILED

  (WL #10956)

- The `TempTable` storage engine now supports storage of binary large object (BLOB) type columns. This enhancement improves performance for queries that use temporary tables containing BLOB data. Previously, temporary tables that contained BLOB data were stored in the on-disk storage engine defined by `internal_tmp_disk_storage_engine`. (WL #11452)

- The MySQL Server code for deserialization of binary log events has been refactored to improve the identification and resilient handling of corrupted and invalid event data during replication, and also during processing with `mysqlbinlog`, whether or not binary log checksums are used on the server. (WL #11567)

- Previously, executing `RENAME TABLE` required that there be no tables locked with `LOCK TABLES`. Now it is possible to rename tables that are locked with a `WRITE` lock or that are the product of renaming `WRITE`-locked tables from earlier steps in a multiple-table rename operation. (WL #9826)

## Bugs Fixed

- **InnoDB; Partitioning:** Removed old `InnoDB` handler and partitioning code that referenced `.frm` files, and thus no longer had any purpose. (Bug #27995316)

- **InnoDB:** An assertion was raised during a `DROP TABLE` operation. A thread that was accessing the table through the memcached API released metadata locks before releasing the table. (Bug #28531148)
- **InnoDB:** The `being_modified` bit in a LOB reference was set but the bit modification was not logged, causing an assertion failure. (Bug #28443837)
- **InnoDB:** Window functions returned incorrect results when the optimizer used the `InnoDB` storage engine for internal temporary tables. (Bug #28430650)
- **InnoDB:** Adjusting the server time to an earlier time caused periodic redo flushes to be missed. (Bug #28430358, Bug #90670)
- **InnoDB:** An `ALTER TABLE` operation that added a primary key produced a segmentation fault. (Bug #28395278)

  References: This issue is a regression of: Bug #27753193.
- **InnoDB:** A conditional check was removed by removing the `ReadView::complete()` function and splitting its work among other functions. This change helps optimize performance on ARM 64-bit. (Bug #28385211, Bug #91759)
- **InnoDB:** Leftover `thread_mutex` code was removed from `InnoDB` source code files. (Bug #28363673, Bug #91678)
- **InnoDB:** Type changes were implemented to eliminate warnings that occurred when compiling `InnoDB` with Microsoft Visual Studio 2017. (Bug #28338720)
- **InnoDB:** An invalid assertion was raised when a B-tree flag used to mark shared index locks was used to mark a shared-exclusive index lock. (Bug #28317172)
- **InnoDB:** The sharp checkpoint mechanism no longer forces preflushing of dirty pages when requesting a checkpoint for the currently available LSN.

  The log checkpointer thread now takes the concurrency margin (the per thread margin for free space in the log) into account when determining if the next checkpoint write is required and whether to wake up page cleaners to force a sync-flush of dirty pages. Page cleaner threads take the concurrency margin into account when determining whether to flush dirty pages and how many pages to flush. (Bug #28297462)
- **InnoDB:** A misplaced debug crash point caused a transaction timeout resulting in test failures. (Bug #28295814)
- **InnoDB:** `InnoDB` error message format was modified to remove duplicate text. (Bug #28289789)
- **InnoDB:** Unnecessary cycles of freeing and allocating memory caused JSON performance degradation on Windows. (Bug #28278737)
- **InnoDB:** `InnoDB` incorrectly permitted dropping an index used in a foreign key constraint using an in-place `ALTER TABLE` statement. This operation is now blocked at the SQL layer. (Bug #28268875)
- **InnoDB:** To avoid checking hardware support each time a hardware-optimized checksum is computed, asserts were converted to debug-only asserts. (Bug #28267334, Bug #91485)
- **InnoDB:** A patch that combined Contention-Aware Transaction Scheduling (CATS) with functionality that releases read locks caused gap locks to be removed without granting locks to waiting transactions, resulting in transaction timeouts. (Bug #28261530)

  References: This issue is a regression of: Bug #28261530.

- **InnoDB:** The `log_checkpointer` thread failed to write new checkpoints in a timely manner when the amount of redo was small. (Bug #28220222)

- **InnoDB:** The server exited during an in-place upgrade from MySQL 5.7 to MySQL 8.0 due to an attempted eviction of a foreign-key-related table from the cache. At the end of the upgrade process, tables with `FULLTEXT` indexes were marked as ready for eviction without checking for foreign key relationships. (Bug #28212734, Bug #91325)

- **InnoDB:** The format of the following Performance Schema and `INFORMATION_SCHEMA` table columns was modified:

  - `data_locks.ENGINE_LOCK_ID`
  - `data_lock_waits.REQUESTING_ENGINE_LOCK_ID`
  - `data_lock_waits.BLOCKING_ENGINE_LOCK_ID`
  - `INNODB_TRX.TRX_REQUESTED_LOCK_ID`

  The previous format was `trx_id:table_id` for table locks and `trx_id:space_id:page_no:heap_no` for record locks. The new format is `trx_immutable_id:table_id:lock_immutable_id` for table locks and `trx_immutable_id:space_id:page_no:heap_no:lock_immutable_id` for record locks.

  `lock_immutable_id` and `trx_immutable_id` are 64-bit values that do not change during the lifetime of a lock or transaction, respectively, and are unique among other instance object IDs. (Bug #28176910)

- **InnoDB:** The list of permitted lock mode descriptors used by the `LOCK_MODE` column of the Performance Schema `data_locks` table was expanded to include `REC_NOT_GAP`, `INSERT_INTENTION`, `PREDICATE`, and `PRDT_PAGE`. `REC_NOT_GAP` indicates a record-only lock. `INSERT_INTENTION` indicates an insert intention lock. `PREDICATE` and `PRDT_PAGE` descriptors indicate a spatial index lock. (Bug #28176805)

- **InnoDB:** Table names were not compared in lowercase on macOS with a setting of `lower_case_table_names=2`, which caused instability after restarting the server. (Bug #28170699, Bug #91204)

- **InnoDB:** Macros used to define constant values in `InnoDB` source code were changed to constant expressions. (Bug #28152926)

- **InnoDB:** A flag that prevents transactions from being rolled back during commit is now set earlier to prevent scenarios in which a high priority transaction could abort a transaction that is in the process of being committed. (Bug #28140462)

- **InnoDB:** A query that scanned the primary key of a table did not return the expected result. (Bug #28104394, Bug #91032)

- **InnoDB:** Unnecessary header file inclusions were removed from `InnoDB` source code files. (Bug #28086759)

- **InnoDB:** An apparent hang due to the log writer running out of free space in the redo log caused the server to exit. (Bug #28072385, Bug #90993)

- **InnoDB:** A query interruption during a lock wait caused an error. (Bug #28068293)

- **InnoDB:** After upgrading from MySQL 5.7 to MySQL 8.0, invalid warnings indicated that undo tablespace IDs were not in the undo tablespace range. The warnings occurred if the MySQL 5.7 installation was configured to use separate undo tablespaces. (Bug #28060337)
- **InnoDB:** Error messaging was improved for startup failures on an incomplete cloned data directory. (Bug #28032131)
- **InnoDB:** A segmentation fault occurred during an `XA COMMIT` operation. (Bug #27995891)
- **InnoDB:** Unused code was removed from the TempTable storage engine source code. (Bug #27978968)
- **InnoDB:** An index record was not found when updating a secondary index defined on a generated column. (Bug #27968952)
- **InnoDB:** A spurious negation operator in an IF condition caused the Contention-Aware Transaction Scheduling (CATS) algorithm to be used for table locks. (Bug #27944920)

  References: This issue is a regression of: Bug #27572937.
- **InnoDB:** The update log applied as part of an online `ALTER TABLE` operation did not take into account the computed value of the generated column in the old row while updating the secondary index. (Bug #27921932)
- **InnoDB:** Buffer pool memory allocation was not fully accounted for in Performance Schema `memory/innodb/buf_buf_pool` statistics. (Bug #27917595, Bug #90561)
- **InnoDB:** An unsupported DDL operation involving a foreign key constraint raised an assertion. (Bug #27912873)
- **InnoDB:** A lock-related debug assertion failure was raised when more than one lock matched a search condition, which could lead to releasing the wrong lock. (Bug #27898384)
- **InnoDB:** A function that removes aborted indexes during the prepare phase of an online `ALTER TABLE` operation did not record its changes. (Bug #27879325)
- **InnoDB:** A general tablespace created in MySQL 5.7 with no assigned table caused a failure when upgrading to MySQL 8.0. (Bug #27877485)
- **InnoDB:** Concurrent undo tablespace truncation and master key rotation operations raised an assertion. (Bug #27872369)
- **InnoDB:** A boolean marker identifying whether a transaction holds a mutex was not placed in the correct location. (Bug #27870035)
- **InnoDB:** An attempted foreign key check on a discarded table caused a segmentation fault. (Bug #27804668)
- **InnoDB:** B-tree bulk load operations could leave a page in a partially initialized state. (Bug #27802098)
- **InnoDB:** Starting the server inside a Docker container on a NUMA enabled operating system raised an “mbind: Operation not permitted” error. (Bug #27792853)
- **InnoDB:** A partitioned table `TABLE_ID` value stored in a storage-engine-private data field in the data dictionary was not adjusted properly after an `ALTER TABLE ... PARTITION` operation. (Bug #27784462)
- **InnoDB:** The server halted with a “log writer overwriting data after checkpoint - waited too long” error. (Bug #27779266)

- **InnoDB:** With `innodb_flush_log_at_trx_commit=2`, the `log_flusher` thread could wait for an event with a timeout period equal to the `innodb_flush_log_at_timeout` setting, causing an initialization delay. (Bug #27762596)
- **InnoDB:** An assertion was raised during an `OPTIMIZE TABLE` operation. (Bug #27753193)
- **InnoDB:** Transaction rollback due to a deadlock caused an assertion failure in debug builds. Initiation of an attachable transaction for accessing the data dictionary was not expected during transaction rollback. (Bug #27729974)
- **InnoDB:** With `innodb_flush_log_at_trx_commit=0` and binary logging enabled, redo logs were not flushed as expected during the commit phase of DDL operations. (Bug #27691035)
- **InnoDB:** With `REDUNDANT` or `COMPRESSED` row format and `READ COMMITTED` isolation level, only the LOB value prefix and possibly the external part of an old LOB value were returned, which could cause a JSON document to be viewed as corrupted. If there was no LOB value prefix, an old LOB value with new values for other fields could be returned, causing data inconsistency. (Bug #27624990)
- **InnoDB:** A debug option that permits pausing periodic checkpoints became obsolete after periodic checkpoint ownership was moved from the master thread to a log checkpointer thread. The debug option was replaced by another method of pausing periodic checkpoints. (Bug #27588328)
- **InnoDB:** A transaction on a table with a spatial index defined on a column with a spatial reference identifier (SRID) was able to insert into an area selected for update by another transaction. (Bug #27577612)
- **InnoDB:** A foreign key constraint name was duplicated during a rename table operation, causing a failure during later query execution. (Bug #27545888)
- **InnoDB:** A serialized dictionary information (SDI) deletion failure raised an assertion. (Bug #27493634)
- **InnoDB:** A server exit occurred after freeing large object (LOB) index entries during a LOB purge or rollback. (Bug #27419474)
- **InnoDB:** In a function called before the execution of a statement in a stored procedure, a read and write operation on `trx->lock.start_stmt` was not protected by a mutex. (Bug #27325898)
- **InnoDB:** The `INFORMATION_SCHEMA.FILES` and `INFORMATION_SCHEMA.INNODB_TABLESPACES` tables did not show the actual undo tablespaces that were present in the MySQL instance. Only the two default undo tablespaces were shown. (Bug #26820406)
- **InnoDB:** An error occurred during a DDL operation due to a mismatch in a `REDUNDANT` row format calculation that determines the length of the online log. (Bug #26375771)
- **InnoDB:** Enabling `innodb_undo_log_truncate` negatively affected transaction processing performance. Instead of performing two checkpoints during an undo tablespace truncate operation, pages that belong to the tablespace file are now flushed from disk. (Bug #26322656)
- **InnoDB:** A helper class was introduced to improve performance associated with reading from secondary keys when there are multiple versions of the same row.

  Thanks to Domas Mituzas for the contribution. (Bug #25540277, Bug #84958)
- **InnoDB:** The location of the Innodb Merge Temp File that reported by the `wait/io/file/innodb/innodb_temp_file` Performance Schema instrument was incorrect. (Bug #21339079, Bug #77519)
- **Partitioning:** When a `CREATE TABLE ... PARTITION BY ...` statement failed due to an invalid partition definition, the server did not remove any partition files which might have been created prior to encountering the invalid `PARTITION` clause. (Bug #27798708)

References: See also: Bug #88043, Bug #26945644.

- **Partitioning:** It was possible to perform `FLUSH TABLES FOR EXPORT` on a partitioned table created with `innodb_file_per_table=1` after discarding its tablespace. Attempting to do so now raises `ER_TABLESPACE_DISCARDED`. (Bug #90545, Bug #27903881)

  References: See also: Bug #80669, Bug #22899690.

- **Partitioning:** An extraneous row lock was imposed by an update to a partitioned `InnoDB` table. (Bug #87253, Bug #26553164)

- **Replication:** When a replication slave is restarted by a `START SLAVE` statement, the columns in the Performance Schema table `replication_applier_status_by_worker` beginning `APPLYING_TRANSACTION` are now reset on a slave that is operating in single-threaded mode. These columns were always reset on a multi-threaded slave, because the existing worker threads were terminated by the statement and the information could not be retained. The behavior has now been standardized across the slave configurations by resetting the columns for a single-threaded slave as well. (Bug #28248026)

- **Replication:** If a multi-threaded replication slave was stopped, changed to a single-threaded slave (by setting `slave_parallel_workers > 0`), and restarted, the Performance Schema table `replication_applier_status_by_worker` showed irrelevant timestamps because the old monitoring information had not been cleared. (Bug #28191382)

- **Replication:** When the `binlog_group_commit_sync_delay` system variable is set to a wait time to delay synchronization of transactions to disk, and the `binlog_group_commit_sync_no_delay_count` system variable is also set to a number of transactions, the MySQL server exits the wait procedure if the specified number of transactions is reached before the specified wait time is reached. The server manages this process by checking on the transaction count after a delta of one tenth of the time specified by `binlog_group_commit_sync_delay` has elapsed, then subtracting that interval from the remaining wait time.

  If rounding during calculation of the delta meant that the wait time was not a multiple of the delta, the final subtraction of the delta from the remaining wait time would cause the value to be negative, and therefore to wrap to the maximum wait time, making the commit hang. The data type for the remaining wait time has now been changed so that the value does not wrap in this situation, and the commit can proceed when the original wait time has elapsed. Thanks to Yan Huang for the contribution. (Bug #28091735, Bug #91055)

- **Replication:** In debug builds, an assertion failed because more than 255 collations are now available in MySQL. (Bug #28015761)

- **Replication:** An assertion was raised in debug builds because the MySQL server recorded a GTID consistency violation, but did not remove the record after the relevant statement failed to execute successfully. The handling of this situation has now been improved to ensure that the server checks at the end of a transaction whether a GTID consistency violation was produced by a failed statement, and if this is the case, restores the previous GTID consistency state. (Bug #27903831, Bug #90551)

- **Replication:** With GTIDs in use for replication, transactions including statements that caused a parsing error (`ER_PARSE_ERROR`) could not be skipped manually by the recommended method of injecting an empty or replacement transaction with the same GTID. This action should result in the slave identifying the GTID as already used, and therefore skipping the unwanted transaction that shared its GTID. However, in the case of a parsing error, because the statement was parsed before the GTID was checked to see if it needed to be skipped, the replication applier thread stopped due to the parsing error, even though the intention was for the transaction to be skipped anyway.

With this fix, the replication applier thread now ignores parsing errors if the transaction concerned needs to be skipped because the GTID was already used. Note that this behavior change does not apply in the case of workloads consisting of binary log output produced by `mysqlbinlog`. In that situation, there would be a risk that a transaction with a parsing error that immediately follows a skipped transaction would also be silently skipped, when it ought to raise an error. (Bug #27638268)

- **Replication:** When a `RESET SLAVE` statement was issued on a replication slave with GTIDs in use, the existing relay log files were purged, but the replacement new relay log file was generated before the set of received GTIDs for the channel had been cleared. The former GTID set was therefore written to the new relay log file as the `PREVIOUS_GTIDS` event, causing a fatal error in replication stating that the slave had more GTIDs than the master, even though the `gtid_executed` set for both servers was empty. Now, when `RESET SLAVE` is issued, the set of received GTIDs is cleared before the new relay log file is generated, so that this situation does not occur. (Bug #27636289)

- **Replication:** The master's receiver thread for semisynchronous replication held a mutex while reading acknowledgements from slaves, but the same mutex was required to add or remove a semisynchronous slave, causing those operations to be delayed by the acknowledgement activity. The issue has now been fixed by not acquiring the mutex to read the acknowledgements from slaves. (Bug #27610678, Bug #89370)

- **Replication:** In code for replication slave reporting, a rare error situation raised an assertion in debug builds, but in release builds, returned leaving a mutex locked. The mutex is now unlocked before returning in this situation. Thanks to Zsolt Parragi for the patch. (Bug #27448019, Bug #89421)

- **Replication:** Automatic retrying of transactions on a replication slave, as specified by the `slave_transaction_retries` system variable, was taking place even if the transaction had a non-temporary error that would repeat on retrying or that indicated wider issues. Now, transactions are only automatically retried if there is either no error, or an error that is only temporary. (Bug #27373559, Bug #89143)

- **Replication:** When `FLUSH` statements for specific log types (such as `FLUSH SLOW LOGS`) resulted in an error, the statements were still written to the binary log. This stopped replication because the error had occurred on the master, but did not occur on the slave. MySQL Server now checks on the outcome of these `FLUSH` statements, and if an error occurred, the statement is not written to the binary log. (Bug #24786290, Bug #83232)

- **Replication:** The `PASSWORD()` function, which produces a hash of the password, was deprecated in MySQL 5.7 and removed in MySQL 8.0. When a `SET PASSWORD` statement that used this function was replicated from a MySQL 5.6 master to a MySQL 5.7 slave, or from a MySQL 5.7 master with the `log_builtin_as_identified_by_password` system variable set to ON to a MySQL 5.7 slave, the password hash was itself also hashed before being stored on the slave. The issue has now been fixed and the replicated password hash is stored as originally passed to the slave. (Bug #24687073)

- **Replication:** If an `ORDER BY` clause was used in retrieving records from certain Performance Schema tables relating to replication, an empty set was returned. The issue has now been fixed. (Bug #22958077, Bug #80777)

- **Replication:** When replication channels are used on a slave for multi-source replication, a `START SLAVE` statement that does not specify an individual channel (so without the `FOR CHANNEL` clause) should start the I/O thread and the SQL thread for all of the channels on the replication slave. However, if a `RESET SLAVE` statement was used on such a slave, a subsequent `START SLAVE` statement did not start the non-default channels. Now, replication channels that are deinitialized as a result of a `RESET SLAVE` statement, rather than as a result of an error in the initialization process, are identified and are restarted by a `START SLAVE` statement that applies to all channels. (Bug #22809607)

- **Replication:** Issuing `RESET SLAVE` on a replication slave does not change any replication connection parameters such as master host, master port, master user, or master password, which are retained in memory. However, these connection parameters are reset if you issue `RESET SLAVE ALL`. Previously, if the slave `mysqld` was restarted immediately after issuing `RESET SLAVE` (including a server crash as well as a deliberate restart), the connection parameters were reset as if `RESET SLAVE ALL` had been used.

  Now, when `master_info_repository=TABLE` is set on the server (which is the default from MySQL 8.0), replication connection parameters are preserved in the crash-safe `InnoDB` table `mysql.slave_master_info` as part of the `RESET SLAVE` operation. They are also retained in memory. In the event of a server crash or deliberate restart after issuing `RESET SLAVE` but before issuing `START SLAVE`, the replication connection parameters are retrieved from the table and reused for the new connection.

  If `master_info_repository=FILE` is set on the server (which is the default in MySQL 5.7), replication connection parameters are only retained in memory, so the behavior remains the same as previously. If the slave `mysqld` is restarted due to a server crash or a deliberate restart immediately after issuing `RESET SLAVE`, the connection parameters are lost. In that case, you must issue a `CHANGE MASTER TO` statement after the server start to respecify the connection parameters before issuing `START SLAVE`.

  If you want to reset the connection parameters intentionally, you need to use `RESET SLAVE ALL`, which clears the connection parameters. In that case, you must issue a `CHANGE MASTER TO` statement after the server start to specify the new connection parameters. (Bug #20280946)

- **Replication:** When a group member resumes after being suspended for some time and is not able to process all pending messages, it enters the `ERROR` state. However, the remaining members see it as `UNREACHABLE`, and wait until the member's suspicion expires to evict it from the group. The behavior has now been modified and a member stopping due to some error tries to connect to a known peer to request its removal from the group, before installing the leave view. (Bug #91433, Bug #28252687)

- **Replication:** Compilation warnings related to unused functions in `xdr_utils` have been reduced. Thanks to Zsolt Parragi for the patch. (Bug #91071, Bug #28099963)

- **Replication:** If the `group_replication_recovery_retry_count` variable was modified while the member was already making a reconnection attempt, the connection attempt could enter an infinite loop. (Bug #91057, Bug #28092714)

- **Replication:** The `group_replication_exit_state_action` variable enables you to specify what action is taken if a member involuntarily leaves the group, but when starting a server with `group_replication_start_on_boot` enabled the `group_replication_exit_state_action` variable was being ignored during the following scenarios:

  - valid number of group members was exceeded
  - incompatible configuration of the member system variables (various)
  - the joining member had more transactions than the group
  - the joining member's version was not compatible with the group

  (Bug #90494, Bug #27881311)

- **Replication:** Issuing `START GROUP_REPLICATION` and then forcibly stopping the `mysqld` process, for example using **control-C**, could result in an unexpected halt of the server. (Bug #90457, Bug #27873419)

- **Group Replication:** Entries in the relay log info log for the `group_replication_applier` and `group_replication_recovery` channels were not cleared by `RESET SLAVE` or `RESET SLAVE ALL`. (Bug #27411175)
- **Group Replication:** When Group Replication was started on server with an invalid `group_replication_group_name`, the server stopped unexpectedly. (Bug #91347, Bug #28219136)
- **Group Replication:** When `group_replication_group_seeds` contained a DNS based entry which resolved to its own local address, Group Replication was unable to start. (Bug #90483, Bug #27882096, Bug #28074929)
- **Microsoft Windows:** An error now is written to the server log when the presence of the `NO_AUTO_CREATE_USER` value for the `sql_mode` option in the options file prevents a MySQL 8.0 server from starting. (Bug #28061945, Bug #90967)
- **Microsoft Windows:** On Windows, uninstallation of the MySQL Server MSI package through MySQL Installer produced a spurious popup window. (Bug #27463864)
- **Microsoft Windows:** On Windows, `DBUG_ABORT` did not print the custom stack trace and other information. (Bug #21383530)
- **Microsoft Windows:** Starting MySQL as a Windows service with the service-installation command that specified a service name following the `--install` option ignored the directives in the named service group of the `my.ini` or `my.cnf` options files and used default options instead. Only the default service names (`mysqld`, `mysql_cluster`, `server`, `mysqld-8.0`) could load different parameters from an options file. (Bug #90383, Bug #27852209)
- **JSON:** The server did not reject creation of a table with a generated column in which the generated column used `JSON_TABLE()`, even though subqueries, parameters, variables, stored functions, and loadable functions are not permitted in expressions for generated columns. The server now checks more aggressively to make sure any of the disallowed constructs (including `JSON_TABLE()`) are rejected for use in such expressions. (Bug #28518485)
- **JSON:** `SELECT ... FROM JSON_TABLE()` sometimes failed with a permissions error for a user other than MySQL `root`. This issue could also occur when such a query was used as the basis for a view, and a `SELECT` from the view failed. (Bug #28255453, Bug #27923406)

  References: See also: Bug #27189940.
- **JSON:** The `JSON_TABLE()` function subjected integer values greater than or equal to $2^{31}$ to wraparound. For example, the query `SELECT id FROM JSON_TABLE('[{"id":"2147483648"}]', '$[*]' COLUMNS (id BIGINT UNSIGNED PATH '$.id')) AS json` returned `-2147483648`. (Bug #27856835)
- **JSON:** In some contexts, the `NULLIF()` function returned its first argument as a boolean value rather than its actual type. This was noticed when the result of this function was used as an argument to `JSON_ARRAYAGG()` or `JSON_OBJECTAGG()`, but could have occurred in other cases in which `NULLIF()` was used in a similar fashion. (Bug #90833, Bug #28007237)
- **JSON:** When a `JSON` document which contained binary data was converted to base-64 encoded text for display, newline characters in the encoded string were not properly escaped, so that the text representation could not be parsed as `JSON`, and was thus truncated, corrupted, or both. Now MySQL makes sure that any newline characters in the encoded string are escaped. (Bug #90503, Bug #27891359)
- `filesort` operations could cause a server exit. (Bug #28791531, Bug #92777)
- Debian packaging now supports Ubuntu 18.10 (Cosmic Cuttlefish). (Bug #28765706)

- On the Fedora 29 platform, OpenSSL 1.0.x is used to build packages because OpenSSL 1.1.1 support is not ready. If you build MySQL from source, it is recommended that you build using the `compat-openssl10-devel` package. (Bug #28737143)
- On the Fedora 29 platform, upgrading from MariaDB to MySQL 8.0.13 failed due to missing obsoletes. (Bug #28727698)
- MySQL binary distributions for SLES 12 now are built using GCC 7. The lowest supported GCC version on this platform is now 5.3 (previously 4.8.5). (Bug #28542723)
- Trying to add a functional index on a subquery should not be possible, and caused the server to hit an assertion when trying to resolve the indexed expression. Now in such cases, the expression is disallowed, and the servers returns an appropriate error message. (Bug #28526493)
- Added a range check when performing calculations with exponents. (Bug #28505423)
- `CMAKE -DWITHOUT_SERVER=1` resulted in build errors. (Bug #28501563, Bug #92011)
- When a prefix index was specified with a length of 8 bytes, the `Sub_part` column in the output of `SHOW INDEXES` was `NULL`. (Bug #28499603)
- Compilation failed for GCC 8 with MySQL configured to use some system libraries. (Bug #28471072, Bug #91914)
- For debug builds, if the server bootstrapping phase failed, missing cleanup code caused an assertion to be raised. (Bug #28435378, Bug #91847)
- Numeric ranges in MySQL builds could differ between the ARM and Intel x64 platforms. (Bug #28401869)
- For `mysqldump --tables` output, file names now always include a `.txt` or `.sql` suffix, even for file names that already contain a dot. Thanks to Facebook for the contribution. (Bug #28380961, Bug #91745)
- Concurrent `INSERT` and `SELECT` statements on a `MERGE` table could result in a server exit. (Bug #28379285)
- On SLES 15, upgrading from MariaDB packages to MySQL packages failed due to incorrect `obsoletes` information. (Bug #28292138)
- The data dictionary auto releaser now allocates maps for data dictionary object types dynamically. Also, maps were added for column statistics and resource groups used in size calculations and object removal, and an auto releaser was added for `ANALYZE TABLE` operations. (Bug #28245522, Bug #91420)
- It was possible for a `UNION` query that mixed different character sets to fail with a spurious error. (Bug #28237675)

  References: This issue is a regression of: Bug #83895, Bug #25123839.
- `CHECK TABLE ... FOR UPGRADE` on temporary tables could raise an assertion. (Bug #28220374)

  References: This issue is a regression of: Bug #24741307.
- A divide-by-zero error could occur in the range optimizer. (Bug #28214186)
- Due to a GCC 8 bug, recursion in foreign key checks could exhaust stack space. (Bug #28200774, Bug #28421040, Bug #91823)

- When `sql_mode` was set to the empty string, queries of the form `SELECT CONCAT( FORMAT(LPAD(char,2,''), 1) )` were not handled properly; the same was also true for similarly-formed queries using `RPAD()` in place of `LPAD()`. (Bug #28197977)

- Non-privileged users could change their own account password history and reuse properties. (Bug #28191838)

- The OpenSSL 1.1 `OPENSSL_malloc_init()` call did not work well on Windows. To address this, alternative allocation wrappers are used instead of the defaults from OpenSSL. Consequently, OpenSSL memory instrumentation is turned off. (Bug #28179051)

- Allocation of certain object types during data dictionary lookups resulted in a stack buffer overflow. (Bug #28176453)

- An internal server operation that attempted to perform a commit while fetching table statistics from the `INFORMATION_SCHEMA` could raise an assertion. (Bug #28165060)

- Certain stored procedures could cause a server exit. (Bug #28156802)

- Output for `CREATE USER` statements could differ in the general query log and audit log. (Bug #28147710)

- For `UPDATE` and `DELETE` statements that produce an error due to `sql_safe_updates` being enabled, the error message was insufficiently informative. The message now includes the first diagnostic that was produced, to provide information about the reason for failure. For example, the message may indicate that the `range_optimizer_max_mem_size` value was exceeded or type conversion occurred, either of which can preclude use of an index.

  Additionally: (1) Using `EXPLAIN` for such statements does not produce an error, enabling users to see from `EXPLAIN` plus `SHOW WARNINGS` output why an index is not used. (2) For multiple-table deletes and updates, an error is produced with safe updates enabled only if any target table uses a table scan. (Bug #28145710, Bug #91080)

- MySQL Server and test RPM packages were missing `perl-Data-Dumper` as a dependency. (Bug #28144933, Bug #72926)

- Server startup could fail if a service needed by a component was not yet initialized. (Bug #28142250)

- `SHOW CREATE TABLE` could omit foreign key `RESTRICT` options. This in turn could cause foreign key `RESTRICT` options to be lost from tables dumped with `mysqldump` and restored from the dump file. (Bug #28122781, Bug #91110)

- The `mysql` client was slow to import large multiple-line statements in batch mode. Memory allocation is now more efficient for this situation. Thanks to Sinisa Milivojevic for the patch. (Bug #28116512, Bug #85155)

- The `SUM()` and `AVG()` functions did not handle correctly a string argument used with a window function. (Bug #28105241)

- Excessive nesting of geometry collections caused the server to exhaust stack space. The server now raises an error if there is a danger of stack overrun. (Bug #28100563)

- For the `mysql` client, the `-b` short option was associated with two long options, `--no-beep` and `--binary-as-hex`. The `-b` option now is associated only with `--no-beep`. (Bug #28093271)

- For a table with an auto-increment primary key, concurrent `ALTER TABLE ... ADD ... VIRTUAL` and `INSERT` statements could lead to duplicate-key errors. (Bug #28089240)

- Handling of floating-point values by `SUM()` was improved. (Bug #28080199)
- Some expressions using `ST_Simplify()` could cause a server exit. (Bug #28079969)
- In builds with Undefined Behavior Sanitizer enabled, testing returned a “member call on null pointer of type 'struct Event_db_repository'” error. (Bug #28066155)
- The `WITH_GMOCK` `CMake` option did not handle Windows path names properly. (Bug #28061409, Bug #90964)
- `INFORMATION_SCHEMA` queries that attempted to cache table statistics could raise a debugging assertion. (Bug #28035207)
- If roles were assigned to accounts, certain `sql_mode` settings could cause the server to be unable to start. (Bug #28030423)
- Group lookups for LDAP authentication plugins could fail if the user had insufficient privileges. Now, group search operations bind again using `root` credentials if those are available. (Bug #28016008)
- `ANALYZE TABLE ... UPDATE HISTOGRAM` statements produced by `mysqldump` contained a syntax error. (Bug #28014376, Bug #90846)
- Generated columns having indexes and that used a string function were not always populated correctly. (Bug #27973409)
- Fixed-length sort keys, such as those used by priority queues, that fit exactly were assumed to have failed. (Bug #27970481, Bug #92448, Bug #28654343)
- For an empty result, `REPLACE()` sometimes returned a null string rather than an empty string. (Bug #27960921)
- Dropping a table that was created with a user-defined collation no longer available could cause a server exit. (Bug #27952999)
- Joining the `INFORMATION_SCHEMA` `REFERENTIAL_CONSTRAINTS` and `TABLE_CONSTRAINTS` tables on the `CONSTRAINT_NAME` failed because the column collations differed. (Bug #27945704, Bug #90690)
- Some operations on `DECIMAL` values could cause a server exit. (Bug #27942277)
- Very long table keys were handled incorrectly on replication slaves. (Bug #27930505)
- `mysqld` did not determine its installation directory correctly if that directory was the last one listed in `$PATH`. (Bug #27922896)
- During server startup/shutdown, PID files could be mishandled. (Bug #27919254)
- When the server was started with `--skip-grant-tables`, some account-management statements were not disabled. (Bug #27906226)
- On rare occasions, setting a savepoint could raise an assertion. (Bug #27898591)
- Runtime errors could occur for calls to `copy_integer()`. `myfunc_int()`, `mysql_sys_var_int()`, or `thd_killed()`; for calls to `MyISAM` sorting functions; or for values outside the range of representable values of type `unsigned long long`. (Bug #27894901, Bug #90515, Bug #27871951, Bug #27918095, Bug #90609, Bug #27937522, Bug #90661, Bug #27978325, Bug #27962900)
- `mysql_install_plugin` failed to report plugin-specific errors if the server was unable to store plugin metadata in the data dictionary. (Bug #27893406)

- `ssl_fips_mode` cannot be set to `ON` unless the OpenSSL requirements are satisfied, but a failed attempt still resulted in the value being displayed as `ON`. (Bug #27891890)
- The symbol for the `mysql_result_metadata()` C API function was not exported by the client library on Windows. (Bug #27868095)
- For `InnoDB` tables, self-referencing foreign keys could be created, causing server misbehavior. (Bug #27864515)
- A `UNION` query mixing different character sets produced invalid output when performed in a prepared statement or a stored procedure. (Bug #27849293)
- Debian packages were missing a dependency for `libcurl-dev`. (Bug #27844465)
- `--help` output for client programs did not include the current `--ssl-fips-mode` value. (Bug #27838966)
- A runtime error could occur for calls to `Derived_key_comp()`. (Bug #27830679)
- For `MEMORY` tables, memory overflow errors could occur. (Bug #27799513)
- When converting from a `BLOB` (or `TEXT`) type to a smaller `BLOB` (or `TEXT`) type, no warning or error was reported informing about the truncation or data loss. Now an appropriate error is issued in strict SQL mode and a warning in nonstrict SQL mode. (Bug #27788685, Bug #90266)
- For debug builds, `my_strnxfrm_unicode_full_bin()` could raise an assertion. (Bug #27752619)
- In builds with Undefined Behavior Sanitizer enabled, signed integer overflow could occur in GIS calculations. (Bug #27751479, Bug #27744399, Bug #27811282)
- It was possible in some cases for the internal function `regexp::EvalExprToCharset()` to bind a reference to a misaligned address. (Bug #27743722)
- The severity of messages produced by the server about being unable to read key files has been escalated from INFORMATION to WARNING. (Bug #27737195)
- No temporary table field was created for an expression using window functions combined with a rollup, causing queries containing these to fail. (Bug #27735167)
- The expression `FIND_IN_SET( JSON_UNQUOTE( JSON_SET('{}','$','') ), 1)` was not evaluated properly. (Bug #27731699)
- Failure to create a temporary table during a `MyISAM` query could cause a server exit. Thanks to Facebook for the patch. (Bug #27724519, Bug #90145)
- `parser_max_mem_size` was ineffective when parsing stored program definitions. (Bug #27714748)
- A `BETWEEN` clause comparing negative values could lead to erroneous results. (Bug #27691347)
- Some typos in server error messages were fixed. Thanks to Thomas Tsiakalakis for the contribution. (Bug #27688294, Bug #90048)
- These scripts are no longer included in RPM packages (they are unnecessary because they are compiled into the `mysqld` binary): `fill_help_tables.sql`, `mysql_sys_schema.sql`, `mysql_system_tables.sql`, `mysql_system_tables_data.sql`, `mysql_system_users.sql`. (Bug #27672991)
- `Unique::io_cache` was not cleaned up properly when restarting an index merge operation, leading to an assertion in debug builds. (Bug #27599292)

- Using a binary or hexadecimal literal value in a `ENUM` column resulted in a string type with bytes not permitted by the data dictionary character set. (Bug #27592803)
- It was possible for `CONVERT()` to fail under the following conditions:

  The size of the result of the conversion was greater than `max_allowed_packet`

  `sql_mode` was '' (that is, empty).

  The character set of the result differed from `character_set_connection` (Bug #27592714)
- Use of `ENUM` or `SET` fields with `JSON_TABLE()` could result in unexpected behavior. (Bug #27571251)
- Host name resolution errors could cause the `audit_log` plugin to fail. (Bug #27567003)
- Unsuccessful connection attempts were not being written to the error log when `log_error_verbosity=3`. (Bug #27539838)
- Persisted system variables with a value of `NULL` were written to `mysqld-auto.cnf` incorrectly. (Bug #27512616)
- An earlier code cleanup caused `FEDERATED` storage engine failures. (Bug #27493633, Bug #89537)

  References: This issue is a regression of: Bug #25943754.
- Setting `max_execution_time` sometimes had no effect when used with full-text search. (Bug #27155294)
- An attempted read of an uncommitted transaction raised an assertion. (Bug #26876608)
- `ALTER TABLE ... REORGANIZE PARTITION ...` could result in incorrect behavior if any partition other than the last was missing the `VALUES LESS THAN` part of the syntax. (Bug #26791931)
- In some cases when `ER_DATA_TOO_LONG` was the appropriate error, `ER_TRUNCATED_WRONG_VALUE_FOR_FIELD` was reported instead. (Bug #26475237, Bug #87100)
- A `last_checked_for_upgrade` column that stores the current server version number was added to the `mysql.tables` data dictionary table to track successful `CHECK TABLE ... FOR UPGRADE` operations. This data was previously stored in `.frm` files to avoid rechecking tables unnecessarily. (Bug #24741307, Bug #83169)
- The `MYSQLparse()` function generated from the `sql_yacc.yy` file had exceeded a size limit that caused the Microsoft Visual C++ compiler to abandon optimization. The `/d2OptimizeHugeFunctions` compiler switch was enabled to restore optimization (available in Visual Studio 2017 and higher). (Bug #24600301)
- A range check for the product of a signed and unsigned integer could be performed incorrectly. (Bug #17081376, Bug #69715)
- Specifying the maximum possible value for a `YEAR` column failed when expressed as a real constant such as `2155.0E00` or `2.15E3`. (Bug #91226, Bug #28172538)
- Prevented potential member access within a null pointer of type `SELECT_LEX`. (Bug #90689, Bug #27945614)
- The base used for calculation of line numbers reported in warnings generated by `SELECT` statements has been changed from the number of input rows read to the actual row numbers seen by the user. For statements using `GROUP BY`, this number is now based on the number of aggregate groupings. (Bug #90667, Bug #27938507)

- Estimates contained by histograms for the number of distinct values in each bucket, which can be employed to estimate the number of distinct values in a table, can now be used by the Optimizer's `histograms::Histogram` interface. (Bug #90465, Bug #27872281)
- Selecting from the `INFORMATION_SCHEMA.KEYWORDS` table failed when no default database had been selected. (Bug #90160, Bug #27729859)
- `ANALYZE TABLE UPDATE HISTOGRAM` failed to return an error condition to the client when `lock_wait_timeout` was exceeded during statement processing. (Bug #89738, Bug #27574375)
- It was possible to create a table `t` having a generated column `g` and then to execute `ALTER TABLE t ALTER COLUMN g SET DEFAULT ...`; attempting to set the default for a generated column in this fashion now fails with Error 1221 `Incorrect usage of DEFAULT and generated column`. This is the same behavior as for a similar `CREATE TABLE` statement. (Bug #88619, Bug #27164393)

# Changes in MySQL 8.0.12 (2018-07-27)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.


- Audit Log Notes
- Compilation Notes
- Configuration Notes
- Data Dictionary Notes
- Deprecation and Removal Notes
- Firewall Notes
- SQL Function and Operator Notes
- Logging Notes
- Optimizer Notes
- Packaging Notes
- Plugin Notes
- Security Notes
- Spatial Data Support
- SQL Syntax Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Audit Log Notes

- For new MySQL installations, the `USER` and `HOST` columns in the `audit_log_user` table used by MySQL Enterprise Audit now have definitions that better correspond to the definitions of the `User` and `Host` columns in the `mysql.user` system table.

For upgrades to an installation for which MySQL Enterprise Audit is already installed, it is recommended that you alter the table definitions as follows:

```
ALTER TABLE mysql.audit_log_user
  DROP FOREIGN KEY audit_log_user_ibfk_1;
ALTER TABLE mysql.audit_log_filter
  CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_ci;
ALTER TABLE mysql.audit_log_user
  CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_as_ci;
ALTER TABLE mysql.audit_log_user
  MODIFY COLUMN USER VARCHAR(32);
ALTER TABLE mysql.audit_log_user
  ADD FOREIGN KEY (FILTERNAME) REFERENCES mysql.audit_log_filter(NAME);
```

(Bug #23706056)

## Compilation Notes

- **Microsoft Windows:** For OpenSSL 1.1, the library names have changed for Windows. The `CMake` configuration code now accounts for this. Also, it is now possible on Windows to use `-DWITHOUT_SERVER=1` to build 32-bit client binaries. (Bug #28170711, Bug #91223)
- The minimum version of the Boost library for server builds is now 1.67.0. (Bug #27866110)

## Configuration Notes

- The maximum value was increased from 256 to 1024 for these Performance Schema system variables:

  ```
  performance_schema_max_cond_classes
  performance_schema_max_file_classes
  performance_schema_max_mutex_classes
  performance_schema_max_rwlock_classes
  performance_schema_max_socket_classes
  performance_schema_max_stage_classes
  performance_schema_max_thread_classes
  ```

  The default value was increased from 250 to 300 for this Performance Schema system variable:

  ```
  performance_schema_max_mutex_classes
  ```

  (Bug #27647918)

- For `RelWithDebInfo` builds on Linux, `REPRODUCIBLE_BUILD` now defaults to `ON`. (Bug #27483447)
- The default value for the `slave_pending_jobs_size_max` system variable has increased from 16M to 128M. On a multi-threaded slave ( with `slave_parallel_workers > 0`), this value sets the maximum amount of memory that is made available to slave worker queues holding events that have not yet been applied. The potentially larger memory usage should be taken into account when calculating the resources required to deploy a multi-threaded slave. Masters and single-threaded slaves are not affected by this setting.

  The new default is double the recently increased default value specified by the `max_allowed_packet` system variable (now 64M). The value of `slave_pending_jobs_size_max` is a soft limit, so events larger than its value (consisting of one or more packets) can be queued and processed. However, a large transaction is held until all the slave workers have empty queues, and then processed. All subsequent transactions are held until the large transaction has been completed. The delay to clear the queues of all the slave workers and the wait to queue subsequent transactions can cause lag on the replication slave and decreased concurrency of the slave workers. The value of

`slave_pending_jobs_size_max` should therefore be set high enough to accommodate most expected event sizes. (WL #11348)

## Data Dictionary Notes

- The following data dictionary enhancements were introduced:
  - An internal method was added to protect against global read locks. The method is called before storing, updating, or dropping data dictionary objects.
  - Assertion code was added to the data dictionary API to protect against global read locks when storing, updating, or dropping data dictionary objects.
  - Protection is now implicitly acquired against backup locks and global read locks when an exclusive metadata lock is acquired by a storage engine.
  - A cost model cache validity check is now performed before releasing a cost model entry.
  - A data dictionary API function was added for committing and rolling back transactions, and for releasing transactional meta data locks.

  (Bug #27937059)

## Deprecation and Removal Notes

- Explicit `ASC` or `DESC` qualifiers for `GROUP BY` clauses are now deprecated and will be removed in a future MySQL version. (WL #11459)

## Firewall Notes

- The MySQL Enterprise Firewall `firewall_whitelist` table now contains a primary key column named `ID`. (Bug #27164826)

## SQL Function and Operator Notes

- The `BIT_AND()`, `BIT_OR()`, and `BIT_XOR()` aggregate functions now can be used as window functions, if an `OVER` clause is present. See Aggregate Function Descriptions. (WL #11573)

## Logging Notes

- Previously, successful assignments to the `dragnet.log_error_filter_rules` system variable resulted in a Note and a nonzero warning count. To avoid generating a diagnostic for a successful operation, this Note is no longer generated. Instead, the `dragnet.Status` status variable can be consulted to determine the result of the most recent assignment to `dragnet.log_error_filter_rules`. (Bug #27910708, Bug #90571)
- Previously, log components listed in the `log_error_services` system variable had to be separated by semicolons. Components now can be separated by commas. A given setting cannot use both semicolon and comma separators. (Bug #27788925, Bug #90268)

## Optimizer Notes

- The `filesort` algorithm (used by the optimizer to satisfy an `ORDER BY` clause when no index can be used) is now more memory efficient. The primary change is that the optimizer allocates memory buffers incrementally as needed, up to the size indicated by the `sort_buffer_size` system variable,

rather than allocating a fixed amount of `sort_buffer_size` bytes up front. This enables users to set `sort_buffer_size` to larger values to speed up larger sorts, without concern for excessive memory use for small sorts. (This benefit may not occur for multiple concurrent sorts on Windows, which has a weak multithreaded `malloc`.)

In addition, `filesort` is less pessimistic with respect to worst-case scenarios. Previously, `filesort` produced an error without starting if merge operations or sort-key generation might fail. Now, because the worst case is unlikely to occur in practice, `filesort` produces an error only for actual rather than prospective failures. (WL #11590)

- Previously, extended `EXPLAIN` output was produced only for `SELECT` statements. Extended output is now produced for `INSERT`, `REPLACE`, `UPDATE`, and `DELETE` statements as well. (To display extended output, use `SHOW WARNINGS` following `EXPLAIN`. See Extended EXPLAIN Output Format.) (WL #6033)

## Packaging Notes

- For Linux, the generic binary distribution compression algorithm changed from Gzip to XZ; and the file extension changed from .tar.gz to .tar.xz. (Bug #28450941, Bug #91889)
- Docker disables NUMA-related operations by default, so the `mysqld` included in Docker images now has NUMA support disabled so that the error log does not fill with NUMA warnings. (Bug #28081363)
- RPM `.spec` files were updated to reflect the removal of yaSSL and that OpenSSL is the default SSL library for all builds. (Bug #28025427)
- For generic Linux `.tar.gz` distributions, the `mysqlxtest` client has been moved from the server package to the test package. (Bug #27744137)
- An RPM package for installing ARM 64-bit (aarch64) binaries of MySQL Server on Oracle Linux 7 is now available in the MySQL Yum Repository and for direct download.

  **Known Limitation for this ARM release**: You must enable the Oracle Linux 7 Software Collections Repository (ol7_software_collections) to install this package, and must also adjust the libstdc++7 path. See Yum's Platform Specific Notes for additional details.

## Plugin Notes

- Previously, the `Rewriter` query rewrite plugin permitted rewriting only for `SELECT` statements. Rewritable statements now include `INSERT`, `REPLACE`, `UPDATE`, and `DELETE` as well. See The Rewriter Query Rewrite Plugin. (WL #8776)

## Security Notes

- The linked OpenSSL library for MySQL Server has been updated to version 1.0.2o. Issues fixed in the new OpenSSL version are described at http://www.openssl.org/news/vulnerabilities.html. (Bug #28025379)

## Spatial Data Support

- **Incompatible Change:** The new spatial functions `ST_Longitude()` and `ST_Latitude()` return the longitude or latitude, respectively, of their `Point` argument. If called with a second argument, these functions use the argument to set the longitude or latitude, respectively, of their `Point` argument.

  In addition, the `ST_X()` and `ST_Y()` functions now interpret their `Point` argument according to its spatial reference system definition (SRS):

- `ST_X()` returns the coordinate value of the axis that appears first in the SRS definition, and `ST_Y()` returns the coordinate value of the axis that appears second in the SRS definition.
- If `ST_X()` and `ST_Y()` are used to set the longitude or latitude of a `Point` argument that has a geographic SRS, the longitude or latitude value must be in the proper range for the SRS, or an error occurs.

**Note**

If spatial data contains geometry values that now are interpreted differently by `ST_X()` and `ST_Y()`, existing queries using these functions will return different results, compared to previous MySQL versions.

(Bug #27125600, Bug #88503, WL #11453)

- The EPSG data set containing spatial reference system data for spatial calculations has been upgraded from version 9.2 to 9.3. (Bug #27977721)
- Spatial functions that accept geometry arguments and previously returned an `ER_LONGITUDE_OUT_OF_RANGE` or `ER_LATITUDE_OUT_OF_RANGE` error for longitude or latitude values out of range now return `ER_GEOMETRY_PARAM_LONGITUDE_OUT_OF_RANGE` or `ER_GEOMETRY_PARAM_LATITUDE_OUT_OF_RANGE` instead. This does not apply to spatial functions that parse or modify geometries: `ST_`*`xxx`*`FromText()`, `ST_`*`xxx`*`FromWKT()`, `ST_`*`xxx`*`FromWKB()`, `ST_GeomFromGeoJSON()`, `ST_X()`, `ST_Y()`, `ST_Longitude()`, and `ST_Latitude()`. (Bug #27892138)
- Previously, the parser recognized `USING RTREE` in index specifications, but this could not be specified for any storage engine. Now `USING RTREE` is permitted for spatial index specifications. (Bug #27836608)
- Indexes on geometry columns now must be spatial indexes. Consequently, the `SPATIAL` keyword is optional but implicit for creating an index on a spatial column. A spatial index on a geometry column is not permitted for a primary key or unique index. Also, `ASC` and `DESC` are not permitted for spatial indexes. (Bug #21087676, WL #11808)

## SQL Syntax Notes

- **Important Change:** MySQL now allows a query having a `WITH ROLLUP` modifier to use `DISTINCT`. See SELECT Statement, for more information. (Bug #87450, Bug #26640100, Bug #20671578, Bug #76229)
- **Important Change:** MySQL now allows the use of `ORDER BY` with grouping functions in `SELECT`. statements. (Bug #86312, Bug #27063525)
- **Important Change:** MySQL now allows `ORDER BY` to be used in a query having a `WITH ROLLUP` modifier. For more information, see SELECT Statement. (Bug #86311, Bug #26073513)

## X Plugin Notes

- Authentication errors returned from a MySQL 8.0.11 server to an X Plugin client were marked as fatal, which caused the client to stop attempting its automatic sequence of authentication attempts using different methods. X Plugin now ignores fatal authentication errors, and subsequent expected errors such as write timeout errors, while the sequence of authentication attempts is in progress. If none of the authentication methods tried are successful, X Plugin now selects and returns the most relevant error message received from the server, which is not necessarily the last error message received. (Bug #28135006)

- During the X Plugin automated authentication sequence, connection attempts are made using different authentication methods in a pre-set sequence. X Plugin checks before starting the sequence whether the connection uses SSL or not, and does not attempt connection with the `PLAIN` authentication method if the connection is not encrypted. For additional verification, a check and error were added so that the `PLAIN` authentication method is disallowed if attempted when the connection is not encrypted. (Bug #27691189)

- In the event of an authentication failure, X Plugin always returned the error code `ER_ACCESS_DENIED`, with a custom X Protocol error message for the error situation. The error codes that are received from the MySQL server for authentication failures are now passed on to the client instead, with the standard MySQL error messages that go with them. (Bug #27675699)

- During the X Plugin automated authentication sequence, where connection attempts are made using different authentication methods in a pre-set sequence, the error returned after every failed connection attempt was marked with the severity `FATAL`. Now, the errors are marked as `ERROR` while there are still other authentication methods to be attempted, and only the final error that results in disconnection is marked as `FATAL`. (Bug #27636947)

- The X Plugin code was being compiled twice with different preprocessor definitions. (Bug #27267054)

- A number of improvements were made to the handling of authentication issues by X Plugin:

  - The X Plugin automated authentication sequence, where connection attempts are made using different authentication methods in a pre-set sequence, is now halted if an error response indicates that the connection has been broken or reads or writes are timing out.

  - The automated authentication sequence is also now halted in the event of any fatal error.

  - The last significant error is now reported for an abandoned authentication sequence, not just the last error that occurred.

  - A more specific error code and message are now reported if authentication with the `caching_sha2_password` authentication plugin is attempted over an unsecure connection.

  (Bug #27257774)

- In some circumstances, a MySQL server with X Plugin loaded could experience an unexpected halt during a shutdown process initiated by a failed start or by the `RESTART` command. The issue has now been fixed. (Bug #25474793, Bug #27259783)

## Functionality Added or Changed

- **Important Change; Group Replication:** `group_replication_exit_state_action` determines how Group Replication behaves when a server instance leaves the group involuntarily, such as when it is expelled from the group due to an unstable network connection. When `group_replication_exit_state_action` is `ABORT_SERVER` (the default), the instance shuts itself down; when its value is `READ_ONLY`, the instance switches itself to super read-only mode instead, and goes into the Group Replication `ERROR` state. (WL #11568)

- **InnoDB:** `InnoDB` now supports `ALGORITHM=INSTANT` for the following `ALTER TABLE` operations:

  - Adding a column. This feature is referred to as “Instant `ADD COLUMN`”.

  - Adding or dropping a virtual column.

  - Adding or dropping a column default value.

- Modifying the definition of an `ENUM` or `SET` column.
- Changing the index type.
- Renaming a table.

Operations that support `ALGORITHM=INSTANT` only modify metadata in the data dictionary. No exclusive metadata locks are taken on the table during preparation and execution phases of the operation, and table data is unaffected, making the operations instantaneous. If not specified explicitly, `ALGORITHM=INSTANT` is used by default by operations that support it. If `ALGORITHM=INSTANT` is specified but not supported, the operation fails immediately with an error.

For more information about operations that support `ALGORITHM=INSTANT`, see Online DDL Operations.

Thanks to the Tencent Games DBA team for the contribution. (Bug #28100103, Bug #91074, WL #11250)

- **InnoDB:** Undo logging is now supported for small updates to large object (LOB) data, which improves performance of LOB updates that are 100 bytes in size or less. Previously, LOB updates were a minimum of one LOB page in size, which is less than optimal for updates that might only modify a few bytes. This enhancement builds upon support added in MySQL 8.0.4 for partial update of LOB data. (WL #11328)

- **Replication:** You can now use a shell pipe to supply an archive of compressed binary log files to `mysqlbinlog` as streamed input. Previously, the files had to be extracted separately before passing them to `mysqlbinlog`. Thanks to the Facebook team for this contribution.

  In this example, `binlog-files_1.gz` contains multiple binary log files for processing. The pipeline extracts the contents of `binlog-files_1.gz`, pipes the binary log files to `mysqlbinlog` as standard input, and pipes the output of `mysqlbinlog` into the `mysql` client for execution:

  ```
  gzip -cd binlog-files_1.gz | ./mysqlbinlog - | ./mysql -uroot -p
  ```

  (Bug #27445278, Bug #89423)

  References: See also: Bug #27836543.

- **Replication:** The `SYSTEM_VARIABLES_ADMIN` and `PERSIST_RO_VARIABLES_ADMIN` privileges have been added to the `mysql.session` reserved account. (Bug #89873, Bug #27621869)

- **Replication:** Information about XCom's memory usage has been added to the Performance Schema tables. The `setup_instruments` table now has an entry corresponding to the XCom cache, and the `memory_summary_global_by_event_name` table contains the memory usage statistics of the XCom cache. (WL #9855)

- **Microsoft Windows:** On Windows, the forking used to implement `RESTART` was suppressible using the `--gdb` option. However, this was a side effect, performed in addition to the other option actions done to set up a debugging environment. In non-debug settings, the new `--no-monitor` option may be used for the sole purpose of suppressing forking the monitor process. For a server started with either `--gdb` or `--no-monitor`, executing `RESTART` causes the server to simply exit without restarting. (Bug #27801043)

- Checking for foreign key relationships by `DROP TABLE` and `DROP DATABASE` was improved. The check now takes place before any tables are deleted, so that in the event of a problem the statement fails before making any changes. Parent and child tables now can be dropped in arbitrary order, as long as they are dropped by the same `DROP TABLE` statement. In addition, error reporting was improved for

attempts to drop a parent table without dropping a child table. (Bug #27821060, Bug #17564464, Bug #70531, Bug #22359539, Bug #79610)

- When support for roles was added, the `ADMIN` keyword became a reserved keyword. `ADMIN` is now once again a nonreserved keyword. (Bug #27814204)
- When a client shuts down the server, the server now writes a message to the error log indicating which user performed this action. (Bug #26246628, Bug #86635)
- Previously, for the `--ssl-mode=VERIFY_IDENTITY` option, the client checked whether the host name that it used for connecting matched the Common Name value in the certificate but not the Subject Alternative Name value. Now, if the client uses OpenSSL 1.0.2 or higher, the client checks whether the host name matches either the Subject Alternative Name value or the Common Name value in the server certificate. Thanks to Daniël van Eeden for a patch on which this change was based. (Bug #16211011, Bug #68052, Bug #27511233, Bug #89578)

## Bugs Fixed

- **Important Change; JSON:** The `JSON_TABLE()` function now automatically decodes base-64 values and prints them using the character set given by the column specification. (Bug #90157, Bug #27729112)

  References: See also: Bug #89847, Bug #27613276.

- **InnoDB:** New columns added to `INFORMATION_SCHEMA.INNODB_TABLES` and `INFORMATION_SCHEMA.INNODB_COLUMNS` tables in 8.0.12 did not appear after upgrading to 8.0.12. (Bug #28065244)
- **InnoDB:** Large object (LOB) version numbers stored in LOB index entries were left uninitialized at 0 instead of initialized at 1. (Bug #28046298)
- **InnoDB:** Adding a column to a table with a generated column raised an invalid assertion. (Bug #28040201)
- **InnoDB:** An attempt to access freed memory during a checkpoint operation caused the server to exit. (Bug #28039477)
- **InnoDB:** An assertion failure occurred when starting the server with experimental system variables that were unintentionally exposed in normal MySQL builds. (Bug #27899424, Bug #90526, Bug #27898396, Bug #90523, Bug #27898344, Bug #90522)
- **InnoDB:** An invalid system variable setting raised an assertion. (Bug #27898284)
- **InnoDB:** Attempting to move a table from a discarded file-per-table tablespace to a shared tablespace raised an assertion. The operation now produces an error instead. (Bug #27861972)
- **InnoDB:** Potential failures related to page allocation for large objects (LOBs) were not sufficiently handled. (Bug #27852003)
- **InnoDB:** The ngram full-text search parser permitted comma and period characters to be tokenized as words, which caused an inconsistency between boolean and natural language mode search results. Comma and period characters are no longer tokenized. (Bug #27847697)
- **InnoDB:** Checksum mismatches and other issues were caused by concurrent modification of a tablespace header page, and attempting to place a lock on a partially initialized page. (Bug #27823064)
- **InnoDB:** An I/O error returned by an `fsync()` operation is now treated as a hard error. (Bug #27805553, Bug #90296)

- **InnoDB:** In a single thread, two minitransactions latched the first page of a large object value (LOB) while another thread attempted to allocate the page, causing a deadlock. (Bug #27777959)
- **InnoDB:** After upgrading to MySQL 8.0.12, table columns including the FTS_DOC_ID column were missing from the `INFORMATION_SCHEMA.INNODB_COLUMNS` table. (Bug #27774145)
- **InnoDB:** An invalid assertion was raised when the optimizer chose a prefix index defined on a virtual column. (Bug #27755892)

  References: This issue is a regression of: Bug #19806106.
- **InnoDB:** An in-memory table object for a table being truncated could be opened under improper metadata locking protection.

  A new method of preserving the `AUTOINC` counter value during a `TRUNCATE PARTITION` operation was implemented. (Bug #27754995)
- **InnoDB:** An assertion was raised during a mini-transaction commit that included dirty pages but no redo log records. The mini-transaction did not wait for space in the recently-closed buffer and subsequently marked the page with an invalid current LSN value. After the invalid LSN value was added to the flush list, a delayed mini-transaction added dirty pages with an earlier LSN value, breaking the flush list order.

  The `innodb_log_checkpoint_fuzzy_now` debug option was added. Enabling this option forces `InnoDB` to write a fuzzy checkpoint. Also, new internal redo log module counters were added for tracking LSN values. The `log_lsn_buf_dirty_pages_added` counter tracks the LSN value up to which dirty pages have been added, and the `log_lsn_buf_pool_oldest_lwm` counter tracks the low watermark LSN for the oldest modified block in the buffer pool. (Bug #27664539)
- **InnoDB:** The Contention-Aware Transaction Scheduling algorithm (CATS) tracks the number of transactions waiting for another transaction. In the case of a deadlock, the method by which the number of waiting transactions is approximated produced inflated values. (Bug #27646322)
- **InnoDB:** Attempting to acquire an `InnoDB` ticket while holding a victim transaction lock caused a lock order violation. (Bug #27626681)

  References: This issue is a regression of: Bug #23476050.
- **InnoDB:** An attempt to read a zero-length BLOB value raised an assertion failure during recovery. The BLOB value was not yet completely inserted. (Bug #27617389)
- **InnoDB:** A race condition occurred when acquiring a table lock for a table that was already in memory. (Bug #27586419, Bug #27577704)

  References: This issue is a regression of: Bug #26848711.
- **InnoDB:** After restarting the server, attempting to create an encrypted table returned a generic error from the storage engine. (Bug #27577339)
- **InnoDB:** The use of Contention-Aware Transaction Scheduling (CATS) with a spatial index caused a transaction to wait indefinitely. (Bug #27572937)
- **InnoDB:** Incorrect parsing of a tablespace name that included multibyte characters caused a `DROP DATABASE` operation to fail. (Bug #27566937)
- **InnoDB:** A check performed to identify temporary tables caused an error when the data dictionary attempted to open a partitioned table. (Bug #27565997)
- **InnoDB:** A schema mismatch error reported during an import tablespace operation failed to print mismatched table flags in a readable format. (Bug #27542720)

- **InnoDB:** A DDL operation failed to wait for a `FULLTEXT` index optimization operation to finish. (Bug #27326796)

  References: This issue is a regression of: Bug #24938374.

- **InnoDB:** An unnecessary check for read-only transactions was removed from the `trx_set_rw_mode()` function. Thanks to Sandeep Sethia for the patch. (Bug #27211287, Bug #88739)

- **InnoDB:** A DDL operation that added a foreign key constraint raised an assertion when it accessed a stale memory object that belonged to the parent table. (Bug #27208858)

- **InnoDB:** A failing assertion occurred after initiating a `memcached` get operation. (Bug #26876594)

- **InnoDB:** A corrupt index ID encountered during a foreign key check raised an assertion. (Bug #26654685)

- **InnoDB:** A `DROP TABLE` operation on a table residing in a general tablespace raised an assertion when checking index status in the post-DDL phase of the operation. (Bug #26523254)

- **InnoDB:** An internal deadlock during a DDL operation resulted in a long semaphore wait followed by a server exit. (Bug #26225783)

- **InnoDB:** A DDL operation encountered a serious error due to an invalid lock upgrade. (Bug #26225783)

- **InnoDB:** On a Windows 64-bit system, invalid buffer pool configuration values caused the server to exit on startup. (Bug #26100239, Bug #86370)

- **InnoDB:** A debug code block related to mutex type definitions was removed. It no longer differed from the non-debug version of the code block. (Bug #24952279, Bug #83529)

- **InnoDB:** A server failure during recovery occurred due to a log parsing buffer overflow. Space is now allocated to the log parsing buffer dynamically and incremented as necessary until it reaches the size specified by `innodb_log_buffer_size`. (Bug #24734190)

- **InnoDB:** Table locking was optimized for concurrent updates on the same table by avoiding costly iterations over the table locks queue.

  Thanks to Zhai Weixiang for the contribution. (Bug #18955152, Bug #72948)

- **Packaging:** On Windows, the `mysqld.exe --help` command displayed unnecessary and unexpected details in the system output, which this fix eliminates. (Bug #27894020)

- **Partitioning:** Rollback of `ALTER TABLE ... TRUNCATE PARTITION` statements was not handled correctly. (Bug #27603025)

  References: See also: Bug #87562, Bug #26710839.

- **Partitioning:** For a partitioned table, partition update time could be incorrect after rebuilding the table or restarting the server. (Bug #27073100)

- **Replication:** If the option `--skip-log-bin` was specified at server startup to disable binary logging, no warning message was issued if both `expire_logs_days` and `binlog_expire_logs_seconds` were set to a nonzero value. The `--skip-log-bin` option meant there were no binary log files to expire, so the outcome of these binary log expiration period settings (which is that the `expire_logs_days` value is ignored) had no practical effect. However, the warning message is now issued, so that the situation can be corrected before binary logging is enabled for the server. (Bug #27699608)

- **Replication:** The log messages generated when a member fails to join the group have been improved, for example when `group_replication_group_name` on the member joining the group does not match the seed's `group_replication_group_name` this is now described in the log message. (Bug #27628695)

- **Replication:** If a MySQL server is started in crash recovery mode with an `innodb_force_recovery` setting of 4 or greater, which places `InnoDB` into read-only mode, the semisynchronous replication master plugin should not be initialized. Due to a change in plugin initialization order in MySQL 8.0, the server assumed that if the plugin was registered, it had also been initialized, and incorrectly attempted to deinitialize it. The handling of this plugin in crash recovery mode has now been corrected so that the deinitialization function is not executed when the plugin has not been initialized. (Bug #27481872)

- **Replication:** The use of replication filters or binary log filters can cause issues when they are applied to tables that are updated with XA transactions. Filtering of tables could cause an XA transaction to be empty on a replication slave, and empty XA transactions are not supported. Also, with the settings `master_info_repository=TABLE` and `relay_log_info_repository=TABLE` on a replication slave, which became the defaults in MySQL 8.0, the internal state of the data engine transaction is changed following a filtered XA transaction, and can become inconsistent with the replication transaction context state.

  Due to these issues, the use of replication filters or binary log filters in combination with XA transactions is not supported. This fix adds the new error `ER_XA_REPLICATION_FILTERS`, which is logged whenever an XA transaction is impacted by a replication filter, whether or not the transaction was empty as a result. If the transaction is not empty, the replication slave is able to continue running, but you should take steps to discontinue the use of replication filters with XA transactions in order to avoid potential issues. If the transaction is empty, the replication slave stops. In that event, the replication slave might be in an undetermined state in which the consistency of the replication process might be compromised. In particular, the `gtid_executed` set on a slave of the slave might be inconsistent with that on the master. To resolve this situation, isolate the master and stop all replication, then check GTID consistency across the replication topology. Undo the XA transaction that generated the error message, then restart replication. (Bug #27442477)

- **Replication:** When `mysqlbinlog` was used to read a binary log event larger than 128KB, the read failed because 128KB was returned as the size, instead of the actual size. The issue has now been fixed. Thanks to the Facebook team for the patch. (Bug #27417084, Bug #89326)

- **Replication:** When a transaction larger than the binary log transaction cache size (`binlog_cache_size`) was flushed to a temporary file during processing, and the flush failed due to a lack of space in the temporary directory, the flush error was not handled correctly. No message was written to the error log, and the binary log cache was not cleared after the transaction was rolled back. Now, in this situation, the server takes an appropriate action based on the `binlog_error_action` setting (shut down the server or halt logging), and writes a message to the error log. When the transaction is rolled back, the server checks for flush errors and clears the binary log cache if any occurred. (Bug #27399620, Bug #89272)

- **Replication:** When GTIDs are in use for replication, replicated transactions that are filtered out on the slave are persisted. If binary logging is enabled on the slave, the filtered-out transaction is written to the binary log as a `Gtid_log_event` followed by an empty transaction containing only `BEGIN` and `COMMIT` statements. If binary logging is disabled, the GTID of the filtered-out transaction is written to the `mysql.gtid_executed` table. This process ensures that there are no gaps in the set of executed GTIDs, and that the filtered-out transactions are not retrieved again if the slave reconnects to the master. Previously, this process was not done for `CREATE DATABASE`, `ALTER DATABASE`, and `DROP DATABASE` statements, but it is now carried out for those statements as well as for others. (Bug #27308751, Bug #88891)

- **Replication:** On a multithreaded slave, when a `STOP SLAVE` statement is executed on the slave, followed by a `START SLAVE` statement, the error log can report a different position in the binary log for the slave SQL thread when exiting, compared to the position reported for the slave SQL thread at the subsequent initialization.

  For a multithreaded slave, the position reported for the SQL thread on exit is a low water mark, up to which the replication stream is consistent and has no gaps. Transactions appearing before the position are guaranteed to have committed, but transactions after the position may have committed or not. However, this low water mark was being reported before the process to stop the worker threads was actually carried out, and the low water mark was subsequently updated by a checkpoint routine during that process. The timing of the log message has now been changed so that the final low water mark is reported as the position for the SQL thread on exit. (Bug #27300658)

- **Replication:** An online upgrade of a replication group that was running a version of MySQL earlier than 5.7.23 was not possible. When a member running a version earlier than either MySQL 5.7.22 or MySQL 8.0.11 tried to join a group with members running MySQL 5.7.21 or earlier, it failed to join the group because MySQL 5.7.21 does not send the value of it `lower_case_table_names` system variable. This mean that the later version compared the `lower_case_table_names` variable with an invalid value and exited the group. The work around is to take the group offline, upgrade to version 5.7.23 or 8.0.12 and later. Subsequent upgrades of the group can then be carried out without taking the group offline. See Upgrading Group Replication. (Bug #90794, Bug #27991334)

- **Replication:** It was possible to change the value of `gtid_purged` while running the Group Replication plugin, even though this variable should never be modified in such cases. (Bug #90146, Bug #27724561)

- **Replication:** The `PIT_TRANSACTIONS_NEGATIVE_CERTIFIED`, the `PIT_TRANSACTIONS_ROWS_VALIDATING` and the `PIT_TRANSACTIONS_LOCAL_ROLLBACK` member messages were not being correctly decoded. (Bug #90077, Bug #27692831)

- **Replication:** In certain situations, such as during distributed recovery procedure, the certification info garbage collection was purging more data than it should, resulting in conflicts not being detected. The garbage collection procedure has been improved to take this case in consideration. (Bug #89938, Bug #27652526)

- **Replication:** The `ER_GRP_RPL_SQL_SERVICE_FAILED_TO_RUN_SQL_QUERY` error was being logged incorrectly. (Bug #89788, Bug #27590534)

- **Replication:** Using `group_replication_force_members` to unblock a group, for example after losing majority, sometimes failed with error 1231. (Bug #86957, Bug #26394418)

- **Group Replication:** When the `group_replication_applier` channel's applier thread encountered an error, the `master_log_name` and `end_log_pos` values shown in the resulting error message were incorrect, since the events in a transaction are replicated before they are written to the binary log of the member where the transaction originated, with the result that the final values are unknown at the time they are applied on the replica by the applier thread. To avoid confusion, any such error messages encountered by a `group_replication_applier` channel no longer contain the binary log name or position. (Bug #89146, Bug #27368735)

- **macOS:** On macOS, the server autostarted at system boot time even when set not to in the MySQL preference pane. (Bug #27969174, Bug #90672)

- **Microsoft Windows:** For debug builds on Windows, `CREATE SPATIAL REFERENCE SYSTEM` with an empty definition string could raise assertions. (Bug #27672683)

- In the client library, signals were incorrectly unblocked for `ppoll()`. Thanks to Facebook for the patch. (Bug #28075623, Bug #90999, Bug #28096808, Bug #91067)

- Certain `ALTER TABLE` statements could disable table encryption. (Bug #28045585)

  References: This issue is a regression of: Bug #27389878.

- Under some conditions, `MyISAM` code compilation failed due to a missing include file. (Bug #28039150, Bug #90898)

- The `REGEXP_REPLACE()` function truncated its result when used in an `UPDATE` statement. (Bug #28027093, Bug #90870)

- `REGEXP_REPLACE()` results from one result set row could carry forward to the next row, resulting in accumulation of previous results in the current row. (Bug #27992118, Bug #90803)

- A heap overflow vulnerability in the MySQL client library was fixed. (Bug #27980823)

- Users who had the `SUPER` privilege were not permitted to modify the `keyring_operations` system variable. (Bug #27976270)

- Full-text search auxiliary tables created in MySQL 5.7 and MySQL 8.0.11 with timestamps of 0 caused a data dictionary upgrade failure when upgrading from MySQL 8.0.11 to MySQL 8.0.12. (Bug #27960500)

- A “source and destination overlap in memcpy” Valgrind error was returned when upgrading from MySQL 5.7. (Bug #27945658, Bug #90691)

- Linux binaries were unable to dump core after executing a `setuid()` call. (Bug #27929894, Bug #90642, Bug #23337428, Bug #21723)

- `RESET PERSIST` did not flush variable values from memory. (Bug #27924206)

- For debug builds, using `CREATE TABLE ... SELECT` to create a table with a long numeric string and indexing the column could lead to an assertion being raised. (Bug #27909771)

- For debug builds, `RESET PERSIST` for an unknown variable could cause an assertion to be raised. (Bug #27903874, Bug #90546)

  References: This issue is a regression of: Bug #27374791.

- Updates to column metadata of a view referencing a partitioned table could cause a server exit. (Bug #27903842)

- For debug builds, an assertion could be raised for `UNION` queries when computing the combined data type of a `GEOMETRY` column and `SELECT * FROM (SELECT NULL)`. (Bug #27903792, Bug #90550)

- For debug builds, starting the server with `thread_stack=0` and executing `ALTER TABLE` could raise an assertion. (Bug #27899274, Bug #90525)

- `SHOW TABLE STATUS` could cause a server exit. (Bug #27897815)

- For generated columns that used the `INTERVAL()` function, incorrect behavior could occur. (Bug #27881102)

- For Fedora 28, MySQL Community RPM packages would not install if Connector ODBC native packages were installed. (Bug #27850721)

- It was possible to drop the Performance Schema. (Bug #27830283)

- A runtime error could occur for calls to `mysql_sys_var_longlong()` or `String::copy()`. (Bug #27822413, Bug #27808412)

- An assertion was raised by error logging code that used an inappropriate error number. (Bug #27792903)

  References: This issue is a regression of: Bug #27462408.

- The `CONVERT TO CHARACTER SET charset_name` clause for `ALTER TABLE` produced a syntax error if *`charset_name`* was `DEFAULT`. (Bug #27760787, Bug #90222)

  References: This issue is a regression of: Bug #27389878.

- An unencrypted connection could result from a client connection attempt specifying that an encrypted connection was required, if the server was not configured to support SSL. (Bug #27759871)

- `INSERT INTO ... SELECT` did not perform as many checks as `INSERT INTO ... VALUES()` with respect to requiring valid geometry values for geometry columns. (Bug #27756083)

- `REGEXP_INSTR()` did not return an error message when used with invalid match mode flags. (Bug #27751277)

- Some data dictionary table definitions were modified to align more closely with their initial design. Additionally, the data dictionary version number was incremented, and code that refers to data dictionary table indexes now uses symbolic index names instead of numeric values. (Bug #27745526, Bug #90196)

- Executing a prepared statement to insert rows with large number of parameters was very slow. (Bug #27699248)

- `ibd2sdi` could exit due to performing a divide-by-zero operation. (Bug #27692051, Bug #90071)

- The `mysql-boost-8.0.4.tar.gz` source distribution omitted `NDB` sources. (Bug #27690232)

- `SHOW GRANTS` could cause a server exit if executed for an unprivileged user who inherited privileges from an anonymous user. (Bug #27678129)

- The parser incorrectly permitted the index name for `CREATE INDEX` to be empty. (Bug #27676427)

- Failures of server restarts initiated by systemd or `mysqld_safe` were not written to the error log. (Bug #27675050, Bug #90012)

- For debug builds, if a transaction was in read-only mode, trying to create or drop a histogram could raise an assertion. (Bug #27672693)

- If the server was started with `--skip-grant-tables`, `SET ROLE DEFAULT` caused a server exit. (Bug #27672154)

- Casting of `VARBINARY` values to `TIME` values was not performed correctly. (Bug #27665997)

- Error checking was not performed after creation of a condition for a `const` reference. (Bug #27665085)

- The dictionary information object managed by the MeCab model was used after the MeCab model was released. (Bug #27660368)

- Attempting to abort events of the `MYSQL_AUDIT_AUTHENTICATION_CLASS` class caused a server exit. These events are now not abortable. (Bug #27645636)

- These resource-group statements are now disallowed in stored functions because they cause an implicit commit, which is not permitted in a stored function: `CREATE RESOURCE GROUP`, `ALTER RESOURCE GROUP`, `DROP RESOURCE GROUP`. (Bug #27638623, Bug #89914)

- Audit log filter rules did not permit class names to be specified as an array of strings. That is now possible. Example:

```
{
  "filter": {
    "class": [
      { "name": [ "connection", "general", "table_access" ] }
    ]
  }
}
```

(Bug #27628325)

- On Windows, if the Visual C++ Redistributable for Visual Studio had been removed, MySQL uninstallation using the MSI installer failed. (Bug #27621546)
- `mysql_secure_installation` now loads the `validate_password` component rather than the deprecated `validate_password` plugin. (Bug #27619667)
- `REGEXP_LIKE()` using an expression containing a question mark (`?`) was not handled correctly when included in a prepared statement. (Bug #27595368)
- `REGEXP_SUBSTR()` did not always handle `NULL` patterns correctly. (Bug #27572258)
- Some `SHOW CREATE TABLE` statements could raise an assertion. (Bug #27569314)
- Spurious stages could be seen in the Performance Schema `events_stages_current` table. (Bug #27566220)
- Triggers for a table could execute in an order different from that specified by their `PRECEDES`/`FOLLOWS` clauses. (Bug #27544152)
- Attempts to set some persisted variables at startup (such as `keyring_operations` and `mandatory_roles`) produced a privilege violation error and the server did not start or did not set the variable. (Bug #27523095)
- Under some conditions, an unused-variable warning occurred in the MeCab full-text search code. Thanks to Laurynas Biveinis for the patch. (Bug #27519952, Bug #89598)
- Within stored procedures, it was not possible to write a condition handler to catch foreign key failures if the table involved used the `FEDERATED` storage engine. (Bug #27509959)
- It was possible to drop a primary key that served as the parent key in a foreign key relationship. In addition, for table creation, the SQL layer now checks whether there is a parent key for the foreign key and reports an error if it is missing. (Bug #27506922, Bug #89570, Bug #11754696, Bug #46337)
- With the `NO_ENGINE_SUBSTITUTION` SQL mode disabled, an error (rather than substitution) occurred for `CREATE TABLE` and `ALTER TABLE` if the desired engine was disabled using the `disabled_storage_engines` system variable. (Bug #27502530)
- Key rotation using the `keyring_encrypted_file` keyring plugin could cause a server exit. (Bug #27497018)
- If MySQL Enterprise Firewall was configured with a large number of rules, server shutdown could take a long time. (Bug #27492122)
- `SET PERSIST_ONLY` *`var_name`* `= DEFAULT` persisted the current global value, not the default value. (Bug #27489026)
- If MySQL is configured with the `LINK_RANDOMIZE` `CMake` option enabled, the `--gc-sections` compiler option is now disabled; the combination causes segmentation faults. (Bug #27484106)

- During server installation, the `mysql.infoschema`, `mysql.session`, and `mysql.sys` reserved accounts were associated with the `mysql_native_password` authentication plugin even when the default plugin was `caching_sha2_password`. (Bug #27454299)
- Some window functions used the wrong precision for their aggregated data types. (Bug #27452179)
- For a hexadecimal value *H*, `SELECT` with `WHERE (-1) - H` raised an assertion. (Bug #27452082)

  References: See also: Bug #27041382. This issue is a regression of: Bug #21982792.
- A memory leak in the `pfs-t` unit test was fixed. Thanks to Yura Sorokin for the patch. (Bug #27440735, Bug #89384)
- Multiple invocations of `CHAR()` within a stored program could yield different results once an invalid code point was passed as the argument. (Bug #27410088)
- With `automatic_sp_privileges` enabled, the `EXECUTE` and `ALTER ROUTINE` privileges were not correctly granted to routine creators. (Bug #27407480)
- Some queries that used ordering could access an uninitialized column during optimization and cause a server exit. (Bug #27389294)
- For debug builds, an assertion was raised (rather than an error) for `ALTER TABLE` for an `ARCHIVE` table with a `GEOMETRY NOT NULL` column containing empty strings. (Bug #27330634, Bug #89088)
- Accounts that authenticated with the `auth_sock` authentication plugin could not connect using older clients. (Bug #27306178)
- If system tables could not be read or initialized, the server wrote an incomplete error message. Thanks to Daniël van Eeden for a patch related to the fix. (Bug #27302337, Bug #89001)
- `RENAME USER` failed when used to rename a role that was not granted to any account or role. (Bug #27284699)
- The `keyring_okv` plugin did not always fail over to the standby OKV server when the primary server was unavailable. (Bug #27244099)
- The Performance Schema could attempt to access invalid `SOURCE` column values. (Bug #27231036)
- Mishandling of internal privilege structures could cause a server exit. (Bug #27230925)
- For debug builds, use of a too-small type for calculating the determinant in GIS computations could cause an assertion to be raised. (Bug #27135504)
- For attempts to increase the length of a `VARCHAR` column of an `InnoDB` table using `ALTER TABLE` with the `INPLACE` algorithm, the attempt failed if the column was indexed.

  If an index size exceeded the `InnoDB` limit of 767 bytes for `COMPACT` or `REDUNDANT` row format, `CREATE TABLE` and `ALTER TABLE` did not report an error (in strict SQL mode) or a warning (in nonstrict mode). (Bug #26848813)
- In MySQL 5.7.19, the `mysql.gtid_executed` table was excluded from dumps. This table is no longer excluded but its data is not dumped. (Bug #26643180, Bug #87455)

  References: This issue is a regression of: Bug #82848, Bug #24590891.
- The `slave_rows_search_algorithms` system variable controls how rows are searched for matches when preparing batches of rows for row-based logging and replication. Specifying `INDEX_SCAN` as one of the search algorithms performs an index scan if an index is present. In the situation where a different

primary key is used on the master and the slave, and a unique key is present on the slave, a bug in the code meant that the index scan was not being performed as it should be, and a slower table scan was being performed instead. The issue has now been corrected so that an index scan is used. (Bug #26450129, Bug #23311892, Bug #81500, Bug #81501)

- Certain cases of subquery materialization could cause a server exit. These queries now produce an error suggesting that materialization be disabled. (Bug #26402045)
- Spatial operations on geographic `GeometryCollection` values could return `ER_GIS_INVALID_DATA` for valid geometries. (Bug #26174808)
- For `MyISAM` tables, particular sequences of `INSERT` and `DELETE` statements could cause table corruption. (Bug #25541037)
- The `SHOW CREATE USER` statement requires the `SELECT` privilege for the `mysql` database, except to see information for the current user. In addition, for the current user, the statement requires the `SELECT` privilege for the `mysql.user` system table for display of the password hash in the `IDENTIFIED AS` clause; otherwise, the hash displays as `<secret>`. (Bug #24911117)
- `CREATE FUNCTION` produced an error if there was no default database and the function body called another function. (Bug #24357244, Bug #82350)
- For debug builds, a `SELECT` statement executed within a stored program could raise an assertion if derived table resolution failed. (Bug #23221336)
- If `mysqldump` or `mysqlpump` were used to dump binary data without the `--hex-blob` option, reloading the dump file could produce spurious warnings (values were inserted correctly regardless of the warnings). Such values are now written preceded by the `_binary` introducer to silence the warnings. (Bug #22601255, Bug #80150)
- String comparisons involving wildcards could reference uninitialized memory. (Bug #12635103)
- `REGEXP_REPLACE()` raised a `Valgrind` warning when invoked with an invalid capture group. (Bug #87842, Bug #27612255)

# Changes in MySQL 8.0.11 (2018-04-19)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

**Note**

Downgrade from MySQL 8.0 to MySQL 5.7 (or from a MySQL 8.0 release to a previous MySQL 8.0 release) is not supported. The only supported alternative is to restore a backup taken *before* upgrading.

- Character Set Support
- Compilation Notes
- Configuration Notes
- Deprecation and Removal Notes
- Error Handling
- INFORMATION_SCHEMA Notes

- Installation Notes
- Logging Notes
- Packaging Notes
- Performance Schema Notes
- Security Notes
- Spatial Data Support
- sys Schema Notes
- Test Suite Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed

## Character Set Support

- The `utf8mb3` character set will be replaced by `utf8mb4` in a future MySQL version. The `utf8` character set is currently an alias for `utf8mb3`, but will at that point become a reference to `utf8mb4`. To avoid ambiguity about the meaning of `utf8`, consider specifying `utf8mb4` explicitly for character set references instead of `utf8`. (WL #10778)

## Compilation Notes

- MySQL source code has been converted to Google style rules using `clang-format`. (Bug #27414321)
- Sun RPC is being removed from `glibc`. `CMake` now detects and uses `libtirpc` if `glibc` does not contain Sun RPC. (You might find it necessary to install `libtirpc` and `rpcgen` to take advantage of this `CMake` feature.) (Bug #27368272, Bug #89168)
- The minimum version of the Boost library for server builds is now 1.66.0. (Bug #27137349, Bug #88540)
- When building from source, the provided `CMake` script assumed that JUnit and Ant were installed and available at set paths. If this was not the case, building the GCS Java wrapper resulted in build errors. To address this a new `CMake` option `WITH_ANT=path_name` has been added to the build script, and which works in a similar way to the existing `WITH_BOOST` `CMake` option. Set `WITH_ANT` to the path of a directory where the Ant tarball, or an already unpacked archive, is saved. When `WITH_ANT` is not set, or is set with the special value `system`, the build assumes a binary `ant` exists in `$PATH`.

  Similarly, the `CMake` option `ENABLE_DOWNLOADS` already exists to download required libraries during a build, such as the Google Test distribution. Now, when `ENABLE_DOWNLOADS` is set, and JUnit or Ant are not found they are downloaded as part of the build process. (Bug #26678793)

## Configuration Notes

- A new `CMake` option, `BUNDLE_RUNTIME_LIBRARIES`, enables bundling of runtime libraries with server MSI and Zip packages for Windows. (Bug #27584796, Bug #89771)
- systemd service files now increase `LimitNOFILE` to 10000, to accommodate higher default settings in MySQL 8.0. (Bug #27530282)

- The new `WITH_SYSTEM_LIBS` `CMake` option, when enabled, selects the `system` value for any of the following options that are not set explicitly: `WITH_EDITLINE`, `WITH_ICU`, `WITH_LIBEVENT`, `WITH_LZ4`, `WITH_PROTOBUF`, `WITH_SSL`, `WITH_ZLIB`. (Bug #27356668, Bug #89136)

- The version of the compiled-in `zlib` library is now available at runtime as the value of the `version_compile_zlib` system variable. Thanks to Daniël van Eeden for the patch. (Bug #27235065, Bug #88825)

## Deprecation and Removal Notes

- **Incompatible Change:** These deprecated compatibility SQL modes have been removed: `DB2`, `MAXDB`, `MSSQL`, `MYSQL323`, `MYSQL40`, `ORACLE`, `POSTGRESQL`, `NO_FIELD_OPTIONS`, `NO_KEY_OPTIONS`, `NO_TABLE_OPTIONS`. They can no longer be assigned to the `sql_mode` system variable or used as permitted values for the `mysqldump --compatible` option.

  Removal of `MAXDB` means that the `TIMESTAMP` data type for `CREATE TABLE` or `ALTER TABLE` is treated as `TIMESTAMP`, and is no longer treated as `DATETIME`.

  For MySQL 5.7 applications that use SQL modes removed in MySQL 8.0, statements may fail when replicated from a MySQL 5.7 master to a MySQL 8.0 slave, or may have different effects on master and slave. To avoid such problems, applications that use modes removed in MySQL 8.0 should be revised to avoid them. (WL #8383, WL #9465)

- The following features related to account management have been removed:

  - Using `GRANT` to create users. Instead, use `CREATE USER`. Following this practice makes the `NO_AUTO_CREATE_USER` SQL mode immaterial for `GRANT` statements, so it too is removed.

  - Using `GRANT` to modify account properties other than privilege assignments. This includes authentication, SSL, and resource-limit properties. Instead, establish such properties at account-creation time with `CREATE USER` or modify them afterward with `ALTER USER`.

  - `IDENTIFIED BY PASSWORD '`*`hash_string`*`'` syntax for `CREATE USER` and `GRANT`. Instead, use `IDENTIFIED WITH` *`auth_plugin`* `AS '`*`hash_string`*`'` for `CREATE USER` and `ALTER USER`, where the `'`*`hash_string`*`'` value is in a format compatible with the named plugin.

    Additionally, because `IDENTIFIED BY PASSWORD` syntax has been removed, the `log_builtin_as_identified_by_password` system variable is superfluous and has been removed.

  - The `PASSWORD()` function. Additionally, `PASSWORD()` removal means that `SET PASSWORD ... = PASSWORD('`*`auth_string`*`')` syntax is no longer available.

  - The `old_passwords` system variable.

  For MySQL 5.7 applications that use features removed in MySQL 8.0, statements may fail when replicated from a MySQL 5.7 master to a MySQL 8.0 slave, or may have different effects on master and slave. To avoid such problems, applications that use features removed in MySQL 8.0 should be revised to avoid them and use alternatives when possible. (Bug #27455708, WL #10774)

## Error Handling

- `perror` is more flexible in understanding its arguments. For example, for the `ER_WRONG_VALUE_FOR_VAR` error, `perror` understands any of these arguments: `1231`, `001231`, `MY-1231`, or `MY-001231`, or `ER_WRONG_VALUE_FOR_VAR`. If an error number is in the range where MySQL and operating system errors overlap, `perror` displays both error messages. (Bug #27462408)

## INFORMATION_SCHEMA Notes

- The new `INFORMATION_SCHEMA.KEYWORDS` table lists the words considered keywords by MySQL and, for each one, indicates whether it is reserved. This table provides applications a runtime source of MySQL keyword information. See The INFORMATION_SCHEMA KEYWORDS Table.

  In addition, if you build MySQL from source, the build process generates a `keyword_list.h` header file containing a array of keywords and their reserved status. This file can be found in the `sql` directory under the build directory. This file may be useful for applications that require a static source for the keyword list. (WL #8989)

## Installation Notes

- RPM and Debian packages now add the current MySQL series to the `share` directory name (for example `share/mysql-8.0`). (Bug #27638192, Bug #26990531)

## Logging Notes

- Messages written to the error log now indicate the subsystem in which the event occurred. Possible subsystem values are `InnoDB` (the `InnoDB` storage engine), `Repl` (the replication subsystem), `Server` (otherwise). (WL #11150)

## Packaging Notes

- Installing or upgrading using Debian packages now presents a dialog with an option for setting the default authentication plugin to `mysql_native_password`. This may be useful for backward compatibility at some installation sites. (Bug #27454287, Bug #27667294)

- The `compile_innodb` script is no longer used and has been removed from MySQL distributions. Thanks to Alexey Kopytov for the suggestion. (Bug #27448099, Bug #89437)

## Performance Schema Notes

- A new Performance Schema table `log_status` provides information that enables an online backup tool to copy the required log files without locking those resources for the duration of the copy process. When the `log_status` table is queried, the server blocks logging and related administrative changes for just long enough to populate the table, then releases the resources. The `log_status` table informs the online backup which point it should copy up to in the master's binary log and `gtid_executed` record, and the relay log for each replication channel. It also provides relevant information for individual storage engines, such as the last log sequence number (LSN) and the LSN of the last checkpoint taken for the InnoDB storage engine.

  The `BACKUP_ADMIN` privilege is required for access to the `log_status` table. (Bug #27599831, WL #9452)

- Performance Schema objects now have a default character set of `utf8mb4`. (Bug #27407745)

## Security Notes

- MySQL 8.0 builds now use OpenSSL rather than yaSSL as the default SSL library. MySQL no longer supports building using yaSSL, and source distributions no longer include yaSSL.

  The `WITH_SSL CMake` option no longer permits `bundled` (use yaSSL) as a valid value, and the default option value has changed from `bundled` to `system` (use the version of OpenSSL installed on the host system).

In place of yaSSL, wolfSSL may be used as a functionally equivalent alternative that has a GPLv2-compatible license. In addition, wolfSSL (like OpenSSL) supports the TLSv1.2 protocol, which yaSSL does not.

For instructions about building MySQL from source using wolfSSL, see the `extra/README-wolfssl.txt` file in a MySQL source distribution. The `WITH_SSL` `CMake` option now permits `wolfssl` as a valid value. To use `-DWITH_SSL=wolfssl`, you must follow the instructions in `README-wolfssl.txt`. (Bug #26671671, WL #8998)

- MySQL now supports FIPS mode, if compiled using OpenSSL 1.0.2, and an OpenSSL library and FIPS Object Module are available at runtime. FIPS mode imposes conditions on cryptographic operations such as restrictions on acceptable encryption algorithms or requirements for longer key lengths. The `ssl_fips_mode` system variable enables control of FIPS mode on the server side. The `--ssl-fips-mode` client option enables control of FIPS mode on the client side for these programs: `mysql`, `mysqladmin`, `mysqlbinlog`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlpump`, `mysqlshow`, `mysqlslap`, `mysqltest`, `mysqlxtest`, `mysql_secure_installation`, and `mysql_upgrade`. See FIPS Support. (WL #8102)

## Spatial Data Support

- **Incompatible Change:** These geometry collection changes have been implemented:
  - The standard `GeomCollection` name for the geometry collection data type is supported as a synonym for the `GeometryCollection` data type name.
  - The `GeomCollection()` function is supported as a synonym for the `GeometryCollection()` function.
  - `GeomCollection` and `GeomCollection()` are the preferred data type and function names.

  Incompatibility: As a consequence of the change in preferred data type name, some `INFORMATION_SCHEMA` tables, statements such as `SHOW CREATE TABLE`, and the `ST_GeometryType()` function that display data type information now display `GEOMCOLLECTION` rather than `GEOMETRYCOLLECTION`. Applications that look for `GEOMETRYCOLLECTION` should be adjusted to account for this change.

  For more information, see GeometryCollection Class, and MySQL-Specific Functions That Create Geometry Values. (WL #2388)

## sys Schema Notes

- `sys` Schema objects now have a default character set of `utf8mb4`. (Bug #27675959)

## Test Suite Notes

- Reduction of compiler and platform differences in GIS handling of floating-point results enables simplification of related test cases that no longer need rounding to avoid spurious test failures. Thanks to Daniel Black for the patch. (Bug #26540102, Bug #87223, Bug #27462294)

## X Plugin Notes

- When using MySQL as a document store, the use of `SPATIAL` indexes on geometry-valued columns using the X DevAPI is now fully supported. Previously, a spatial index could be created, but the spatial reference system identifier (SRID) was not included in the indexed column, so the index was not used by the optimizer. The `SRID` attribute is now added to the indexed column. (Bug #27568243)

- Socket files used with X Plugin tests are now limited to 22-character file names. (Bug #27460677, Bug #89464)
- The fix for Bug #26328274 resulted in an assertion in debug builds when an empty host name was used for X Plugin connections. The issue is now fixed. (Bug #27266584, Bug #88925)
- X Plugin did not populate the password cache when a user account with no password set was used with the `PLAIN` authentication method. This caused later authentication attempts to fail. The password cache is now correctly populated with an entry if the user account's password is empty. (Bug #27257722)
- X Plugin now supports the `RESTART` SQL statement, which was added in MySQL 8.0.4 to enable a MySQL server instance to be restarted from within a client session. (Bug #27245319)
- MySQL Shell connections using X Protocol can use either TCP or Unix sockets as of version 8.0.4. This fix allows the use of encryption protocols with UNIX socket connections, so user accounts that require encrypted connections are able to connect. (Bug #27192091)
- The fix for Bug #26647488, which corrected the handling of the `DATETIME` data type, now also correctly decodes messages received from older versions of the plugin that do not have the new field available to identify the data type. (Bug #27169735)
- X Plugin connection attempts using the X Protocol did not return an error when the default database specified in the connection options was invalid, and the connection was allowed with a null default database. Connection attempts using the classic MySQL protocol did return an error and disallowed the connection. X Protocol connection attempts now also disallow the connection if an invalid schema is specified. (Bug #26965020)
- X Plugin is now loaded by default, meaning there is no longer any need to install the plugin. (WL #11040)

## Functionality Added or Changed

- **InnoDB:** The `ibd2sdi` serialized dictionary information (SDI) extraction utility for `InnoDB` tablespaces now supports a `--pretty` option that controls JSON pretty print formatting of SDI data. Pretty print formatting is enabled by default. Disabling pretty print formatting using `--skip-pretty` reduces the size of extracted SDI data. (Bug #27546514)
- **InnoDB:** To improve startup performance on systems with large buffer pools, buffer pool initialization is now multithreaded. (Bug #22963374, Bug #80784)
- **InnoDB:** The following redo logging optimizations were implemented:
  - User threads can now write concurrently to the log buffer without synchronizing writes, and add dirty pages to the flush list in a relaxed order.
  - Dedicated log writer threads are now responsible for writing redo log records from the log buffer to the system buffers and flushing the system buffers to the redo log files.
  - System variables were added for configuring the use of spin delay by user threads waiting for flushed redo:
    - `innodb_log_wait_for_flush_spin_hwm`: Defines the maximum average log flush time beyond which user threads no longer spin while waiting for flushed redo.
    - `innodb_log_spin_cpu_abs_lwm`: Defines the minimum amount of CPU usage below which user threads no longer spin while waiting for flushed redo.

  - `innodb_log_spin_cpu_pct_hwm`: Defines the maximum amount of CPU usage above which user threads no longer spin while waiting for flushed redo.

- The `innodb_log_buffer_size` configuration option is now dynamic, which permits resizing of the log buffer while the server is running.

For more information, see Optimizing InnoDB Redo Logging. (WL #10310)

- **macOS:** The macOS installer now asks the user for the `root` password to use rather than generating a random one. (Bug #22608157, Bug #80144)

- **macOS:** The MySQL preference pane included with the macOS installer now includes a setting that specifies which `my.cnf` file the server should read when started by `launchd`. (Bug #19775549, Bug #74214)

- **Solaris:** On Solaris, MySQL can now be built with Developer Studio 12.6. (Bug #27055190, Bug #88316, Bug #28165246, Bug #91214)

- Logging during data directory initialization (for example, using `mysqld --initialize`) has changed:

  - The startup message indicates that the server is running in initialization mode.

  - A message is written indicating when initialization has completed.

  - The message about the CA certificate being self-signed is suppressed.

  (Bug #27557952, Bug #89693)

- The `-log` suffix is no longer appended to the server version string if logging is enabled. This affects, for example, the value of the `version` system variable. (Bug #27532097, Bug #89626)

- Server shutdown lines written to the error log now include the server version. (Bug #27522788, Bug #87910)

- `SHOW CREATE TABLE` normally does not show the `ROW_FORMAT` table option if the row format is the default format. This can cause problems during table import and export operations for transportable tablespaces. MySQL now supports a `show_create_table_verbosity` system variable that, when enabled, causes `SHOW CREATE TABLE` to display `ROW_FORMAT` regardless of whether it is the default format. (Bug #27516741)

- Performance Schema memory instrumentation overhead was reduced. (Bug #27500610)

- During data directory initialization, the server created objects using `sql_mode=''`. Now the server uses the default `sql_mode` value (which, among other settings, includes strict mode). (Bug #27476029, Bug #89495)

- The collations defined in `mysql-test/std_data/Index.xml` are for test purposes only and should not be used otherwise. Comments were added to the file to make this clear because some users were using them in non-test environments. (Bug #27421838)

- `RESET PERSIST` no longer performs an implicit commit. (Bug #27374791)

- For a persisted variable, the `SET_TIME` column of the Performance Schema `variables_info` table now reflects the time at which the variable was persisted, not server startup time. (Bug #25677422)

- Previously, `SHOW CREATE TABLE` did not show the collation for a column if the collation was the same as the table default, even if the collation was explicitly specified at table-creation time. Now, `SHOW`

`CREATE TABLE` always shows the column collation if the collation was explicitly specified, even if the collation is the same as the table default. (Bug #11754608, Bug #46239)

## Bugs Fixed

- **Important Change:** When `NO_ZERO_DATE` SQL mode is enabled, `ALTER TABLE` rejects attempts to add `DATE` or `DATETIME` columns that are declared `NOT NULL` to a nonempty table if no `DEFAULT` is specified for that column. Since a generated column has no default value, this restriction also prevented addition of a generated `DATE NOT NULL` or `DATETIME NOT NULL` column to a nonempty table. Since the expression used to generate the column can be used in place of the missing default for determining the column value, this restriction is now lifted for generated columns. (Bug #27252354)
- **Important Change:** For `ANALYZE TABLE ... UPDATE HISTOGRAM` statements, the `WITH *N* BUCKETS` clause is now optional. If not specified, the default value for *`BUCKETS`* is 100. (Bug #89909, Bug #27637968)
- **InnoDB:** The server was stopped before a fatal error message was written to the error log. (Bug #27626478)
- **InnoDB:** When determining if a wait lock could be granted, locks held by victim transactions that were being rolled back were ignored. (Bug #27607235)
- **InnoDB:** An error occurred in `Sdi_Decompressor::decompress()` when compiling a debug build using GCC 7.2.0. (Bug #27565889)
- **InnoDB:** An assertion was raised when registering buffer pool chunks during parallel creation of multiple buffer pools. A mutex is now used to protect registration of buffer pool chunks in the buffer pool chunk map. (Bug #27565865)
- **InnoDB:** Enabling the standard monitor caused an error. (Bug #27534041)
- **InnoDB:** An internal field used to mark a column as hidden was changed from a boolean field to an enum field to support additional levels of column visibility. (Bug #27499518, Bug #89553)
- **InnoDB:** An invalid zlib-related Valgrind failure reported that a conditional jump or move depends on uninitialized value. (Bug #27482740, Bug #89509)
- **InnoDB:** An incorrect `GROUP BY` result was returned when using the `TempTable` storage engine and a `NO PAD` collation. (Bug #27454796)
- **InnoDB:** An incorrect compression length value in a page compression function caused hole punching to be skipped the first time pages are compressed. (Bug #27399897)
- **InnoDB:** Starting the server with an incorrect `--log-bin` option value raised an assertion, and subsequent attempts to start the server with a correct `--log-bin` value or without the `--log-bin` option continued to produce assertion failures. (Bug #27393317)
- **InnoDB:** A large object (LOB) reference was not updated when a LOB was inserted during an `UPDATE` operation. (Bug #27368662)
- **InnoDB:** After restarting the server, a `TRUNCATE TABLE` operation raised an assertion due to table flags that were inconsistent with remote data directory information stored in memory. (Bug #27367639)
- **InnoDB:** Attempting to create a temporary table in a file-per-table tablespace using `CREATE TEMPORARY TABLE ... TABLESPACE` syntax failed to report an error. Temporary tablespaces are only permitted in the temporary tablespace. (Bug #27361662)
- **InnoDB:** A latch taken on the first page of a large object (LOB) was not released before another thread attempted to allocate the page, causing a deadlock. (Bug #27360492)

- **InnoDB:** Converting an implicit lock to an explicit lock was optimized by the introduction of a member variable that tracks the smallest active transaction ID.

  Thanks to Zhai Weixiang for the patch. (Bug #27353468, Bug #89127)

- **InnoDB:** The `mysql.innodb_ddl_log` and `mysql.innodb_dynamic_metadata` tables, which are used internally by the `InnoDB` storage engine, are now protected. DDL and DML operations on these tables are no longer permitted, and the tables no longer appear in `INFORMATION_SCHEMA` query results. (Bug #27353455, Bug #89129)

- **InnoDB:** Upgrading a MySQL 5.7 instance with an `innodb_page_size` setting of 4K caused `mysql_upgrade` to fail with a “Specified key was too long; max key length is 768 bytes” error when creating or modifying system tables. The error was suppressed to permit the `mysql_upgrade` client to run on an instance with an `innodb_page_size` setting of 4K. (Bug #27338431)

- **InnoDB:** The recovery process could not properly determine space_id values, causing the server to exit. (Bug #27331936)

- **InnoDB:** An `ALTER TABLE` operation that modified the nullability of an indexed column raised an assertion. `InnoDB` incorrectly assumed that a new primary key must be added instead of created from an existing unique key. (Bug #27330454, Bug #89087)

  References: This issue is a regression of: Bug #24397406.

- **InnoDB:** After an `ALTER TABLE ... IMPORT TABLESPACE` operation, metadata could not be retrieved correctly from the `INFORMATION_SCHEMA.INNODB_COLUMNS` table. (Bug #27322099)

- **InnoDB:** An `INSERT` operation involving a large object (LOB) value raised an assertion. After splitting a LOB fragment, there was insufficient space for a directory entry. (Bug #27320977)

- **InnoDB:** On a server using the utf8mb4 character set, an `INSERT` or `UPDATE` during an `ALTER TABLE` operation that changed the row format of a table that included a virtual column caused an index corruption. (Bug #27319084)

- **InnoDB:** The data retrieved from `INFORMATION_SCHEMA.INNODB_COLUMNS` was incorrect for tables containing a virtual column. (Bug #27316037)

- **InnoDB:** The server failed to start after recovery due to an invalid keyring file. The keyring file created from backup was empty. (Bug #27307740)

- **InnoDB:** A deadlock between background threads, one attempting to evict a full-text search table from the cache, and the other attempting to synchronize a table, caused InnoDB Cluster nodes to fail. (Bug #27304661)

- **InnoDB:** A typo was corrected in an `InnoDB` recovery message. Thanks to Daniël van Eeden for the patch. (Bug #27287979, Bug #27010613, Bug #88185)

- **InnoDB:** A message stating that encryption algorithm support is missing was reported during `InnoDB` recovery. (Bug #27286020)

- **InnoDB:** A missing or mismatched keyring file caused an assertion failure at startup. (Bug #27278324)

- **InnoDB:** An error was returned indicating that encryption support was missing when checking the space ID of an encrypted tablespace file. (Bug #27278100)

- **InnoDB:** A debug message that prints a “too many open files” warning is now only printed if the number of open files reaches a limit of 300. (Bug #27277844)

- **InnoDB:** Failure to skip predicate locks when releasing gaps locks raised debug assertions, as did failure to remove the supremum record bit prior releasing gaps locks on the supremum. (Bug #27272806, Bug #27294066)
- **InnoDB:** A tablespace ID discovery failure occurred when starting the server on a cloned data directory. (Bug #27265874)
- **InnoDB:** An unprotected global variable (`fil_n_file_opened`) caused an assertion when shutting down the server. (Bug #27265473)
- **InnoDB:** An assertion that is no longer valid was raised when checking for pending I/O on the file in use. (Bug #27263206)
- **InnoDB:** An empty general tablespace caused a failure when upgrading to MySQL 8.0. The empty data file was expected to have a specific number of pages. (Bug #27261822)
- **InnoDB:** A startup failure occurred because a directory specified by the `--innodb-directories` option contained a keyring data file. The `--innodb-directories` option should only recognize `InnoDB` tablespace files. (Bug #27253116)
- **InnoDB:** A function that calculates the increase in minimum bounding rectangle (MBR) size incorrectly returned 0, resulting in R-tree corruption. (Bug #27237472)
- **InnoDB:** A `REPLACE` operation on a temporary table raised an assertion. (Bug #27225649, Bug #27229072)
- **InnoDB:** An online `ALTER TABLE` operation on a table accompanied by concurrent DML on the same table raised an assertion. An end-of-log check was not performed prior to accessing the DML log to determine the length of a virtual column. (Bug #27158030)
- **InnoDB:** When the addition of a virtual index failed, the virtual index that was freed was not removed from the lists of virtual column indexes. (Bug #27141613)
- **InnoDB:** Adding a virtual column and index in the same statement caused an error. (Bug #27122803)
- **InnoDB:** An undo tablespace truncate operation raised an assertion. The check that determines if undo tablespace segments are inactive was invalid. (Bug #27114068)
- **InnoDB:** When the TempTable storage engine was unable to allocate memory using temporary files and mmap, an invalid error was reported indicating that the defined on-disk storage engine would be used instead. (Bug #26990375)
- **InnoDB:** Latch order was incorrectly defined in debug instrumentation causing a latch order violation during an `INSERT` operation. (Bug #26983412)
- **InnoDB:** A tablespace import operation on a server with a default row format of `REDUNDANT` raised an assertion failure. (Bug #26960215)
- **InnoDB:** A stored field based on a generated column permitted the base column to have a NULL value. (Bug #26958695)
- **InnoDB:** Evaluation of a subquery in a resolving function raised an assertion. (Bug #26909960)
- **InnoDB:** Locks related to data dictionary operations caused degradation in `CREATE TABLE` performance. (Bug #26848711, Bug #87827)
- **InnoDB:** A `LOCK INSTANCE FOR BACKUP` operation was able to acquire a backup lock without waiting for a concurrent DDL operation to finish. (Bug #26831155, Bug #87812)

- **InnoDB:** An incorrectly specified `innodb_data_file_path` or `innodb_temp_data_file_path` value returned a syntax error that did not specify the name of the system variable that caused the initialization failure. (Bug #26805833)
- **InnoDB:** An handler that was opened for a `TRUNCATE TABLE` operation caused the operation to assert. The server now closes open handlers before dropping the table.

  The patch for this bug also improves `TRUNCATE TABLE` error messages related to foreign key constraints, and makes it possible to truncate a corrupted table. (Bug #26710839)
- **InnoDB:** An online DDL operation that rebuilds the table raised an assertion when the last insert log record to be applied was split across two pages. (Bug #26696448, Bug #87532)
- **InnoDB:** An assertion was raised when attempting to generate a virtual column value dependent on an off-page large object value that was not yet stored. (Bug #26625652)
- **InnoDB:** An R-tree search counted delete-marked records, causing it to find more records than present in the clustered index. As result, the valid R-tree index was marked as corrupted. (Bug #26381207)
- **InnoDB:** A `RENAME TABLE` operation that renamed the schema failed to rename full-text search common auxiliary tables that were left behind when the full-text search index was removed previously, resulting in a assertion failure when attempting to drop the old schema. (Bug #26334149)
- **InnoDB:** An assertion was raised when a thread attempted to read a record containing BLOB data while another thread was writing the same data to external pages. (Bug #26300119)

  References: This issue is a regression of: Bug #23481444.
- **InnoDB:** Column prefixes for externally stored columns exceeded the amount of available memory in the online log. (Bug #25928471)
- **InnoDB:** A problematic lock that was introduced to protect writing of dynamic metadata changes to the redo log during a checkpoint was removed. A mutex is now used instead. The checkpoint LSN was also adjusted to protect dynamic metadata changes. (Bug #25834374)
- **InnoDB:** `InnoDB` failed to account for a virtual column when using the column offset to search an index for an auto-increment column. (Bug #25076416)
- **InnoDB:** An invalid debug condition caused a buffer pool chunk allocation failure, which resulted in an assertion failure when a purge thread attempted to access an unallocated chunk. (Bug #23593654)

  References: This issue is a regression of: Bug #21348684.
- **InnoDB:** Threads were prevented from entering the `InnoDB` storage engine due to ticket acquisition deadlocks. (Bug #23476050)
- **Partitioning:** An error returned by the storage engine while executing `ALTER TABLE EXCHANGE PARTITION` was not handled correctly by the server. (Bug #27320682)
- **Replication:** Empty XA transactions that have been rolled back should not be recorded in the binary log or added to the set of transactions in the `gtid_executed` system variable. However, if rolled-back empty XA transactions were executed after an empty XA transaction ending with an `XA COMMIT ... ONE PHASE` statement, they were being incorrectly added to the binary log as an `XA ROLLBACK` statement, and also to the gtid_executed system variable. The issue is now fixed. (Bug #27435974)
- **Replication:** When the system variable `gtid_next` was set to AUTOMATIC, empty XA transactions that were rolled back and not recorded in the binary log, were being incorrectly added to the set of transactions in the `gtid_executed` system variable, causing an inconsistency. The rolled-back empty XA transactions are no longer recorded in `gtid_executed`. (Bug #27407670)

- **Replication:** The value of `LAST_CONFLICT_FREE_TRANSACTION` in the `replication_group_member_stats` Performance Schema table was not correctly synchronized across group members. (Bug #27317431)

- **Replication:** When a member is joining a group there is a chance of the request to join being rejected. If the rejection resulted in a retry, for example because the seed member being contacted was not in the group, then there was a possibility of the retry cycle continuing infinitely. (Bug #27294009)

- **Replication:** The default binary log expiration period of 30 days is now set as the default for the `binlog_expire_logs_seconds` system variable, instead of the deprecated `expire_logs_days` system variable. `binlog_expire_logs_seconds` now defaults to a value of 2592000 (30*24*60*60 seconds), and `expire_logs_days` now defaults to zero.

  In MySQL 8.0.3 and 8.0.4, the effects of `binlog_expire_logs_seconds` and `expire_logs_days` were cumulative. From MySQL 8.0.11, this is no longer the case, and the new behavior is as follows:

  - If you specify a non-zero value for both variables, only the value of `binlog_expire_logs_seconds` is used to determine the binary log expiration period, and the value of `expire_logs_days` is ignored.

  - If you specify a non-zero value for only one variable, that value is used to determine the binary log expiration period, and the default for the other is not applied.

  - If you specify a value for neither of the variables, the default of 30 days from `binlog_expire_logs_seconds` is used as the binary log expiration period.

  An error message is issued if you attempt to set one of the variables dynamically when the other is already set, and a warning message is issued if you specify a non-zero value for both variables at startup. A deprecation warning is also issued if you set a value for `expire_logs_days`.

  To disable automatic purging of the binary log, specify a value of 0 explicitly for `binlog_expire_logs_seconds`, and do not specify a value for `expire_logs_days`. For compatibility with earlier releases, automatic purging is also disabled if you specify a value of 0 explicitly for `expire_logs_days` and do not specify a value for `binlog_expire_logs_seconds`. In that case, the default for `binlog_expire_logs_seconds` is not applied. You no longer need to set both variables to zero to disable automatic purging. (Bug #27252658)

- **Replication:** The DDL statements `CREATE DATABASE`, `ALTER DATABASE`, and `DROP DATABASE` are no longer allowed to execute in parallel with other statements on a multithreaded slave. These statements require an exclusive metadata lock on the schema, causing a theoretical possibility of a deadlock when `slave_preserve_commit_order=1` is set on the replication slave. (Bug #27165446)

- **Replication:** On a replication slave with binary logging disabled, an empty DDL statement (for example, due to a replication filter) raised an assertion. With binary logging enabled, `BEGIN` and `COMMIT` statements would be written to the binary log to produce a transaction, but these statements were not available with binary logging disabled, causing the assertion. MySQL Server now checks for the situation where the DDL transaction is empty and is not marked as started, and handles it without raising the assertion. (Bug #27164661)

- **Replication:** A new system variable `rpl_read_size` is available to set the minimum amount of data in bytes that is read from the binary log files and relay log files. The default read size is 8KB. If heavy disk I/O activity for these files is impeding performance for the database, increasing the read size might reduce file reads and I/O stalls if the files are not being cached by the operating system.

  Note that a buffer the size of this value is allocated for each thread that reads from the binary log and relay log files, including dump threads on masters and coordinator threads on slaves. Setting a large value might therefore have an impact on memory consumption for servers.

Thanks to the Facebook team for the patch. (Bug #27147095)

- **Replication:** On a server with GTIDs enabled, if `RESET MASTER` was used when a transaction had been flushed but not yet committed, the server GTIDs were left in an incorrect state. At this point, the GTID for the transaction had been added to the set of owned GTIDs, but not yet to the set of executed GTIDs. The GTID was removed from the binary log when the content of the uncommitted transaction was cleared by the `RESET MASTER` statement, but the GTID could not be reused because it was already owned, so the sequence of GTIDs had a gap.

  With this fix, `RESET MASTER` acquires a global read lock to ensure that no transactions are in commit stage during its operation, and no new transactions enter into commit stage during its operation. If there are any ongoing commits, `RESET MASTER` waits until they are complete. The global read lock is released when the reset is finished, and then storage engines are notified of the successful reset. If `FLUSH TABLES WITH READ LOCK` has already been executed for the thread, `RESET MASTER` does not retry the lock. (Bug #27041759)

- **Replication:** Attempts to set a GTID for an unsafe or empty XA transaction using a `SET @@SESSION.GTID_NEXT` statement were not being handled correctly. On a server with `gtid_mode` set to `OFF` or `OFF_PERMISSIVE`, and `enforce_gtid_consistency` set to `OFF`, setting a GTID for an unsafe XA transaction (creating or dropping a temporary table) caused an inconsistency in the server's GTID violation counter and raised an assertion. Testing also showed that setting a GTID for an empty XA transaction caused an incorrect sequence of events to be written to the binary log. Both these situations are now handled correctly. (Bug #27041402, Bug #88272)

- **Replication:** The error message issued for a server stop while rotating the binary log did not state the correct cause for the issue. The error message has been enhanced to include the actual error causing the server stop as part of the message. A new error `ER_OOM_SAVE_GTIDS` has also been added for the situation where an out-of-memory error occurred while saving the set of GTIDs from the last binary log into the `mysql.gtid_executed` table. (Bug #27040070, Bug #88262)

- **Replication:** The statements `CREATE USER IF EXISTS` (or `IF NOT EXISTS`) and `ALTER USER IF EXISTS` (or `IF NOT EXISTS`) were written to the binary log even when the query resulted in an error. MySQL Server now checks for errors that cause these queries to fail (for example, an invalid plugin was specified), and does not log the statement in that situation. Note that if these statements succeed but have no effect on the master because the condition is not met, the statements are written to the binary log, as the condition might be met on a replication slave (see Bug #25813089, Bug #85733). (Bug #26680035)

  References: See also: Bug #25813089, Bug #85733.

- **Replication:** Following the introduction of support for atomic DDL in MySQL 8.0, a replication slave that supports atomic DDL can diverge from a replication master at an earlier MySQL version that does not support atomic DDL. If the DDL statement can only be partly applied, the slave using MySQL 8.0 rolls back the whole DDL statement, but the master using an earlier version might commit the valid parts of the DDL statement. An error message is now logged in the event of an unsuccessful commit for an atomic DDL statement on the master and the slave, and you should check that the states of the master and the slave have not diverged before proceeding. (Bug #26133488)

- **Replication:** When GTIDs are enabled on a replication master and slave, and the slave connects to the master with the `MASTER_AUTO_POSITION=1` option set, the master must send the slave all the transactions that the slave has not already received, committed, or both. If any of the transactions that should be sent by the master have been already purged from the master's binary log, the master sends

the error ER_MASTER_HAS_PURGED_REQUIRED_GTIDS (1789) to the slave, and replication does not start.

The message provided for the error ER_MASTER_HAS_PURGED_REQUIRED_GTIDS has been changed to provide advice on the correct action in this situation, which is for the slave to replicate the missing transactions from another source, or for the slave to be replaced by a new slave created from a more recent backup. The message advises that the master's binary log expiration period can be revised to avoid the situation in future. In addition, the master now identifies the GTIDs of the purged transactions and supplies them in its error log in the warning message ER_FOUND_MISSING_GTIDS (11809), so that you do not need to calculate the missing GTIDs manually. (Bug #26004541, Bug #29769293)

- **Replication:** For updates to virtual generated columns containing the BLOB data type, both the old and the new BLOB values are required by some storage engines for replication. This fix extends the same behavior to JSON and GEOMETRY data types, which are based on the BLOB data type and so produce the same issue when the old value is not stored. (Bug #25873029)

- **Replication:** A behavior change in MySQL 8.0.0 moved the identification of generated columns from before to after the calculation of the write set for binary logging. This caused an issue with NDB cluster replication, because that storage engine omits the generated columns from the log event if they are unnecessary, leading to NULL values being set for them by the applier thread on the replication slave. The identification of generated columns has now been moved back to before the write set calculation, so that the issue does not occur. (Bug #25827408)

- **Replication:** On a multithreaded replication slave (with slave_parallel_workers greater than 0), the slave's lag behind the master was not being reported by the Seconds_Behind_Master field for SHOW SLAVE STATUS. The value is now reported correctly. Thanks to Robert Golebiowski for the patch. (Bug #25407335, Bug #84415)

- **Replication:** A plus sign (+) can be used with a GTID set in the statement SET @@GLOBAL.GTID_PURGED to indicate that the specified GTID set does not intersect with gtid_executed. The plus sign was also being permitted for GTID sets specified for the functions GTID_SUBSET() and GTID_SUBTRACT(), where it was not meaningful. The plus sign can no longer be specified with these functions. Also, error messages relating to the SET @@GLOBAL.GTID_PURGED operation have been improved. (Bug #24675979)

- **Replication:** When invoked with the options --read-from-remote-server and --hexdump, mysqlbinlog was not able to produce a hex dump of the binary log contents following an SQL statement that generated an autoincrement value, referenced a user-defined variable, or invoked RAND(). The event types for these events are followed by an informational row query log event, and mysqlbinlog caches the original event for printing when the subsequent row query log event is received. The pointer to the memory containing the original event was invalidated when the subsequent event was received, so the original data could not be accessed to produce the hex dump. The issue has now been fixed. (Bug #24674276)

- **Replication:** A number of changes were made to the binary log decoding procedure to improve handling of invalid or corrupted binary log entries. (Bug #24365972)

- **Replication:** Following the introduction of binary logging for XA transactions WL#6860, an assertion could be raised in debug builds during replication from a master with the feature to a slave without the feature, if MASTER_AUTO_POSITION=1 was set on the slave. The assertion has been removed, so that debug builds now have the same behavior as non-debug builds, and can attempt replication of unsupported event types whether or not MASTER_AUTO_POSITION=1 is set. (Bug #20677683)

- **Replication:** When using group_replication_ip_whitelist, it was possible to configure a group so that it functioned even though all members could not establish the internal group communication connection to each other, resulting in inconsistent behavior. Now, incoming connections are accepted

if the IP is in the white list or if the IP belongs to a current member of the XCom configuration. This ensures members are always able to create the internal network required for group communication. (Bug #87834, Bug #26846549, Bug #27406775)

- **Group Replication:** When MySQL was started with `--plugin-load='group_replication.so'` but Group Replication was not started, starting an asynchronous slave channel resulted in an unresponsive server. (Bug #27584141)
- **Group Replication:** Group Replication failed to start if certain required variables were persisted, or if persisted variables were not set in proper timestamp order. (Bug #27545544, Bug #27522405)
- **Group Replication:** Group Replication files have been moved from the `rapid` directory to the `plugin` directory. (Bug #27413789)
- **Group Replication:** Conflict detection uses schema and table names as part of the Primary Key Equivalent (PKE) in order to detect and disallow conflicting transactions. The value of `lower_case_table_names` changes how schema and table names are stored and externalized; depending on the value, this could persist a table named `T1` as `t1`. Such a difference in a group could cause inconsistencies. Now, members must all have the same value for `lower_case_table_names`. (Bug #27401817)
- **Group Replication:** Using an IP address or hostname in any Group Replication related configuration on macOS was failing. (Bug #27376511, Bug #89123, Bug #27604471)
- **Group Replication:** Changing the Group Replication required settings incorrectly on online secondary members could result in an unexpected halt. (Bug #27317478, Bug #27157202)
- **Group Replication:** After issuing `START GROUP_REPLICATION` the `gtid_mode` system variable is locked to prevent any modification to its value until the group is online. Any attempt to modify `gtid_mode` during this time is blocked. As part of the process of starting Group Replication the server needs to set `super_read_only=off`, which has dependencies on locks acquired by `SET GTID_MODE`. This could result in Group Replication hanging and there was no possibility to connect to the server to resolve the situation. To prevent this situation, when it is not possible to acquire the locks needed by `SET GTID_MODE` the operation aborts. (Bug #27312241)
- **Group Replication:** In the unlikely event of the group failing to start, for example because no seed member could be contacted, there is a one minute period where the group retries the start process. Attempting to modify a Group Replication plugin option during this period could result in an unexpected halt. Now, attempting to modify a Group Replication plugin option during plugin start or stop results in error. (Bug #27275312)
- **Group Replication:** Attempting to uninstall the Group Replication plugin on a server configured to start the plugin automatically at boot could result in a deadlock. Now, the uninstall process checks to ensure that the plugin is not in the process of starting and fails gracefully if it is. (Bug #27179336)
- **Group Replication:** Group Replication uses transaction write set extraction for conflict detection on group members. A performance regression was found in this process during detailed performance analysis, which is handled by optimizing memory allocation has been for write set extraction and memory copy operations. In addition, foreign key write sets are now collected only when the current table actually has foreign keys. (Bug #27016053)
- **Group Replication:** In a multi-primary group, when a member was also configured with a asynchronous replication channel, there was a possibility that the asynchronous channel could start before Group Replication started. This could result in the asynchronous channel processing transactions before the member became an online member of the group, causing issues when members tried to join the group. The fix ensures that asynchronous channels on group members do not start until the member has become online. (Bug #26648393)

- **Linux:** Builds on Alpine Linux now take advantage of AIO support. (Bug #27327874)
- **macOS:** On macOS, `mysql_config --libs` did not list any SSL libraries. (Bug #27232576)
- **Microsoft Windows:** On Windows, the `RESTART` statement is implemented by having `mysqld` fork, with one process acting as a monitor to the other, which acts as the server. This makes determining the server process to attach to for debugging more difficult. To alleviate this, starting the server with `--gdb` now suppresses forking. A side effect is that for a server started with this option, `RESTART` simply exits and does not restart. (Bug #27273229)
- **JSON:** The `JSON_QUOTE()` function could in some situations write the quoted string to the same buffer that holds the input string, which could lead to wrong results. Now a check is made to be sure that it does not attempt write into the same buffer that it is reading from. (Bug #27312444)
- **JSON:** `JSON_SEARCH()` no longer performs any modification of cached `Json_path` objects, now making any needed updates to a `String` object that represents the path instead. This saves on the number of round trips required between the path's `Json_path` and `String` representations, which speeds up execution. In addition, the *`one_or_all`* and *`escape_char`* arguments to `JSON_SEARCH()` were unnecessarily cached; these arguments are no longer cached, which should also improve this function's execution time. (Bug #87383, Bug #26614455)

  References: See also: Bug #21450084, Bug #77785, Bug #21472872.
- A check was implemented based on data dictionary version information to prevent starting the MySQL 8.0 server with a data directory created by a later MySQL release. The check was necessary to prevent an in-place downgrade, which is currently not supported. (Bug #27708249)
- Protected data dictionary tables are no longer accessible by statements placed in a server initialization file. (Bug #27674311, Bug #90010)
- Data dictionary and `INFORMATION_SCHEMA` version numbers were synchronized with the new MySQL server version number. (Bug #27674285)
- The server did not handle correctly certain `LIKE` queries using a `BLOB` column with a prefix index. (Bug #27660560)
- For builds configured using `-DWITH_PROTOBUF=SYSTEM`, building failed on systems with Protobuf 3.5 installed. (Bug #27638713, Bug #89915)
- The microseconds part of timestamps stored with persisted variables in `mysqld-auto.cnf` was always 000000. (Bug #27629719)
- Debian packages now handle dependencies for MeCab plugin dictionary files. Fedora packages now handle dependencies for both MeCab packages, not just the utf-8 package. (Bug #27612730, Bug #27613539)
- The change in MySQL 8.0.11 for platforms that use systemd to run `mysqld` as a normal process (`Type=notify`) and communicate using a socket file did not work on SLES before SLES 12.2. (Bug #27604999)
- The `NON_UNIQUE` column in the `INFORMATION_SCHEMA.STATISTICS` table had type `BIGINT` prior to MySQL 8.0, but became `VARCHAR` in MySQL 8.0 with the introduction of the data dictionary. The `NON_UNIQUE` column now has an integer type again (`INT` because the column need not be as large as `BIGINT`). (Bug #27593348, Bug #89793)
- With some inputs, `ST_Crosses()` could cause a server exit. (Bug #27576700)
- Some error messages were being written to the error log with an error ID of 0 rather than a legitimate error ID. (Bug #27575608)

- `mysql_upgrade` did not set `MYSQL_SERVER_PUBLIC_KEY` correctly, leading to failure to read the key file. (Bug #27568278)
- If `autocommit` was disabled, `mysql_upgrade` produced an error when upgrading from MySQL 5.7 to 8.0. (Bug #27549249)
- For debug builds, displaying very long stage names from the `PROCESSLIST_STATE` column of the Performance Schema `threads` table could raise an assertion. (Bug #27545688)
- `gtid_purged` handling had a memory leak. (Bug #27537968, Bug #89645)
- Using `SET PERSIST` to persist `optimizer_trace_offset` resulted in an incorrect variable value after server restart. (Bug #27536421)
- The `validate_password` component could leak memory. (Bug #27521770, Bug #89597)
- A typo in `sql/handler.h` was corrected. Thanks to Su Tristan for the patch. (Bug #27516280, Bug #89594)
- The Performance Schema `variables_info` table displayed incorrect `VARIABLE_SOURCE` and `VARIABLE_PATH` values for variables set within option files specified by `!include` or `!includedir` directives. (Bug #27514223)

  References: This issue is a regression of: Bug #25563891.
- After upgrading from MySQL 5.7 to MySQL 8.0, the first attempt to start the server on a MySQL 5.7 data directory with `--default-time-zone` set to a specific time value such as `"+00:00"` caused the server to exit. (Bug #27512609, Bug #89584)
- Lookups of character set internal numbers was not thread safe and could cause memory leaks. (Bug #27507950)
- Using the `-DWITH_LIBEVENT=system` and `-DWITH_ICU=system` `CMake` options together caused configuration to fail. (Bug #27505535, Bug #89398)
- Some messages were being written to the server error log using client error IDs. (Bug #27503787, Bug #89562)
- When run in key migration mode, the server could report an error for successful operations. (Bug #27493997)
- Upgrades from MariaDB to MySQL Community Edition failed on Fedora 27. (Bug #27484835)
- `ALTER TABLE` could hang in a `Waiting for tablespace metadata lock` state. (Bug #27472087, Bug #89487)
- Selecting from the Performance Schema `status_by_thread` or `variables_by_thread` table was not thread safe and could yield incorrect results. (Bug #27471510)
- `INSERT ... ON DUPLICATE KEY UPDATE` could be handled improperly if a source table produced no rows. (Bug #27460607)
- `SET PERSIST_ONLY` ignored `--skip-grant-tables` and continued to perform checking against the dynamic privileges needed to set system variables. (Bug #27455943)
- For RPM packages and Docker RPM packages, the included `my.cnf` file now includes information indicating how to revert to the previous default authentication plugin (changing `caching_sha2_password` to `mysql_native_plugin`), for compatibility with older clients. (Bug #27454015, Bug #27675380)

- In string comparisons using `>`, only one operand was handled as `utf32`. (Bug #27452148)
- A server exit during `InnoDB` initialization caused AddressSanitizer (ASan) to report a memory leak. The patch also implements proper handling of the data dictionary properties table in cases where the table contains data that cannot be parsed. (Bug #27447981, Bug #89433)
- With `-DWITH_SSL=system`, if `CMake` could not find the system OpenSSL libraries and header files, it produced confusing messages. Now it exits immediately with an error. (Bug #27447874)
- Configuring with the `-DWITH_INNODB_EXTRA_DEBUG=ON` `CMake` option resulted in a linker error. (Bug #27444255, Bug #89412)
- Configuring with `-DWITH_ZLIB=system` resulted in a linker error. (Bug #27435371, Bug #89373)
- Metadata from result sets for `UNION ALL` queries could say `NEWDATE` rather than `DATE`. (Bug #27422376)
- In builds with Undefined Behavior Sanitizer enabled, negation of -922337203685477580 could cause a server exit. (Bug #27419181)
- With protocol compression enabled, a spurious assertion could be raised. (Bug #27418207, Bug #89324)

  References: This issue is a regression of: Bug #17922198.
- Performance related to fetching tablespace objects from the data dictionary to populate the data dictionary cache was improved. (Bug #27402243)
- If the server was started with `--skip-grant-tables`, clients that authenticate with the `caching_sha2_password` plugin were unable to connect. (Bug #27400095, Bug #89267)
- Linux RPM and Debian packages now include dependency information for the Perl JSON module required to run the MySQL test suite. Linux RPM packages now include dependency information for the Perl Digest module required to run the MySQL test suite. (Bug #27392800, Bug #89250, Bug #27392808, Bug #89244)
- A new system variable, `default_collation_for_utf8mb4`, is provided for internal use by replication. The system variable is set to the default collation for the `utf8mb4` character set. The value of the variable is replicated from a master to a slave so that the slave can correctly process data originating from a master with a different default collation for `utf8mb4`.The variable is primarily intended to support replication from a MySQL 5.7 or older master server to a MySQL 8.0 slave server, or group replication with a MySQL 5.7 primary node and one or more MySQL 8.0 secondaries. The default collation for `utf8mb4` in MySQL 5.7 is `utf8mb4_general_ci`, but it is `utf8mb4_0900_ai_ci` in MySQL 8.0. The variable is not present in releases earlier than MySQL 8.0, so if the slave does not receive a value for the variable, it assumes the master is from an earlier release, and sets the value to the previous default collation `utf8mb4_general_ci`.

  The default `utf8mb4` collation is used in the following statements:

  - `SHOW COLLATION` and `SHOW CHARACTER SET`.
  - `CREATE TABLE` and `ALTER TABLE` having a `CHARACTER SET utf8mb4` clause without a `COLLATION` clause, either for the table character set or for a column character set.
  - `CREATE DATABASE` and `ALTER DATABASE` having a `CHARACTER SET utf8mb4` clause without a `COLLATION` clause.

  - Any statement containing a string literal of the form `_utf8mb4'some text'` without a `COLLATE` clause.

  (Bug #27389878, Bug #27081073)

- `DESCRIBE` with a column name option ignored the column name. (Bug #27387773, Bug #89224)
- When run in key migration mode, the server ignored invalid options. (Bug #27387331)
- `SHOW GRANTS` for an anonymous user could cause a server exit. (Bug #27375069)
- Path name normalization could fail for very long path names. (Bug #27368298)
- During configuration, `CMake` assumed that `rpcgen` was available rather than checking for it. (Bug #27368078)
- The client authentication process could use memory after it had been freed. (Bug #27366143)
- `-DWITH_ZLIB=system` could cause other `CMake` feature tests to fail. (Bug #27356658, Bug #89135)
- On some systems, using `mysqld --initialize` to initialize the data directory was very slow. (Bug #27349579, Bug #89122)
- Builds using RPM source packages now use a secure connection if Boost must be downloaded. (Bug #27343289, Bug #89104)
- An internal `check_datetime_range` function that converted an ULONGLONG time value raised an assertion on Windows. (Bug #27340709)
- Too much locking was being done for the `caching_sha2_password` authentication plugin (one lock/unlock per connection). This is now improved to one lock/unlock per plugin install/uninstall. (Bug #27335346)
- A `CREATE TABLE ... LIKE` operation that implicitly assigned `'innodb_system'` as the tablespace name for table partitions raised an assertion. (Bug #27331588)
- `SET PERSIST_ONLY` could store large-valued variables incorrectly. (Bug #27322254)
- The `audit_log` plugin could write statements to the binary log even with binary logging disabled. (Bug #27315321)
- Certain queries using GREATEST() or LEAST() produced heap-use-after-free errors. (Bug #27312703)
- An `external_language` column was added to the `mysql.routines` data dictionary table to support stored routines in different languages. The data stored in this column can be accessed through the `EXTERNAL_LANGUAGES` column of the `INFORMATION_SCHEMA.ROUTINES` table. (Bug #27309116, Bug #89038)
- It is now prohibited to start the server with a `lower_case_table_names` setting that is different from the setting used when the server was initialized. The restriction is necessary because collations used by data dictionary table fields are determined by the setting defined when the server is initialized, and restarting the server with a different setting would introduce inconsistencies with respect to how identifiers are ordered and compared. (Bug #27309094, Bug #89035)
- For accounts that authenticated using the `auth_sock` authentication plugin, the server was unable to accept connections from clients from older MySQL versions. (Bug #27306178)
- An `audit_log` plugin memory leak was corrected. (Bug #27302151)

- The `INFORMATION_SCHEMA.FILES` table now obtains from storage engines the information needed for the following columns: `LOGFILE_GROUP_NAME`, `LOGFILE_GROUP_NUMBER`, `ROW_FORMAT`, `VERSION`. (Bug #27292586)
- Dropping a loadable function did not always remove its entry from the Performance Schema `user_defined_functions` table. (Bug #27270498)
- To reduce its size and storage footprint, serialized dictionary information (SDI) is now generated in a compact JSON format. (Bug #27265584)
- Concurrent execution of `RESET PERSIST` and `SET PERSIST` from multiple sessions could cause a server exit. (Bug #27264789)
- Tablespace statistics could be cached even if the statistics-collection process encountered an error. (Bug #27259963)
- `SET PERSIST` could fail to find the proper directory into which to write the `mysqld-auto.cnf` file. (Bug #27253828)
- `ALTER TABLE` could not create generated `NOT NULL` geometry columns if the table contained data. (Bug #27252609)
- The Performance Schema `variables_info` table displayed incorrect `VARIABLE_SOURCE` values for variables that were set in `my.cnf` and also persisted to `mysqld-auto.cnf`. (Bug #27252077)
- Improper handling of plugin loading and unloading could cause a server exit. (Bug #27247280, Bug #27297704)
- The `RESTART` statement did not work for Ubuntu packages. (Bug #27245918)
- LDAP authentication plugins were not built on FreeBSD. (Bug #27238252)
- The server did not handle queries correctly when a cached value was evaluated as `NULL` by a windowing function, or when its argument was part of a `GROUP BY` with a `ROLLUP` operation which could be evaluated as `NULL`. (Bug #27233287)
- Windowing functions such as `LAST_VALUE()` did not work correctly with tables containing `NOT NULL` columns in all cases. (Bug #27230463)
- Role cache invalidation could be performed incorrectly. (Bug #27225806)
- Incorrect handling of persisted variables at server startup could result in a server exit. (Bug #27224682)
- For upgrades from MySQL 5.7 to 8.0, the `sql_mode` value for object definitions could contain `NOT_USED`. (Bug #27219709)
- `JSON_TABLE()` failed if no default database was selected. (Bug #27217897)
- Adding a unique index to an `InnoDB` table on which multiple locks were held could raise an assertion. (Bug #27216817)
- `CREATE VIEW` statements that used `JSON_TABLE()` in the view definition failed if no database was selected. (Bug #27189940)
- The Doxygen documentation for `CLIENT_SSL_VERIFY_SERVER_CERT` was corrected. Thanks to Bradley Grainger for the patch. (Bug #27189363, Bug #88686)
- Queries on Performance Schema replication tables could return incorrect results, particularly when the execution plan used an index. (Bug #27184567)

- For some statements, the `FILE` privilege was not properly checked. (Bug #27160888)
- To better enable setting persisted variables at startup in the same order they were persisted with `SET PERSIST`, settings in `mysqld-auto.cnf` are first sorted based on timestamp (which is now stored in the file). (Bug #27157520)
- Setting the `dragnet.log_error_filter_rules` system variable to a very long value could cause a server exit. (Bug #27120953)
- For platforms that use systemd (see Managing MySQL Server with systemd), systemd was not always able to infer the state of a double-forked `mysqld` process. Consequently, systemd would attempt to restart `mysqld` even when that process terminated with status 1. systemd is now configured to run `mysqld` as a normal process (`Type=notify` rather than `Type=forking`). If the name of a socket file is specified in the environment variable `NOTIFY_SOCKET`, `mysqld` attempts to open a connection for communicating with systemd and writes its state changes there. (Bug #27109556, Bug #88463, Bug #26538598, Bug #87210)
- `ST_IsValid()` returned incorrect results for some geographic polygons on non-WGS 84 ellipsoids. (Bug #27074700)
- A multiple-insert statement on a table containing a `FULLTEXT` key and a `FTS_DOC_ID` column caused a server error. (Bug #27041445, Bug #88267)

  References: This issue is a regression of: Bug #22679185.
- In strict SQL mode, assignment of invalid values to `AUTO_INCREMENT` columns could be handled incorrectly, resulting in an assertion being raised. (Bug #27041393, Bug #88273)
- Thread stack exhaustion could raise an assertion rather than returning an error. (Bug #27041350, Bug #88277)
- `SET PERSIST_ONLY` did not properly consider whether the runtime validation function for persisted variables should be invoked. (Bug #27016247)
- The `audit_log` plugin could mishandle aborts of event executions, causing a server exit. (Bug #27008133)
- Some boundary cases for negation of large signed integers were corrected. (Bug #27004880, Bug #27008075)
- An `ALTER TABLE` operation attempted to set the `AUTO_INCREMENT` value for table in a discarded tablespace. (Bug #26935001)
- `MyISAM` index corruption could occur for bulk-insert and table-repair operations that involve the repair-by-sorting algorithm and many (more than 450 million) rows. (Bug #26929724, Bug #88003, Bug #28483283)
- The Performance Schema could produce `DIGEST_TEXT` values with a trailing space. This no longer occurs. (Bug #26908015)
- Dropping an index from a system table could cause a server exit. (Bug #26881798)
- A prepared statement using `CREATE TABLE ... SELECT` led to unexpected behavior when it referred in a `GROUP BY` to a view having the same name. (Bug #26881703)
- With auto-commit disabled and an XA transaction in PREPARED state, attempts to execute `XA COMMIT` or `XA ROLLBACK` failed. (Bug #26848877, Bug #87836)

- Some diagnostic messages produced by LDAP authentication plugins misleadingly suggested an error when no error had occurred. (Bug #26844713)
- Initialization code for `mysql_upgrade` and `mysqlpump` was reorganized to avoid assertion failures. (Bug #26802211)
- Full-text searches could raise an assertion due to improper handling of errors that occurred while attempting to acquire metadata locks. (Bug #26799898)
- A keyring file created by the `keyring_file` plugin on a 32-bit server was inaccessible by the `keyring_file` on a 64-bit server, and vice versa. (Bug #26793060)
- Metadata locks for column statistics were not displayed properly in the Performance Schema `metadata_locks` and `events_waits_xxx` tables. (Bug #26772858, Bug #87708)
- The `my_snprintf` plugin service was removed and reimplemented using C++11 `snprintf`. (Bug #26696147, Bug #87547)
- After an in-place upgrade from MySQL 5.7 to MySQL 8.0 on a server with a large number of tables, server startup failed to complete due to excessive memory consumption during data dictionary creation. (Bug #26486160)
- An error occurring in a statement that modifies user privileges could result in deadlock for other transactions attempting to access the user privilege cache. (Bug #26475282)
- The thread pool plugin logged too much information for failed connections. (Bug #26368725, Bug #86863)
- Enabling multiple components concurrently could result in a server exit. (Bug #26171471, Bug #86514)
- A malformed `mysqld-auto.cnf` file could cause a server exit. (Bug #26085774)
- Creating a table in a reserved tablespace did not return an error. (Bug #26073851, Bug #86309)
- For debug builds, using `KILL` to terminate a stored routine could raise an assertion. Thanks to Laurynas Biveinis for the patch. (Bug #26040870, Bug #86260)
- If the `init_connect` system variable was set, its contents could not be executed by clients with expired passwords, who therefore were prevented from connecting. Now, if a client has an expired password, `init_connect` execution is skipped, which enables the client to connect and change password. (Bug #25968185)
- Dates using the *`YYYYMMDD`* format were not recognized correctly in a query meeting all three of the following conditions:

  The query performed a left join.

  A `DATE` column in the inner table of the join was part of a multi-column primary key.

  Every column in the inner table's primary key was compared with another value; this could be either a literal or a column value. (Bug #25949639)
- An in-place `ALTER TABLE` operation on a table with foreign keys resulted in a table definition mismatch. The new table definition passed to storage engine methods during the `ALTER TABLE` execution contained invalid foreign key names. (Bug #25915132, Bug #85997)
- It was possible to assign nonexistent roles to an account as its default roles. (Bug #25755666, Bug #85561)

- Using the C API, when trying to execute an `INSERT` prepared statement with `CURSOR_TYPE_READ_ONLY` set, the client hung. (Bug #25701141, Bug #85105)
- `RENAME USER` failed even though the user or role to be renamed was not present in any role graph. (Bug #25658586)
- MySQL client programs could exit unexpectedly if malformed client/server protocol packets were received. (Bug #25471090)
- Memory statistics collected by the Performance Schema could be incorrect due to race conditions. (Bug #25212799)
- Incorrect handling by the `CONNECTION_CONTROL` plugin of an internal hash led to spurious messages in the error log and eventual server exit. (Bug #25052009)
- `CURRENT_ROLE()` and `ROLES_GRAPHML()` now return a string with the `utf8` character set. Previously, they incorrectly returned a binary string. (Bug #24843257)
- Killing `INSTALL COMPONENT` or `UNINSTALL COMPONENT` could result in multiple rows for a single component in the `mysql.component` system table. (Bug #24660436)
- Performing `SHOW PROCESSLIST` while running a high load concurrently using the X Plugin could lead to an unplanned server exit. (Bug #24621170)

  References: See also: Bug #23057045.
- The regular expression implementation in MySQL uses a `String` object—intended to be initialized as empty—to hold the current subject, and used the literal `""` for this value. This could interfere with other functions that do likewise. This is fixed by constructing the `String` with a `nullptr` instead. (Bug #23276471)
- `mysqlpump` could leak memory or exit when errors occurred. (Bug #23102944)
- LDML 2.8 collation definitions could cause unexpected server behavior. (Bug #22819030)
- Log-opening operations accessed log-name system variables without holding a lock on them. Multiple threads accessing such a variable could lead to a race condition and unexpected server behavior. (Bug #22591899)
- When the range optimizer computed the prefix for a string of sufficient length, it was possible for it to truncate the string in the middle of a character, which could lead to assertion and other failures in debug builds. Now steps are taken in such cases to make sure that the string is truncated at a character boundary. (Bug #22475473, Bug #13712655)
- Queries having subqueries or expressions in the `GROUP BY` clause could in some situations return random results, due to reading of uninitialized data. (Bug #20035304)

  References: See also: Bug #21922202.
- Queries having subqueries or expressions in a `GROUP BY` clause sometimes returned random results due to reading of uninitialized data. (Bug #20035304)

  References: See also: Bug #21922202.
- The `audit_log` plugin did not log placeholder values for prepared statements. (Bug #16617026)
- `ALTER TABLE ... RENAME` operations could lose foreign keys defined on the table if executed using the `COPY` algorithm. (Bug #11756183, Bug #48070)

- Integer columns in `UNION` statements could be cast to `BIGINT` even if a smaller integer type was more appropriate. (Bug #11747567, Bug #33004)
- Added `ER_REGEXP_INVALID_CAPTURE_GROUP_NAME` (Error 13110). (Bug #89796, Bug #27597980)
- Window function row-buffer handling has been refactored to reduce the number of handler reads by 25%. (Bug #89512, Bug #27484133)
- For some frames, the functions `VARIANCE()`, `VAR_POP()`, and `VAR_SAMP()` produced wrong results when evaluated with the `windowing_use_high_precision` system variable set to `false`. (Bug #89390, Bug #27438725)
- Generated columns with a prefix index were not considered when the optimizer attempted to substitute an expression with an equivalent generated column. This prevented use of prefix indexes to speed up queries with predicates employing a generated column expression. (Bug #89291, Bug #27403367)
- `FIRST_VALUE()` did not always return `NULL` for an empty frame. (Bug #89116, Bug #27348276)
- Regular expression functions accepting optional arguments did not return `NULL` as expected when one or more of these arguments was passed as an explicit `NULL`. (Bug #88872, Bug #27252630)
- Row-based replication used the wrong set of indexes on the slave. (Bug #88847, Bug #27244826)
- When executed with no default database having been chosen, `EXPLAIN EXTENDED` raised Error 1046 `No database selected` instead of the SQL syntax error expected due to the fact that the `EXTENDED` keyword has been removed. (Bug #88591, Bug #27153116)
- The CMake option `OPTIMIZER_TRACE=0` had no effect on builds and so is no longer used. (Bug #88520, Bug #27130109)
- A `SELECT` statement using `GROUP BY` with no constant or aggregate column failed a check for functional dependencies, correctly, on the first attempt but succeeded, wrongly, on the next. (Bug #88474, Bug #27427677)

  References: See also: Bug #21807579.
- A `CREATE TABLE ... SELECT` statement with a `UNION` in the `SELECT` failed in strict mode for a `DATE` column declared as `NOT NULL`. (Bug #87711, Bug #27068222)
- Prepared statements using nested sub-selects were not always handled correctly. (Bug #87484, Bug #26657904)
- The cost estimates for a query containing window functions did not take into account the cost of `PARTITION BY` or `ORDER BY`, nor did they include the cost of using the frame buffer. (Bug #87373, Bug #26612356)

  References: See also: Bug #26502118.
- When processing a query with a subquery, the subquery is no longer optimized if outer query is known to produce an empty result; in such cases, `EXPLAIN` now shows `Not optimized, outer query is empty`.

  This change is made because, when the table used by the outer query is found empty at optimization, this sets the join column to `NULL` even when defined as not nullable, which leads to a conflict when optimizing the subquery.

  An exception is made if the outer query has aggregates without any `GROUP BY`; this means that it has a non-empty result, and that any subquery in the `SELECT` list must be evaluated and optimized,

which reopens the nullability issue, which is now handled correctly for such cases. (Bug #83115, Bug #24713879)

References: See also: Bug #83216, Bug #24760317.

- When selecting from all columns making up a unique key containing nullable columns, with all columns in the `WHERE` condition set to non-null values, MySQL did not take into account their uniqueness, with the result that `only_full_group_by` missed detecting a functionally dependent column. (Bug #79291, Bug #22279903)
- When using a partial index, the optimizer performed a more expensive table lookup instead of using the index, even when the partial index covered the entire set of data needed. (Bug #74359, Bug #19806106)

# Changes in MySQL 8.0.5 - 8.0.10 (Skipped version numbers)

There are no release notes for these skipped version numbers.

# Changes in MySQL 8.0.4 (2018-01-23, Release Candidate)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

**Note**

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward. (Making a backup before the upgrade is a prudent precaution in any case.)

Beginning with MySQL 8.0.4, macOS 10.13 is a supported platform.

- Compilation Notes
- Component Notes
- Configuration Notes
- Deprecation and Removal Notes
- SQL Function and Operator Notes
- InnoDB Notes
- Installation Notes
- Keyring Notes
- Logging Notes
- Packaging Notes
- Performance Schema Notes
- Security Notes

- Server Administration
- Spatial Data Support
- Test Suite Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Compilation Notes

- **Linux:** Binary packages on Linux platforms now are built using GCC 6. The optimization level has changed from `-O3` to `-O2`. (WL #10343)
- **macOS; Microsoft Windows:** For compiling MySQL from source, the `-DWITH_SSL=system` `CMake` option now works on Windows and macOS. (Bug #26907731, Bug #87938)
- On platforms for which the GNU `gold` linker is used, removal of unused functions is now enabled, reducing the size of binaries. (Bug #26612067, Bug #87372)
- `#include` directives in source files were rewritten and reorganized to be unambiguous. (Bug #26597243, Bug #87358, Bug #26897738)
- The `BUILD` directory containing compilation scripts is no longer maintained and has been removed from MySQL source trees. (Bug #26576219, Bug #87323)
- The minimum version of the Boost library for server builds is now 1.65.0. (Bug #26574924, Bug #87317)
- MySQL can now be linked against OpenSSL 1.1 on Unix and Unix-like systems. (Bug #25094892, Bug #83814)
- Work was done to clean up the source code base, including: Removing unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

## Component Notes

- The `validate_password` plugin has been reimplemented to use the component infrastructure. To install and uninstall the `validate_password` component, use these statements:

  ```
  INSTALL COMPONENT 'file://component_validate_password';
  UNINSTALL COMPONENT 'file://component_validate_password';
  ```

  `INSTALL PLUGIN` and `UNINSTALL PLUGIN` still work to install and uninstall the `validate_password` plugin as before, but generate warnings. The plugin form of `validate_password` is deprecated and will be removed in a future MySQL version. MySQL installations that use the plugin should transition to the component instead. See Transitioning to the Password Validation Component.

  A usage difference between the component and plugin implementations of `validate_password` is that, whereas the plugin exposes system and status variables with names that begin with `validate_password_`, the component uses the convention of exposing variables with names that begin with the component name and a period. For example, the plugin exposes the `validate_password_policy` system variable and `validate_password_dictionary_file_words_count` status variable, whereas the corresponding variables exposed by the component are `validate_password.policy` and

`validate_password.dictionary_file_words_count`. The plugin variables, like the plugin itself, are deprecated and will be removed in a future MySQL version. (WL #6667)

## Configuration Notes

- **Replication:** When calculating the binary log expiration period, any value that you specify for the deprecated `expire_logs_days` system variable is now ignored if `binlog_expire_logs_seconds` is set, and only the value of `binlog_expire_logs_seconds` is used. Previously, the effects of `binlog_expire_logs_seconds` and `expire_logs_days` were cumulative.

  If you set a value for both system variables at startup, a warning message is issued stating that the value of `expire_logs_days` is ignored. You cannot set or change the value of one option dynamically while the other is set, and an error message is issued in this situation.

  The default binary log expiration period of 30 days (as changed in MySQL 8.0.2) applies if neither `binlog_expire_logs_seconds` nor `expire_logs_days` has a value set at startup. If a value for either `binlog_expire_logs_seconds` or `expire_logs_days` is set at startup, this value overrides the default binary log expiration period. To disable automatic purging of the binary log, you must set both `binlog_expire_logs_seconds` and `expire_logs_days` explicitly to 0 at startup. (Bug #26274274, Bug #86698)

- For RHEL, SLES, and Fedora RPMs, the default plugin directory for debug builds has been changed from `/usr/lib64/mysql/plugin` to `/usr/lib64/mysql/plugin/debug`. (Bug #27072155, Bug #88363)

- The new `WITH_LZMA` and `WITH_RE2` `CMake` options control whether to compile with the bundled or system LZMA and RE2 libraries.

- The `table_open_cache` system variable default variable has been increased from 2000 to 4000. This additionally changes the default for the autosized `table_definition_cache` system variable from 1400 to 2000. (WL #9703)

- The `optimizer_trace_max_mem_size` system variable default value was changed from 16KB to 1MB to lessen the likelihood of optimizer trace truncation. (WL #11345)

- The `log_error_verbosity` system variable default value was changed from 3 (error, warning, and information messages) to 2 (error and warning messages) to make `mysqld` error logging less verbose by default. (WL #11143)

## Deprecation and Removal Notes

- **InnoDB:** The `innodb_undo_tablespaces` configuration option is deprecated and will be removed in a future MySQL version. (WL #10473)

- **Replication:** The `group_replication_primary_member` status variable has been deprecated and will be removed in a future MySQL version. (WL #10958)

- **Replication:** The `group_replication_allow_local_disjoint_gtids_join` system variable has been removed. (WL #11139)

- When the `libmysqld` embedded server library was removed in MySQL 8.0.1, the following `mysql_options()` options used only for `libmysqld` were not removed. They have now been removed.

```
MYSQL_OPT_GUESS_CONNECTION
MYSQL_OPT_USE_EMBEDDED_CONNECTION
MYSQL_OPT_USE_REMOTE_CONNECTION
MYSQL_SET_CLIENT_IP
```

(Bug #26712418)

- `mysqlpump` no longer includes the `SQL_NO_CACHE` modifier in statements because that modifier is now deprecated and results in deprecation warnings. (Bug #26694675)
- Generated columns no longer permit deprecated functions in the generation expression, to avoid problems when MySQL is upgraded to a version in which the deprecated functions have been removed. (Bug #26279884, Bug #86712)
- The Performance Schema `setup_timers` table has been removed, as has the `TICK` row in the `performance_timers` table. (Bug #18296337, WL #10985, WL #10986)

## SQL Function and Operator Notes

- **Incompatible Change:** Previously, MySQL used the Henry Spencer regular expression library to support regular expression operators (`REGEXP`, `RLIKE`). Regular expression support has been reimplemented using International Components for Unicode (ICU), which provides full Unicode support and is multibyte safe. The `REGEXP_LIKE()` function performs regular expression matching in the manner of the `REGEXP` and `RLIKE` operators, which now are synonyms for that function. In addition, the `REGEXP_INSTR()`, `REGEXP_REPLACE()`, and `REGEXP_SUBSTR()` functions are available to find match positions and perform substring substitution and extraction, respectively. The `regexp_stack_limit` and `regexp_time_limit` system variables provide control over resource consumption by the match engine.

  MySQL source distributions bundle the ICU library, and the `WITH_ICU CMake` option controls whether to compile with the bundled or system ICU library. The `ICU_VERSION()` function returns the ICU library version.

  For more information, see Regular Expressions. For information about ways in which applications that use regular expressions may be affected by the implementation change, see Regular Expression Compatibility Considerations. (WL #8987)
- Two new SQL functions are available to provide digest information about SQL statements. Given an SQL statement as a string, `STATEMENT_DIGEST()` returns the statement digest hash value, and `STATEMENT_DIGEST_TEXT()` returns the normalized statement digest. See Encryption and Compression Functions. (WL #9637)

## InnoDB Notes

- The restriction enacted in MySQL 8.0.3 against renaming of columns in a parent foreign key has been lifted. (Bug #26659110, Bug #87490, Bug #25722221)

  References: See also: Bug #26334071.

## Installation Notes

- For platforms that use systemd (see Managing MySQL Server with systemd), the data directory is initialized if empty at server startup. This might be a problem if the data directory is a remote mount that has temporarily disappeared: The mount point would appear to be an empty data directory, which then would be initialized as a new data directory. It is now possible to suppress this automatic initialization behavior. Specify the following line in the `/etc/sysconfig/mysql` file (create the file if it does not exist):

  ```
  NO_INIT=true
  ```

  (Bug #26595288, Bug #87287)

## Keyring Notes

- MySQL now supports key migration between underlying keyring keystores, permitting DBAs to switch a MySQL installation from one keyring plugin to another. See Migrating Keys Between Keyring Keystores. (WL #9769)

## Logging Notes

- A new error log filter component, `log_filter_dragnet`, enables control over events written to the error log, based on user-defined rules specified as the value of the `dragnet.log_error_filter_rules` system variable. This new filtering capability is more flexible than is achievable using the built-in `log_filter_internal` filter component and `log_error_verbosity` system variable. For more information, see Types of Error Log Filtering. (WL #9651)

- Messages written to the error log by the `log_sink_internal` log sink component now contain an error-ID indicator. This ID has a format of `[error_id]`. It follows the severity indicator and precedes the message text. For more information, see Error Log Output Format. (WL #11009)

- When the `log_error_verbosity` system variable for error logging is set to filter out warning and information messages, selected important messages about non-error situations are now printed to error logs as system messages. These messages were previously handled and labeled as errors so that they were logged when `log_error_verbosity=1` was set. The labeling function has now been decoupled from the handling function for these messages, so that the messages can be given an appropriate label.

  System messages now have the label “System” in the MySQL error log. Other log sinks that support custom labels might also use this convention. In logs that do not support custom labels, system messages are printed with the label used for the information level of severity. However, they are included even if the `log_error_verbosity` setting normally excludes messages at this level.

  If your system has additional configuration to discard messages labeled as information level, or to redirect them to a different destination from messages labeled as errors, system messages do not override this behavior. If the log sink involved does not use the “System” label, system messages are now discarded or redirected along with other messages labeled as information level.

  The messages that are now printed with the “System” label where supported include startup and shutdown messages, and some significant changes to settings, as follows:

  - `ER_STARTING_AS`
  - `ER_STARTUP`
  - `ER_XA_STARTING_RECOVERY`
  - `ER_XA_RECOVERY_DONE`
  - `ER_NORMAL_SHUTDOWN`
  - `ER_SHUTDOWN_COMPLETE`
  - `ER_RPL_SLAVE_CONNECTED_TO_MASTER_REPLICATION_STARTED`
  - `ER_RPL_SLAVE_CONNECTED_TO_MASTER_REPLICATION_RESUMED`
  - `ER_SLAVE_CHANGE_MASTER_TO_EXECUTED`
  - `ER_GTID_PURGED_WAS_CHANGED`

- `ER_GTID_EXECUTED_WAS_CHANGED`
- `ER_CHANGED_GTID_MODE`
- `ER_RPL_SLAVE_DUMP_THREAD_KILLED_BY_MASTER`

(WL #10942)

## Packaging Notes

- All MySQL binary distributions now are linked against OpenSSL, including Community distributions, which previously were linked against yaSSL. In addition, OpenSSL is linked dynamically rather than statically, which enables substitution of alternative SSL libraries for use with MySQL if desired. For some platforms, binary distributions bundle OpenSSL libraries to ensure library availability:
  - Windows: Distributions bundle `libeay32.dll` and `ssleay32.dll`, which are installed in the same directory as MySQL binaries. Other libraries can be used by replacing those library files with alternatives, or by using some other library-selection method supported on Windows. (See https://msdn.microsoft.com/en-us/library/7d83bc18.aspx.)
  - macOS: Distributions bundle `libssl.dylib` and `libcrypto.dylib`. MySQL binaries are linked to expect the libraries in the same directory, and symbolic links there point to the actual library locations.
  - Linux (for generic compressed `tar` file distributions only): Distributions bundle `libssl.so` and `libcrypt.so`,which are installed in the same directory as the `libmysqlclient.so` client library.

  (Bug #26272084, Bug #26134893, Bug #26927607, Bug #87996, WL #10524)

## Performance Schema Notes

- The Performance Schema now uses SHA-256 hashes for statement digests rather than MD5 hashes. To accommodate the increased storage required for SHA-256 values, `DIGEST` columns in Performance Schema tables are widened from `VARCHAR(32)` to `VARCHAR(64)`. (Bug #26727443)

## Security Notes

- **Incompatible Change:** Passwords are now restricted to a maximum of 256 characters for the `caching_sha2_password` and `sha256_password` authentication plugins, and for the `PASSWORD()` function when `old_passwords=2`. Also, the number of password hashing rounds is capped to limit CPU time used. (Bug #27099029, Bug #27194270)
- **Incompatible Change:** The `caching_sha2_password` and `sha256_password` authentication plugins provide more secure password encryption than the `mysql_native_password` plugin, and `caching_sha2_password` provides better performance than `sha256_password`. Due to these superior security and performance characteristics of `caching_sha2_password`, it is now the preferred authentication plugin, and is also the default authentication plugin rather than `mysql_native_password`. This change affects both the server and the `libmysqlclient` client library:
  - For the server, the default value of the `default_authentication_plugin` system variable changes from `mysql_native_password` to `caching_sha2_password`.
  - The `libmysqlclient` library treats `caching_sha2_password` as the default authentication plugin rather than `mysql_native_password`.

This change has the following implications:

- The change affects only the authentication plugin used for creating new MySQL accounts. For accounts already existing in an upgraded installation, their authentication plugin remains unchanged.
- Clients that use an account that authenticates with `caching_sha2_password` must use either a secure connection (made using TCP using TLS/SSL credentials, a Unix socket file, or shared memory), or an unencrypted connection that supports password exchange using an RSA key pair. This security requirement does not apply to `mysql_native_passsword`, so the switch to `caching_sha2_password` may require additional configuration (see Caching SHA-2 Pluggable Authentication). However, client connections in MySQL 8.0 prefer use of TLS/SSL by default, so clients that already conform to that preference may need no additional configuration.
- Because `caching_sha2_password` is also now the default authentication plugin in the `libmysqlclient` client library, authentication requires an extra round trip in the client/server protocol for connections from MySQL 8.0 clients to accounts that use `mysql_native_password` (the previous default authentication plugin), unless the client program is invoked with a `--default-auth=mysql_native_password` option.
- Incompatibility: Clients and connectors that have not been updated to know about `caching_sha2_password` *cannot* connect to accounts that authenticate with `caching_sha2_password` because they do not recognize this plugin as valid. To work around this issue, relink clients against `libmysqlclient` from MySQL 8.0.4 or higher, or obtain an updated connector that recognizes `caching_sha2_password`.
- Incompatibility: Clients and connectors that have not been updated to know about `caching_sha2_password` may have trouble connecting to a MySQL 8.0 server configured with `caching_sha2_password` as the default authentication plugin, even to use accounts that do not authenticate with `caching_sha2_password`. This issue occurs because the server specifies the name of its default authentication plugin to clients. If a client or connector is based on a client/server protocol implementation that does not gracefully handle an unrecognized default authentication plugin, it may fail with an error.

For more information about the more prominent role of `caching_sha2_password`, including discussion of potential compatibility issues and solutions, see caching_sha2_password as the Preferred Authentication Plugin.

Additionally, replication slaves are now able to connect to masters using RSA key pair-based password exchange. This RSA capability applies for accounts that authenticate with the `caching_sha2_password` or `sha256_password` authentication plugin. (Previously, slaves could authenticate with those plugins, but required a secure connection and could not use RSA password exchange.) The following changes enable use of this new slave RSA capability:

- The `CHANGE MASTER TO` statement has clauses for specifying RSA public key information.
- For Group Replication, the `group_replication_recovery_public_key_path` and `group_replication_recovery_get_public_key` system variables serve the same purpose.
- The `SHOW SLAVE STATUS` statement and the Performance Schema `replication_connection_configuration` table display replication slave RSA public key information.

More programs now support a `--server-public-key-path` option to enable specifying a client-side file containing the public key for RSA key pair-based password exchange, for accounts that authenticate with the `caching_sha2_password` or `sha256_password` authentication plugin: `mysqladmin`,

`mysqlbinlog`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlpump`, `mysqlshow`, `mysqlslap`, `mysql_upgrade`. Previously, only `mysql` and `mysqltest` supported `--server-public-key-path`. The `--server-public-key-path` option is an alternative to `--get-server-public-key`, and may be used when a client-side file is available that contains a copy of the RSA public key required by the server, rather than sending a request for the public key to the server.

The new `mysql_reset_server_public_key()` C API function clears any cached copy of the server RSA public key from the client library. See mysql_reset_server_public_key().

The new `caching_sha2_password_auto_generate_rsa_keys` system variable enables automatic RSA private/public key-pair file generation, similar to the `sha256_password_auto_generate_rsa_keys` system variable. See Automatic SSL and RSA File Generation.

The new `Caching_sha2_password_rsa_public_key` status variable exposes the public key used by the `caching_sha2_password` authentication plugin for RSA key pair-based password exchange. (Bug #26751594, WL #11057)

- The linked OpenSSL library for MySQL Server has been updated to version 1.0.2n. Issues fixed in the new OpenSSL version are described at http://www.openssl.org/news/vulnerabilities.html. (Bug #27212666, Bug #27236394)

- Certificates automatically generated by `mysqld` and `mysql_ssl_rsa_setup` now use X509 v3 rather than v1. (Bug #26521654)

## Server Administration

- A new `RESTART` SQL statement is available that enables restarting a MySQL server instance from within a client session. This feature is available for platforms on which a monitoring process is able to detect a server shutdown performed for restart purposes: Windows (`mysqld` started as a Windows service or standalone), and Unix and Unix-like systems that use systemd or `mysqld_safe` to manage `mysqld`. See RESTART Statement. (WL #9809)

## Spatial Data Support

- **Incompatible Change:** These spatial import functions now return an error for data in a geographic spatial reference system for which the coordinates are outside the legal range of values: `ST_GeomCollFromText()`, `ST_GeomCollFromTxt()`, `ST_GeomCollFromWKB()`, `ST_GeometryCollectionFromText()`, `ST_GeometryCollectionFromWKB()`, `ST_GeometryFromText()`, `ST_GeometryFromWKB()`, `ST_GeomFromGeoJSON()`, `ST_GeomFromText()`, `ST_GeomFromWKB()`, `ST_LineFromText()`, `ST_LineFromWKB()`, `ST_LineStringFromText()`, `ST_LineStringFromWKB()`, `ST_MLineFromText()`, `ST_MLineFromWKB()`, `ST_MPointFromText()`, `ST_MPointFromWKB()`, `ST_MPolyFromText()`, `ST_MPolyFromWKB()`, `ST_MultiLineStringFromText()`, `ST_MultiLineStringFromWKB()`, `ST_MultiPointFromText()`, `ST_MultiPointFromWKB()`, `ST_MultiPolygonFromText()`, `ST_MultiPolygonFromWKB()`, `ST_PointFromText()`, `ST_PointFromWKB()`, `ST_PolyFromText()`, `ST_PolyFromWKB()`, `ST_PolygonFromText()`, `ST_PolygonFromWKB()`. (`ST_LatFromGeoHash()`, `ST_LongFromGeoHash()`, and `ST_PointFromGeoHash()` do not have this behavior because they cannot represent out-of-range values.)

  `ST_GeomFromGeoJSON()` also now returns an error if a `crs` member that specifies an SRID different from the top-level object SRID is found at a lower level of the GeoJSON document.

**Note**

If spatial data contains SRID values that refer to a geographic spatial reference system and the data coordinates are out of range, existing queries using these functions will return different results, compared to previous MySQL versions.

(Bug #26941370, Bug #88031, WL #9449)

- **Incompatible Change:** Previously, these spatial functions ignored the spatial reference system (SRS) for geometry arguments and computed results on a Cartesian plane. They now support computations for geometry arguments that specify a geographic SRS: `ST_Distance_Sphere()`, `ST_IsSimple()`, `ST_IsValid()`, `ST_Length()`.

  Previously, these spatial functions ignored the SRS for any geometry arguments and computed results on a Cartesian plane. They now produce an error when invoked with geometry arguments that specify a geographic SRS: `ST_Area()`, `ST_Buffer()`, `ST_Centroid()`, `ST_ConvexHull()`, `ST_Difference()`, `ST_Envelope()`, `ST_Intersection()`, `ST_IsClosed()`, `ST_MakeEnvelope()`, `ST_Simplify()`, `ST_SymDifference()`, `ST_Union()`, `ST_Validate()`.

  Previously, these spatial functions permitted geometry arguments with an undefined SRS. They now produce an error when invoked with geometry arguments that have an undefined SRS: `ST_Dimension()`, `ST_Distance_Sphere()`, `ST_EndPoint()`, `ST_ExteriorRing()`, `ST_GeometryN()`, `ST_GeometryType()`, `ST_InteriorRingN()`, `ST_IsEmpty()`, `ST_IsSimple()`, `ST_IsValid()`, `ST_Length()`, `ST_NumGeometries()`, `ST_NumInteriorRing()`, `ST_NumInteriorRings()`, `ST_NumPoints()`, `ST_PointN()`, `ST_StartPoint()`, `ST_SwapXY()`, `ST_X()`, `ST_Y()`.

  Previously, the `ST_GeoHash()` spatial function accepted points with any SRID. `ST_GeoHash()` now accepts only points with SRID 0 or 4326.

  **Note**

  If spatial data contains geometry values that now are interpreted differently by the functions just listed, existing queries using these functions will return different results, compared to previous MySQL versions.

  (WL #10416, WL #10926, WL #10927, WL #10999, WL #11096)

- **Incompatible Change:** For the following spatial functions that take an SRID argument, they now return an error if the SRID does not exist: `ST_GeomCollFromText()`, `ST_GeomCollFromTxt()`, `ST_GeomCollFromWKB()`, `ST_GeometryCollectionFromText()`, `ST_GeometryCollectionFromWKB()`, `ST_GeometryFromText()`, `ST_GeometryFromWKB()`, `ST_GeomFromGeoJSON()`, `ST_GeomFromText()`, `ST_GeomFromWKB()`, `ST_LineFromText()`, `ST_LineFromWKB()`, `ST_LineStringFromText()`, `ST_LineStringFromWKB()`, `ST_MLineFromText()`, `ST_MLineFromWKB()`, `ST_MPointFromText()`, `ST_MPointFromWKB()`, `ST_MPolyFromText()`, `ST_MPolyFromWKB()`, `ST_MultiLineStringFromText()`, `ST_MultiLineStringFromWKB()`, `ST_MultiPointFromText()`, `ST_MultiPointFromWKB()`, `ST_MultiPolygonFromText()`, `ST_MultiPolygonFromWKB()`, `ST_PointFromGeoHash()`,

`ST_PointFromText()`, `ST_PointFromWKB()`, `ST_PolyFromText()`, `ST_PolyFromWKB()`, `ST_PolygonFromText()`, `ST_PolygonFromWKB()`. (`ST_SRID()` already had this behavior.)

**Note**

If spatial data contains SRID values that refer to a nonexistent spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

(WL #9450)

- The server now prints a warning if a `SPATIAL` index is created that the optimizer will never use. (Bug #27015964)
- The EPSG Dataset containing spatial reference system data for spatial calculations has been upgraded from version 9.0 to 9.2. (Bug #26711258, Bug #87564, Bug #26990896)
- MySQL now supports `CREATE SPATIAL REFERENCE SYSTEM` and `DROP SPATIAL REFERENCE SYSTEM` statements for creating and removing spatial reference system (SRS) definitions in the data dictionary. See CREATE SPATIAL REFERENCE SYSTEM Statement, and DROP SPATIAL REFERENCE SYSTEM Statement. With the implementation of these statements, direct access to the `mysql.st_spatial_reference_systems` data dictionary table using statements such as `SELECT`, `INSERT`, and `UPDATE` is not needed, and is no longer permitted. Read access to SRS definitions remains available through the `INFORMATION_SCHEMA ST_SPATIAL_REFERENCE_SYSTEMS` table, as before. (WL #9059)

## Test Suite Notes

- Documentation for the MySQL Test Suite is now maintained in the MySQL source tree using Doxygen (see the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html.) The related Unix man pages that previously were produced from the old test suite manual are no longer updated and have gone out of date. Consequently, they are no longer included in MySQL distributions. (Bug #27021754)
- The MySQL test suite now includes `CRC32()` tests. Thanks to Daniel Black for the patch. (Bug #26495791, Bug #87136)
- `mysqltest` now accepts an optional *`retry`* argument for these commands: `copy_file` `copy_files_wildcard`, `file_exists`, `move_file`, `remove_file`, and `remove_files_wildcard`. Usage is as follows:

```
--file_operation file_name ... [retry]
```

When the *`retry`* argument is specified, the command retries a failed operation the given number of times at one-second intervals. (Bug #24671890)

## X Plugin Notes

- X Plugin could incorrectly remove quotes from parameters that were passed to functions expecting JSON strings, causing an error. (Bug #26906519)
- After an index was created on a collection, X Plugin would list the collection as a relational table, and no longer list it as a collection. (Bug #26906487)
- X Protocol displayed the `DATE` type in the same way as the `DATETIME` type. The two data types are now treated differently. (Bug #26647488)
- `mysqlxtest` did not display fractional seconds in `DATETIME` values correctly. (Bug #26638422)

- X Plugin returned a column metadata field showing fractional digits for a JSON column, for which that information was irrelevant. The field is no longer returned. (Bug #26258481)
- X Plugin handled user account lookup incorrectly if the IP address resolved to an empty host name. (Bug #26042786)
- X Plugin can now be configured to abort any authenticated connections which are considered idle. The following system variables have been added:
  - `mysqlx_interactive_timeout`
  - `mysqlx_read_timeout`
  - `mysqlx_wait_timeout`
  - `mysqlx_write_timeout`

  (WL #9267)
- X Plugin now supports Caching SHA-2 Pluggable Authentication. To enable the support, issue:

  ```
  INSTALL PLUGIN mysqlx_cache_cleaner SONAME 'mysqlx.so';
  ```

  This plugin maintains the cache and removes all entries for accounts that were modified. (WL #10992)

  References: See also: Bug #27213213, Bug #27042109.

## Functionality Added or Changed

- **InnoDB:** Obsoleted `InnoDB` system tables that are no longer required after upgrading to MySQL 8.0.3 or later are now dropped after a successful upgrade. (Bug #26757171)
- **InnoDB:** Support was added to automatically upgrade data dictionary table definitions when starting the MySQL server after upgrading MySQL binaries to a new version. At startup, the data dictionary version of the server is compared to the version information stored in the data dictionary to determine if data dictionary tables should be upgraded. If an upgrade is necessary and supported, the server creates data dictionary tables with updated definitions, copies persisted metadata to the new tables, atomically replaces the old tables with the new ones, and reinitializes the data dictionary. If an upgrade is not necessary, startup continues without updating the data dictionary tables.

  The `mysqld --no-dd-upgrade` option can be used to prevent automatic upgrade of data dictionary tables at startup. (WL #9553)
- **InnoDB:** Moving or restoring tablespace files to a new location while the server is offline is supported by the new `--innodb-directories` option, which defines directories to scan at startup for tablespace files. For more information, see Moving Tablespace Files While the Server is Offline.

  With the introduction of the `--innodb-directories` feature, the location of file-per-table and general tablespace files created with an absolute path or in a location outside of the data directory should be added to the `innodb_directories` argument value. Otherwise, `InnoDB` is not able to locate these files during recovery. To view tablespace file locations, query the `INFORMATION_SCHEMA.FILES` table:

  ```
  mysql> SELECT TABLESPACE_NAME, FILE_NAME FROM INFORMATION_SCHEMA.FILES \G
  ```

  The `CREATE TABLESPACE ... ADD DATAFILE` statement now requires that a specified directory is known to `InnoDB`. Known directories include those implicitly and explicitly defined by the `innodb_directories` option.

`innodb_directories` replaces `innodb_scan_directories`, which has been removed. (WL #8619)

- **InnoDB:** `InnoDB` now supports partial update of large object (LOB) data stored in external fields outside of clustered index pages, including `JSON` documents updated using `JSON_SET()` and `JSON_REPLACE()`. Previously, LOB values could only be read or modified in full, and updates of `JSON` column values were done by completely removing the previous document and writing the new one in its place.

  Partial update uses an internal LOB index, or ZLOB index in the case of compressed LOB data, which is created and stored in one or more LOB pages when a row containing LOB data is inserted or updated. Thus, partial fetch and update of LOB data is supported for newly inserted rows, but existing rows containing LOB data must be updated to add partial fetch and update support. The addition of LOB index data increases the storage space required by LOB values by a small percentage.

  This feature adds the following `InnoDB` page types for storing uncompressed and compressed LOB data:

  - FIL_PAGE_TYPE_LOB_INDEX
  - FIL_PAGE_TYPE_LOB_DATA
  - FIL_PAGE_TYPE_LOB_FIRST
  - FIL_PAGE_TYPE_ZLOB_FIRST
  - FIL_PAGE_TYPE_ZLOB_DATA
  - FIL_PAGE_TYPE_ZLOB_INDEX
  - FIL_PAGE_TYPE_ZLOB_FRAG
  - FIL_PAGE_TYPE_ZLOB_FRAG_ENTRY

  (WL #8960)

- **InnoDB:** The storage engine private field of the `mysql.tablespaces` data dictionary table and page 0 of `InnoDB` tablespace files now store MySQL server version and tablespace version information. The MySQL server version is the version that created the tablespace, the version into which the tablespace was imported, or the version of the last major MySQL upgrade. The tablespace version tracks tablespace format changes. `SERVER_VERSION` and `SPACE_VERSION` fields were added to the `INFORMATION_SCHEMA.INNODB_TABLESPACES` table to provide access to this data. (WL #5989, WL #11063)

- **Group Replication:** Host names can now be specified as part of a whitelist for group replication connections, using the `group_replication_ip_whitelist` system variable. Host names support CIDR notation. Host names that resolve to IPv6 addresses are not supported.

  For host names, name resolution takes place only when a connection request is made by another server. A host name that cannot be resolved is not considered for whitelist validation, and a warning message is written to the error log. Forward-confirmed reverse DNS (FCrDNS) verification is carried out for resolved host names.

  **Warning**

  Host names are inherently less secure than IP addresses in a whitelist. FCrDNS verification provides a good level of protection, but can be compromised by

> certain types of attack. Specify host names in your whitelist only when strictly necessary, and ensure that all components used for name resolution, such as DNS servers, are maintained under your control. You can also implement name resolution locally using the hosts file, to avoid the use of external components.

(WL #10803, WL #11298)

- **JSON:** Added the `JSON_TABLE()` function, which accepts `JSON` data and returns it as a relational table whose columns are as specified. This virtual table can be accessed using standard SQL mechanisms

  This function has the syntax `JSON_TABLE(expr, path COLUMNS column_list) [AS] alias)`, where *`expr`* is an expression that returns JSON data, *`path`* is a JSON path applied to the source, and *`column_list`* is a list of column definitions. Columns can be of the types `FOR ORDINAL`, `PATH`, `EXISTS PATH`, and `NESTED PATH` or `NESTED`, as described in the following list:

  - `FOR ORDINAL`: The column is a counter, similar to an `AUTO_INCREMENT` column.
  - `PATH`: The column holds a scalar value using the specified JSON path. `ON ERROR` and `ON EMPTY` options are supported for handling illegal values (such as nonscalars) and empty values, respectively.
  - `EXISTS PATH`: The column value is 1 if a match exists for the specified JSON path, and 0 otherwise.
  - `NESTED PATH`: Nested objects or arrays in JSON data found in the given JSON path are flattened into a single row along with the JSON values from the parent object or array. The `PATH` keyword is optional.

  Two simple examples are shown here:

```
mysql> SELECT *
    -> FROM
    ->   JSON_TABLE(
    ->     '[1, 3, 5, 7, 9]',
    ->     "$[*]" COLUMNS(rowid FOR ORDINALITY, col VARCHAR(50) PATH "$")
    ->   ) AS t1;
+-------+------+
| rowid | col  |
+-------+------+
|     1 | 1    |
|     2 | 3    |
|     3 | 5    |
|     4 | 7    |
|     5 | 9    |
+-------+------+

mysql> SET @j = '[{"a": [{"x":"3"},{"y":"2"}]},
    '>   {"b": [{"x":"1"},{"y":"1"}]},
    '>   {"a": [{"x":"2"},{"y":"3"}]}]';

mysql> SELECT *
    -> FROM  JSON_TABLE(
    ->         @j,
    ->         "$[*]"  COLUMNS(
    ->                   rowid FOR ORDINALITY,
    ->
    ->                   xa INT EXISTS PATH "$.a",
    ->                   xb INT EXISTS PATH "$.b",
    ->
    ->                   ja JSON PATH "$.a",
    ->                   jb JSON PATH "$.b"
    ->                 )
    ->       ) AS jts;
+-------+------+------+-------------------------+-------------------------+
```

```
| rowid | xa   | xb   | ja                       | jb                       |
+-------+------+------+--------------------------+--------------------------+
|     1 |    1 |    0 | [{"x": "3"}, {"y": "2"}] | NULL                     |
|     2 |    0 |    1 | NULL                     | [{"x": "1"}, {"y": "1"}] |
|     3 |    1 |    0 | [{"x": "2"}, {"y": "3"}] | NULL                     |
+-------+------+------+--------------------------+--------------------------+
```

The JSON source expression can be any expression that yields a valid JSON document, including a JSON literal, a table column, or a function call that returns JSON such as `JSON_EXTRACT(t1, data, '$.post.comments')`. For complete syntax and other information about this function, see JSON Table Functions. (WL #8867)

- The `-DWITH_ASAN_SCOPE` `CMake` option enables the AddressSanitizer `-fsanitize-address-use-after-scope` Clang flag for use-after-scope detection. The default is off. To use this option, `-DWITH_ASAN` must also be enabled. (Bug #27095089)
- Handling of empty `JSON` documents has been made more robust. (Bug #26787468, Bug #87734)
- MySQL 8.0.3 imposed a restriction against `ALTER TABLE RENAME` on tables in a foreign key relationship if a `LOCK TABLES` was active. This restriction has been lifted. (Bug #26647340, Bug #87467)
- If the server PID file is configured to be created in a world-writable location, the server now issues a warning suggesting use of a more secure location. (Bug #26585560)
- For constructs using *`_charset`* character set introducers (for example, `_utf8mb4 'abc'`), lookup performance was improved for mapping the introducer onto the proper character set. (Bug #25680866, Bug #85331)

## Bugs Fixed

- **Important Change:** The following changes are made to the `PERIOD_ADD()` and `PERIOD_DIFF()` functions:
  - A period value used with one of these functions may not be negative.
  - The month part of a period value may not be equal to 0.

  A period value used with one of these functions for which at least one of these conditions is true now causes the function to fail with an error. (Bug #27004699, Bug #27004729)
- **Important Change:** The `LEAST()` and `GREATEST()` functions no longer attempt to infer a context for their arguments from expressions in which they are used. For example, `LEAST('11', '45', '2')` returns `'11'`, but `LEAST('11', '45', '2') + 0` treated the function arguments as integers rather than as strings, and returned `2`. Now these functions always evaluate their arguments strictly according to type, and any data type coercion due to their inclusion in an expression is performed only on the result returned by the function. This means that the expression `LEAST('11', '45', '2') + 0` now evaluates to `'11' + 0`, and thus to the integer value `11`.

  This change has been made due to the following considerations:
  - Rules for deriving the context were not always clear or consistent.
  - The results of these functions when used in expressions were not consistent with the results of `COALESCE()`, or of a `UNION` query.

  Applications that use these functions within expressions should be checked to make sure that they do not depend on the previous behavior, and updated if they do so. (Bug #83895, Bug #25123839)

- **InnoDB:** Concurrent XA transactions that ran successfully to the XA prepare stage on the master conflicted when replayed on the slave, resulting in a lock wait timeout in the applier thread. The conflict was due to the GAP lock range which differed when the transactions were replayed serially on the slave. To prevent this type of conflict, GAP locks taken by XA transactions in `READ COMMITTED` isolation level are now released (and no longer inherited) when XA transactions reach the prepare stage. (Bug #27189701, Bug #25866046)

- **InnoDB:** An `ALTER TABLE` operation that added a foreign key constraint referencing a table with generated virtual columns raised an assertion. (Bug #27189701)

- **InnoDB:** A `DROP DATABASE` operation raised an assertion due to a missing general tablespace data file. (Bug #27151163)

- **InnoDB:** On Windows, an operation that altered a table partition raised an assertion. The table name was not parsed correctly. (Bug #27075816)

- **InnoDB:** A `TRUNCATE TABLE` operation on a temporary table raised an assertion. (Bug #27073280)

- **InnoDB:** A call to a recovery-related function during the post-DDL phase of a DDL operation raised an assertion. (Bug #27041487, Bug #88263)

- **InnoDB:** Adding a spatial index that referenced a column with an SRID attribute returned an error. (Bug #27021029)

- **InnoDB:** A table with a 64-character foreign key name caused an upgrade failure. Foreign key names up to 64 characters in length should be permitted. (Bug #27014308, Bug #88196)

- **InnoDB:** The `InnoDB` recovery process failed with a tablespace size error for a compressed table that was upgraded from MySQL 5.7 to MySQL 8.0.

  The tablespace file for a compressed table is now created using the physical page size instead of the `InnoDB` page size, which makes the initial size of a tablespace file for an empty compressed table smaller than in previous MySQL releases. (Bug #27014083, Bug #88195)

- **InnoDB:** Unnecessary tablespace fetch and cache update operations caused a server startup delay. (Bug #26995951)

  References: This issue is a regression of: Bug #26832347.

- **InnoDB:** An orphan `.frm` file caused an upgrade failure, and subsequent upgrade attempts were unsuccessful due to a full-text search auxiliary table that was renamed during the first upgrade attempt. (Bug #26995951)

- **InnoDB:** Workarounds introduced to address conflicting serialized dictionary information (SDI) inserts during concurrent `CREATE TABLE` operations were removed. (Bug #26995534)

  References: See also: Bug #26539665.

- **InnoDB:** A “no space left on device” error reported an invalid error message. (Bug #26960345)

- **InnoDB:** During a fast shutdown, `InnoDB` attempted to write dynamic metadata to the data dictionary after files were closed, resulting in an initialization failure due pending I/O on the data dictionary tablespace. (Bug #26950659)

- **InnoDB:** A stack overflow error was encountered on startup after upgrading to MySQL 8.0.4 due to repeated attempts to load an evicted `InnoDB` system table. (Bug #26945437, Bug #88042)

- **InnoDB:** Importing a compressed table raised an assertion. The operation used the clustered index of the table instead of the serialized dictionary information (SDI) index to transform SDI pages. (Bug #26938297)
- **InnoDB:** In debug builds, failed temporary table creation during a `REPLACE` operation raised an invalid assertion. (Bug #26919378, Bug #26958868)
- **InnoDB:** `DROP DATABASE` failed if database tables were created in a general tablespace. General tablespace flags were registered incorrectly causing the serialized dictionary information (SDI) operation to fail. (Bug #26834496)
- **InnoDB:** With binary logging enabled, an `ALTER TABLESPACE ... RENAME` operation failed with a “cannot find space” error. (Bug #26832347)
- **InnoDB:** An operation that failed to add an index raised an invalid adaptive hash index assertion. (Bug #26788968)
- **InnoDB:** A valid table row type value read from the data dictionary raised an invalid assertion. (Bug #26773152)
- **InnoDB:** Starting an upgrade with `innodb_force_recovery=5` initialized `InnoDB` background threads but did not exit the threads gracefully when an error was encountered. Upgrading with a nonzero `innodb_force_recovery` setting is no longer permitted. (Bug #26766632)
- **InnoDB:** A failed `CREATE TEMPORARY TABLE` statement left an entry in `INFORMATION_SCHEMA.INNODB_TEMP_TABLE_INFO`. The in-memory table object was not freed. (Bug #26765438)
- **InnoDB:** `InnoDB` looked up the name of a virtual column in the wrong `dict_table_t` array when attempting to locate a qualifying index for a foreign key. (Bug #26764604)
- **InnoDB:** Attachable read-write transactions that update the `table_stats` and `index_stats` data dictionary tables attempted to update the same row, causing a deadlock. (Bug #26762517)
- **InnoDB:** During recovery, the tablespace name in an in-memory tablepace object was defined using the file name character set instead of table name character set, resulting in a “missing tablespace” error. (Bug #26761960)
- **InnoDB:** Bootstrap code did not reserve the first 1024 table IDs for data dictionary tables. (Bug #26757227)
- **InnoDB:** Multiple updates from different clients on a partitioned table caused an unexpected lock wait timeout due to an incorrectly set lock type. (Bug #26731025, Bug #87619)
- **InnoDB:** An asynchronous rollback thread that attempted to acquire a metadata lock was interrupted, but the resulting error was not returned to the server. This issue was addressed by removing the metadata lock acquisition, which was not necessary for asynchronous rollback.

  Only in-memory tables were checked when opening tables for undo processing. The data dictionary is now checked as well, in case tables are not present in memory. (Bug #26678883)
- **InnoDB:** An assertion was raised when attempting to open a full-text auxiliary table with a name that was longer than expected. (Bug #26649020)
- **InnoDB:** Data dictionary table open functions did not properly handle table and schema name character set conversion, resulting in an error during recovery. (Bug #26640776)
- **InnoDB:** A transaction `end_stmt()` function was not called in some `ALTER TABLE ... PARTITION` scenarios, resulting in a timeout. (Bug #26629790, Bug #25886814)

- **InnoDB:** Acquiring a metadata lock on the serialized diction information (SDI) table during the commit phase of a DDL operation would fail due to a lock wait timeout or halting of the query. (Bug #26628126)
- **InnoDB:** Redo logs for dynamic metadata updates were not considered when checking redo log margin. Also, in read-only mode, the `innodb_dynamic_metadata` data dictionary table was opened unnecessarily for writing of metadata from the redo log. (Bug #26589535)
- **InnoDB:** An unexpected error occurred after a failed attempt to install the `memcached` plugin. (Bug #26588738)
- **InnoDB:** The state of a buffer pool page was altered by another thread while a buffer pool resize operation was in progress. (Bug #26588537)
- **InnoDB:** Debug functions that assert for conflicting locks did not account for transaction locks that are to be committed or rolled back. (Bug #26562371)
- **InnoDB:** Variance-Aware Transaction Scheduling (VATS) functionality that updates the age of waiting record locks failed to ignore table locks, causing an assertion failure. (Bug #26538702)
- **InnoDB:** A DDL operation that created or modified a table partition unintentionally altered the row format of other partitions, resulting in a row format mismatch. (Bug #26535746)
- **InnoDB:** An `ALTER TABLE` operation caused the server to halt. (Bug #26492721)
- **InnoDB:** The `innodb_table_stats` data dictionary table was not updated with new partition names when renaming a partitioned table. (Bug #26390658, Bug #86927)
- **InnoDB:** Due to a regression introduced in MySQL 8.0.0, the `innodb_change_buffering` configuration option could not be set dynamically. (Bug #26389442)
- **InnoDB:** The online log for a freed index was accessed while rolling back a concurrent `UPDATE` statement during an online DDL operation. (Bug #26334475)
- **InnoDB:** A `REPLACE` operation on a table with a secondary index on the prefix of a virtual column raised an assertion. (Bug #26330279)
- **InnoDB:** Setting `tmpdir` to the root of a drive caused “Invalid (old?) table or database name” error messages to be printed to the error log. (Bug #26299984, Bug #86737)
- **InnoDB:** A race condition occurred during an `INFORMATION_SCHEMA` query when attempting to check the transaction state without acquiring a transaction mutex. (Bug #26299705)
- **InnoDB:** A `FLUSH TABLES` operation failed to drop an aborted index. While removing the table from the cache, the clustered index was dropped prior to checking for the aborted index. (Bug #26256456, Bug #86607)
- **InnoDB:** For `InnoDB` tables, `CREATE TABLE ... LIKE` did not respect the `innodb_file_per_table` system variable setting, and `SHOW CREATE TABLE` displayed a `TABLESPACE` clause even though the user specified no explicit tablespace during table creation. (Bug #26199233, Bug #86589)
- **InnoDB:** An iterative approach to processing foreign cascade operations resulted in excessive memory use. (Bug #26191879, Bug #86573)

  References: This issue is a regression of: Bug #16244691.
- **InnoDB:** The lock acquisition sequence for a buffer pool eviction operation that evicts compressed pages was incorrect. (Bug #25972975)
- **InnoDB:** Metadata locks were released while data dictionary objects were still in use. (Bug #25928984)

- **InnoDB:** `innochecksum` returned a Valgrind error when run on `InnoDB` files with a 1K compressed page size. (Bug #25922124, Bug #85993)
- **InnoDB:** A kill thread failed to close the socket of another thread that was executing a `TRUNCATE TABLE` operation, causing an assertion. (Bug #25887335, Bug #85925)
- **InnoDB:** An `INSERT` operation on a table with a spatial index raised an assertion due to a failure that occurred during a lock conflict check. (Bug #25729649)
- **InnoDB:** A debug sync point intended for user tables was activated for data dictionary tables. (Bug #25508568)
- **InnoDB:** A server-side check was added to prevent a foreign key constraint from being placed on the base column of a generated stored column. (Bug #25339192)
- **InnoDB:** Warnings that should only appear in debug builds of MySQL were printed to the error log when the length of the history list exceeded 2000000. (Bug #24296076, Bug #82213)
- **InnoDB:** Attempting to reduce the buffer pool size to less than the buffer pool chunk size did not report a warning. (Bug #23590280)
- **InnoDB:** A “wrong key column” error was added to address an unsupported index creation scenario. (Bug #22486025)
- **InnoDB:** Full-text search on indexed columns that use a binary collation did not return case-sensitive matches. (Bug #21625016, Bug #78048)
- **Packaging:** When trying to install MySQL Server on Fedora 27 using the MySQL Yum repository, installation failed due to a conflict with the native `mariadb-connector-c-devel` package. With this fix, the appropriate “obsoletes” have been added for that and other native packages. (Bug #26963839)
- **Partitioning:** When creating a partitioned table using an implicit tablespace, the effect is to place each partition in its own tablespace, with no designated tablespace for the table as a whole. Since serialized dictionary information (SDI) was stored in all tablespaces used by a given table, the cost of storing it in a table with many tablespaces became prohibitive. This problem is solved by including only the tablespace for the first partition in the set of tablespaces used to store the SDI. (Bug #26762973)

  References: See also: Bug #26765252.
- **Partitioning:** An assertion could be raised on `CREATE VIEW` on partitioned tables when the server tried to prune partitions of the underlying tables. (Bug #26659699)
- **Partitioning:** When renaming a partitioned table, the table statistics were not updated with the new partition names. (Bug #86074, Bug #25953183)
- **Replication; JSON:** For row-based replication, partial updates to JSON documents were not applied if the server variable `binlog_row_value_options=PARTIAL_JSON` (introduced in MySQL 8.0.3) was not specified on the replication slave, as well as on the master. Now, a replication slave applies partial updates to JSON documents whenever these are received from the master, whether or not the slave has `binlog_row_value_options=PARTIAL_JSON` in its own settings. (Bug #26762675)
- **Replication:** The function `set_unknow_error()` in the `Binlog_sender` class has been renamed to `set_unknown_error()`. Thanks to Simon Mudd for the fix (and also for the typo fix in Bug 88149). (Bug #27149075, Bug #88559)

  References: See also: Bug #26996065, Bug #88149.
- **Replication:** When you invoke `mysqld` with the `--initialize` or `--initialize-insecure` option to initialize the data directory, a warning message is no longer issued regarding the availability of the

`mysql.gtid_executed` table, which should not be available at that stage. Also, the message formerly issued as a warning regarding the generation of a new UUID is now issued as a note, because the generation of a new UUID is normal in that situation. (Bug #27115183)

- **Replication:** In MySQL 8.0.3, the default base name for the binary log files and index file was *`host_name`*`-bin`, using the name of the host machine. This default name was used if the `--log-bin` option was not supplied, and also if the `--log-bin` option was supplied with no string or with an empty string. From MySQL 8.0.4, if you do not supply the `--log-bin` option, MySQL now uses `binlog` as the default base name for the binary log files and index file. In releases before MySQL 8.0.3, there was no binary log with that configuration, so there is no incompatibility with existing binary logs at upgrade. However, for compatibility with existing binary logs from releases before MySQL 8.0.3, if you supply the `--log-bin` option with no string or with an empty string, the base name defaults to *`host_name`*`-bin`, using the name of the host machine.

  The warning messages that were previously issued at startup if you did not specify a binary log file name using the `--log-bin` option (`ER_LOG_BIN_BETTER_WITH_NAME`) and if you did not specify a server ID by setting the `server_id` system variable (`ER_WARN_NO_SERVERID_SPECIFIED`) are now issued as informational messages. A warning message is still issued if replication is attempted with a nonunique server ID. (Bug #27082922)

- **Replication:** In the `Gtid_log_event` that precedes every GTID transaction in the binary log file, the `transaction_length` field used 8 bytes for transactions with 16777216 bytes or more, when it should have used the maximum permitted 9 bytes. (Bug #26993433)

- **Replication:** The fix for Bug #22671846 was missing from MySQL version 8.0.3. (Bug #26985976)

- **Replication:** The fix for Bug #26117735 (MySQL Bug #86288) could cause a debug assertion when running `mysqlbinlog` with the `--read-from-remote-server` option and the `--rewrite-db` option, depending on the database names specified in the rewrite rule. The issue has now been corrected. (Bug #26878022)

- **Replication:** With MySQL compiled using yaSSL, and semisynchronous replication in use, a deadlock could be caused by incorrect handling of acknowledgement packets. Multiple acknowledgement packets can be read together by yaSSL, but the receiver thread for semisynchronous replication only handled the first acknowledgement packet seen after polling. Now, the receiver thread handles all acknowledgement packets that are present in the buffer. (Bug #26865538)

- **Replication:** With semisynchronous replication in use, if `RESET MASTER` was issued while an active transaction was waiting for an acknowledgement from the slave, the count of waiting sessions in the `Rpl_semi_sync_master_wait_sessions` server status variable was incorrect after the wait was completed. (Bug #26748533)

- **Replication:** The `--log-slave-updates` and `--slave-preserve-commit-order` options require binary logging. If you specify these options and also disable binary logging using the `--skip-log-bin` or `--disable-log-bin` option, a warning or error message is issued. The `--skip-log-bin` and `--disable-log-bin` options now disable the `--log-slave-updates` and `--slave-preserve-commit-order` options by default, so when those options are not specified, the warning or error message is not issued. (Bug #26666259)

- **Replication:** `XA ROLLBACK` statements that failed because an incorrect transaction ID was given, could be recorded in the binary log with the correct transaction ID, and could therefore be actioned by replication slaves. A check is now made for the error situation before binary logging takes place, and failed `XA ROLLBACK` statements are not logged. (Bug #26618925, Bug #87393)

- **Replication:** For the NDB storage engine, when the slave used hashing for searches of rows (which is included by default in the setting for the `slave_rows_search_algorithms` system variable from

MySQL 8.0.2), the table used to store the row hashes was not cleaned up correctly after records were removed on the slave. The issue was caused by a variant error value returned by the NDB storage engine, which has now been corrected to the expected value. (Bug #26434966)

- **Replication:** MySQL internal administration commands that update replication-specific repository tables, for example during a replication synchronization check using the `mysqlrplsync` utility, can now bypass read locks. This enables such commands to execute regardless of the settings for the `read_only` and `super_read_only` system variables and the `autocommit` mode. (Bug #26414532, Bug #86224)

- **Replication:** The binary log function `MYSQL_BIN_LOG::new_file_impl` returned the error "Can't open file" (`ER_CANT_OPEN_FILE`) when it should have returned "Error writing file" (`ER_ERROR_ON_WRITE`). (Bug #26370868, Bug #86870)

- **Replication:** When write sets are used for parallelization by a replication slave (as specified by the `binlog_transaction_dependency_tracking` system variable), the case and accent sensitivity of the database are now taken into account when generating the write set information. Write set information is generated when the `transaction_write_set_extraction` system variable is enabled. Previously, duplicate keys could be incorrectly identified as different, causing transactions to have incorrect dependencies and so potentially be executed in the wrong order. (Bug #26277771, Bug #86078)

- **Replication:** The receiver thread for semisynchronous replication was not able to receive acknowledgements from slaves that used compression of the master/slave protocol (`slave_compressed_protocol=ON`). The receiver thread now handles compressed acknowledgements correctly. (Bug #26027024, Bug #86230)

- **Replication:** The `mysql_reset_connection()` function now clears the write set session history. (Bug #25950554, Bug #86063)

- **Replication:** On replication slaves, in the `XA_STATE` field in the Performance Schema table `events_transactions_current`, the state of XA transactions was incorrectly reported as `COMMITTED` instead of `PREPARED` after the `XA PREPARE` statement was applied on the slave. (Bug #25940184)

- **Replication:** In a multi-source replication topology, a memory leak could occur on the slave when `binlog_rows_query_log_events` was enabled on the master, and a statement already applied from another channel was skipped on the slave. In this situation, the instance of the `Rows_query` log event stored on the slave was not being deleted. The log event instance is now cleaned up and the memory is freed. Thanks to Vlad Lesin for his contribution to the patch. (Bug #25695434, Bug #85371, Bug #85034)

- **Replication:** Queries to the Performance Schema `replication_applier_global_filters` and `replication_applier_filters` tables, which show the global and channel-specific replication filters configured on a replication slave, have been optimized so that a view is generated only when the filters are changed. Previously, a view was generated for every row that was created. (Bug #25694140)

- **Replication:** A memory leak was fixed in GTID-based replication. Memory was not being freed after the repository tables were updated for skipped or ignored events. (Bug #25656123, Bug #85251)

- **Replication:** When a worker thread on a multithreaded slave failed to apply a transaction on which a later transaction depended, the coordinator thread could begin scheduling the dependent transaction before being notified of the issue. If a `STOP SLAVE` request was made during this situation, it caused an assertion to be raised in debug builds. (Bug #25585436)

- **Replication:** With statement-based replication in use, if an `UPDATE` or `DELETE` statement was used inside an XA transaction ending with `XA COMMIT ONE PHASE`, and the statement did not affect any rows, a replication error occurred. An `XA END` statement was not written to the binary log, so slave servers identified the XA transaction as still being active at the time of the commit request. The required

`XA END` statement is now written even if the transaction affected no rows. (Bug #24812958, Bug #83295)

- **Replication:** Replication clients no longer enable `LOCAL` capability for `LOAD DATA` statements, because they do not use `LOAD DATA LOCAL` statements. (Bug #24763131)

- **Replication:** The behavior of mixed-format replication (`binlog_format=MIXED`) has changed with regards to temporary tables. Previously, when mixed-format binary logging was in use, if a statement was logged by row and the session that executed the statement had any temporary tables, all subsequent statements were treated as unsafe and logged in row-based format until all temporary tables in use by that session were dropped. Also, on a replication slave with `log_slave_updates` enabled, row-based logging was incorrectly continued across all subsequent sessions for the duration of the connection, as reported in the bug.

  Now, when mixed binary logging format is in use, statements that exclusively use temporary tables are not logged. Statements that involve a mix of temporary and nontemporary tables are logged on the master only for the operations on nontemporary tables, and the operations on temporary tables are not logged. The exception is if the creation of a temporary table was recorded in the binary log using statement-based format. In this case, a `DROP TEMPORARY TABLE IF EXISTS` statement is logged on the master when the temporary table is dropped.

  With this change in behavior, the remaining statements in the session that do not involve temporary tables no longer need to be treated as unsafe. The safe statements are now logged in statement-based format, and the unsafe statements are logged in row-based format, according to the normal behavior for mixed format replication, regardless of the presence of temporary tables in the session. Also, the issue reported in the bug has been fixed so that subsequent sessions using the connection now use the appropriate logging format for the session, regardless of the format used by earlier sessions.

  When `binlog_format` is `ROW` or `STATEMENT`, the behavior remains as before. For row-based binary logging format, operations on temporary tables are not logged, with the exception of the `DROP TEMPORARY TABLE IF EXISTS` statement as for mixed format. For statement-based binary logging format, operations on temporary tables are logged on the master and replicated on the slave, provided that the statements involving temporary tables can be logged safely using statement-based format. `binlog_format=STATEMENT` is now the only logging mode in which temporary tables are replicated on the slave.

  You cannot now change the `binlog_format` setting from `ROW` or `MIXED` to `STATEMENT` at runtime, because any `CREATE TEMPORARY TABLE` statements will have been omitted from the binary log in the previous mode. You can still switch from `STATEMENT` to `ROW` or `MIXED` format, even when temporary tables have been created.

  Thanks to George Lorch and Laurynas Biveinis from Percona for the patch. (Bug #18843730, Bug #72475)

- **Replication:** During distributed recovery as part of joining the group, when the applier was signaling that it had applied all transactions, it was also blindly searching for partial transactions. This was to avoid future applier errors, which would happen if the applier stopped at this point. However, this search and remove only made sense for applier stop cases. Upon execution completeness it should not be done, otherwise it can corrupt or purge the applier relay log, which can led to data loss. To solve this issue, when the applier is waiting for execution completeness, it no longer searches for and removes partial transactions. (Bug #88304, Bug #27049034)

- **Replication:** In a group with heavy load, joining members could need to retrieve a large amount of data to gain synchrony with the group. If the amount of data retrieved exceeded the `View_change` packet size of 4Mb the members would fail to join the group and enter `Error` state. Now, the packet size is taken from `slave_max_allowed_packet`, which defaults to 1GB. Depending on

the load your group processes, you might want to increase the packet size further by configuring `slave_max_allowed_packet`. (Bug #87701, Bug #26770576)

- **Replication:** In a group where a joining member consistently received transactions, the joining member could sometimes not enter the online state. This was due to the way the incoming queue of messages was tested. (Bug #87631, Bug #26731317)
- **Group Replication; Microsoft Windows:** On Windows, errors generated by Group Replication now contain a detailed error message rather than just the error number. (Bug #24918678)
- **Group Replication:** The thread where the Group Replication plugin was started was not being correctly killed, making it impossible to stop or start the plugin after killing this thread. (Bug #26435775)
- **Group Replication:** When `transaction_write_set_extraction` was enabled, there was a risk of unnecessary serialization while foreign keys were gathered if concurrent DDL took place. Group Replication now takes advantage of the new Data Dictionary to interact with table definitions and foreign keys, which has solved this potential serialization. (Bug #26187850)
- **Group Replication:** When `group_replication_enforce_update_everywhere_checks` is `ON`, the Group Replication plugin checks whether there are any foreign key cascades and disallows updates to tables affected by them.

  `SET NULL` operations were not checked as part of this process, which could lead to data inconsistency. Now, when this value is `ON`, operations on child tables are blocked if the table has a `SET NULL` option configured. (Bug #25404162)
- **Group Replication:** All servers that belong to a group must have unique UUIDs set by `server_uuid`, but this was not enforced by Group Replication, so that it was possible to add members with duplicate UUIDs. (Bug #88452, Bug #27105803)
- **Group Replication:** When starting or stopping, the Group Replication plugin executes internal operations on the server such as enabling or disabling read only mode using an internal session. When this internal session was opened, if the total number of sessions exceeded the number of permitted open sessions set by `max_connections`, the operation failed as expected but a thread was left behind, which later caused issues. (Bug #88182, Bug #27008102, Bug #27016552)
- **Group Replication:** If Group Replication was configured to start on server boot when the server was being initialized using `--initialize` or `--initialize-insecure`, because the replication applier infrastructure was not initialized, this resulted in an assertion. Now, Group Replication is not started when the server is being initialized. (Bug #87759, Bug #26802395)
- **Group Replication:** Changes to Group Replication variables while starting or stopping the plugin were not being correctly validated. Now, the variables can only be changed if the plugin is not changing state. (Bug #86874, Bug #26372117)
- **Group Replication:** Regardless of the number of virtual IPs configured on a machine, Group Replication could access only the first 12 addresses. (Bug #86772, Bug #26324852)
- **Group Replication:** The delayed initialization mechanism used for server starts has been improved. Now, it blocks connections only until the server is in read mode. (Bug #86271, Bug #26037344)

  References: See also: Bug #84731, Bug #25475132.
- **Group Replication:** When a primary member, such as the primary in a single-primary group or in a multi-primary group which also had asynchronous replication channels feeding data into it, was stopped, the asynchronous channels would continue applying changes. Although `super_read_only` was set when `STOP GROUP_REPLICATION` was issued, this did not stop any running asynchronous replication

channels already running. This meant that changes could be made locally on the member, and that the asynchronous replication channels had to be stopped manually. Now when Group Replication stops, either due to an error or when `STOP GROUP_REPLICATION` is issued, all asynchronous replication channels are stopped. (Bug #86222, Bug #26024253)

- **Group Replication:** Group Replication logging has been improved, and now includes includes about when a member joins or leaves, when the view changes, and so on. (Bug #84798, Bug #25495393)

  References: See also: Bug #26422857.

- **Linux:** On Alpine Linux, `mysql` would lose its connection to the server if its standard output was not writable. Also, for `mysql` and `mysqldump`, order of result flushing for `stdout` and `stderr` is now deterministic. (Bug #27169809)

  References: See also: Bug #17583.

- **Microsoft Windows:** On Windows, with the `myisam_use_mmap` and `flush` system variables enabled, `MyISAM` did not always flush table files properly. (Bug #26880757)

- **JSON:** JSON expressions used as arguments with the `LAG()` function were not always evaluated correctly. (Bug #26740557)

- **JSON:** Repeated execution of a prepared statement that employed `JSON_ARRAY()` was not handled correctly. (Bug #26704312)

  References: This issue is a regression of: Bug #25867454.

- **JSON:** When executing the `JSON_INSERT()` function, the check that is performed to determine whether or not a given insert is being made into the root element tested whether the length of the path was 1—that is, whether the path consisted of a single leg determining which position the inserted element has inside the root element. A problem occurred when there were auto-wrapping path legs at the beginning of the path, in which case a path whose length is greater than 1 might also refer to an element in the root, so that checking the path length did not reliably inform us whether the target element of the insert was the root or some other element.

  To fix this, the check of the path length for detection of the root element has been replaced with a check as to whether the matched element has a parent; if it has none, it must be the root element. (Bug #26649978)

  References: This issue is a regression of: Bug #86213, Bug #26022576.

- **JSON:** When serializing a JSON value to its binary representation, it is necessary to make sure that the destination buffer has sufficient space to hold an integer or double value of the required size. Allocation of this buffer previously reserved only the minimum amount of memory needed, which made it very likely that a reallocation would be needed shortly thereafter. This could adversely affect performance, especially when serializing arrays with many numeric values. The serialization is now performed in a manner such that the allocation increases the size of the destination buffer size exponentially, which reduces the amount of the time spent performing this task when processing large arrays. (Bug #88656, Bug #27171283)

- **JSON:** When a JSON document was converted to string representation, floating-point values that had no fractional part could be represented such that they became indistinguishable from integers. When the string representation of such a JSON document was passed through the JSON parser again, the information that the numeric value was originally specified as a floating-point value was lost.

  To rectify this problem, a fractional part is now added to the string representation of a floating-point value in a JSON document if the value has no fractional part and is not represented using scientific format.

This makes the string representation of a floating-point value distinguishable from that of an integer, so that it continues to be treated as a floating-point number even if the string is parsed again.

This fix also makes `ST_GeomFromGeoJSON()` use the same JSON parser as the other JSON functions rather than its own custom parser as had been the case since MySQL 5.7.8; this special handling was due to the fact that `ST_AsGeoJSON()` dropped the fractional part of negative zero (-0 instead of -0.0), causing the JSON parser to interpret -0 as integer 0, thus losing the distinction between positive and negative zero. Since `ST_AsGeoJSON()` now uses the standard JSON parser, it represents negative zero as correctly as -0.0, obviating any need for `ST_GeomFromGeoJSON()` to preserve negative zero explicitly on its own when parsing the output from `ST_AsGeoJSON()`. (Bug #88230, Bug #27028889)

References: See also: Bug #19504183.

- **JSON:** When inserting `JSON` values created from the result of a `GROUP BY` query, the inserted values could sometimes include the concatenation of all the values previously inserted into that column. (Bug #87854, Bug #26867509)
- **JSON:** When called using strings extracted from JSON documents as arguments, the `LEAD()` and `LAG()` functions returned the same value for every row. (Bug #87839, Bug #26848089)
- **JSON:** The microseconds part of the last-updated field in each histogram in the `INFORMATION_SCHEMA.COLUMN_STATISTICS` table (showing when the histogram was last updated) was dropped when serializing the histogram into JSON and so was not stored. (Bug #87833, Bug #26846289)
- **JSON:** When a `JSON_SET()` statement updated a `JSON` value to the same value using a partial update (in other words, when the partial update was essentially a `NOOP`), it was possible that logical diffs for this operation were produced, even though no binary diffs were produced. Now in such cases, neither logical update nor binary diffs are generated. (Bug #87113, Bug #26483625)
- **JSON:** Following the implementation of JSON partial updates, the same JSON document could have different binary representations on the master and the slave. This could lead row-based replication—which uses binary equality to find the matching row on the slave—to fail if this occurred. Now the string representation of the JSON document is used for the comparison instead.

  Also as a result of this fix, updates can be skipped in more cases than previously; this is true where the binary representation has changed, but not the contents of the document. (Bug #86532, Bug #26177130)
- Ubuntu 14.04 and Debian 8 are no longer supported. (Bug #27422291)
- For builds on 32-bit platforms with Undefined Behavior Sanitizer enabled, a stack-overrun check could cause a server exit. (Bug #27224961)
- The server could hang during spatial reference system (SRS) creation while another session was using that SRS. (Bug #27220467)
- A lock for the privilege cache was acquired unnecessarily during privilege-checking operations not involving table permissions. (Bug #27197483)
- Persisting the read-only `gtid_owned` or `gtid_executed` system variable caused an assertion failure at server startup. These variables can no longer be persisted. (Bug #27193853)
- Improper handling of plugin loading and unloading could cause a server exit. (Bug #27151550, Bug #88589, Bug #27116827, Bug #88483)
- Error propagation from some windowing functions was not always performed correctly. (Bug #27135084, Bug #27136492)

- Negation of some very large values was not handled correctly by an internal function. (Bug #27134168)
- Instituted stricter checks when performing addition involving date intervals. (Bug #27134148)
- Recently introduced SRID and COLUMN_STATISTICS metadata locks were not instrumented by the Performance Schema. (Bug #27124506)
- The name of a derived table was not saved before the table was materialized and assigned the name of the temporary table. Later, when trying to reset the table name, this caused the server to fail due to the missing reference to the original value of the name. (Bug #27121663)
- Performance Schema queries that used indexes on `OBJECT_TYPE` columns could return incorrect results. (Bug #27121500)
- Compiling with `-DWITH_ASAN=1` and `-DWITH_ASAN_SCOPE=1` detected a stack-use-after-scope memory error. (Bug #27108794, Bug #88460)
- `FILE` privilege checking for prepared `SELECT ... INTO OUTFILE` statements was incorrect. (Bug #27094955)
- Some messages with information priority were written to the error log when `log_error_verbosity` was less than 3. (Bug #27082862)
- The `ha_create_table_from_engine` function failed to pass a table object to the `ha_create` routine. (Bug #27066335)
- Partition by and order by elements of unused window definitions were not included when estimating memory requirements. This is resolved by assigning a parsing context (`CTX_WINDOW`), but only when the current context is `CTX_NONE`. As part of this fix, unused window definitions are now removed after being checked for syntax and semantic errors. (Bug #27062031)
- `GROUP BY` with a `ROLLUP` that generated `NULL` was not handled correctly. (Bug #27060420)
- An Event Scheduler event for which global autocommit was disabled at event expiration time caused an assertion to be raised. (Bug #27041552, Bug #88255)
- Length calculations for string-valued user-defined variables could be incorrect if the collation was changed. (Bug #27041543, Bug #88256)
- `CREATE TABLE ... SELECT` statements that attempted to create a non-`InnoDB` table raised an assertion if a pre-existing view referenced the table to be created. (Bug #27041536, Bug #88258)
- When used as an argument to the `IF()` function, the value of a `TIMESTAMP` column could be handled differently for different collations. (Bug #27041526, Bug #88259)
- With statement-based binary logging, using `CREATE TABLE ... SELECT` to create a `BLACKHOLE` table caused an assertion to be raised. (Bug #27041516, Bug #88260)
- For debug builds, a `TIMESTAMP`-related assertion could be raised with `explicit_defaults_for_timestamp` enabled. (Bug #27041502, Bug #88261)
- Under `LOCK TABLES`, an attempt to execute a DML statement on a table with foreign keys led to assertion failure if the statement was incompatible with the mode under which the tables in the foreign key relationship were locked. (Bug #27041477, Bug #88264)
- With a `LOCK TABLES` statement active, queries that select from the `INFORMATION_SCHEMA.FILES` table could raise an assertion trying to obtain a metadata lock. (Bug #27041452, Bug #88266)

- Component installation did not properly perform auto-increment handling, which could result in a server exit. (Bug #27041374, Bug #88276)
- With `big_tables=1` and `character_set_connection=ucs2`, `SHOW TABLE STATUS` could raise an assertion. (Bug #27041323, Bug #88279)
- With a backup lock active, removal of binary log files and relay log files incorrectly was permitted. (Bug #27030339, Bug #88238)
- When evaluating an end-range condition in a scan of a non-covering secondary index, and the end-range condition referenced an indexed virtual column, `InnoDB` sometimes read the wrong column from the index, which could lead to assertion failures or wrong results. (Bug #27010089)
- In `event` items in filter rules, the `audit_log` plugin did not properly process values specified as a `JSON` array. (Bug #27010045)
- An integer overflow was sometimes possible when adding date values, with the potential to return invalid results. (Bug #27004806)
- `TRUNCATE TABLE` on an `InnoDB` table with an active `LOCK TABLES` raised an assertion if more than one table instance was locked. (Bug #26999526)
- Some windowing functions including `NTH_VALUE()` did not return `NULL` in all cases in which they should have. (Bug #26975882)
- If one of the temporal arguments to `LEAST()` or `GREATEST()` was of type `DATETIME`, but the aggregated data type was something other than `VARCHAR` or a temporal type, the `LEAST()` or `GREATEST()` item had temporal properties set for it in spite of the fact that it was not of a temporal data type. The temporal properties for the `LEAST()` or `GREATEST()` item, including its fractional precision, were therefore not calculated correctly, leading to failure at a later point in statement execution.

  To fix this issue, the data type of `LEAST()` or `GREATEST()` is now temporarily set to a temporal type if one or more, but not all, of its arguments are of temporal types regardless of their aggregated data type. (Bug #26975864)
- `IFNULL()` did not always check for errors correctly when processing multiple arguments. (Bug #26975848, Bug #27062796, Bug #27062694)
- With compression enabled for the client/server protocol, logical packets that are multiples of size `0xFFFFFF` could cause the connection to be dropped. Thanks to Facebook Inc. for the patch. (Bug #26974113, Bug #88092)
- Installing and uninstalling a plugin many times from multiple sessions could cause the server to become unresponsive. (Bug #26946491)
- A virtual column definition that included a function removed in MySQL 8.0 caused startup to fail with a “No database selected” error when starting the MySQL 8.0 server on a MySQL 5.7 data directory. (Bug #26945125, Bug #88040)
- These errors occurred after an in-place upgrade from MySQL 5.7 to 8.0:
  - Starting the server with an `--explicit-defaults-for-timestamp=0` setting returned an `Invalid default value for 'cached_time'` error.
  - Starting the server with `--initialize` and `--explicit-defaults-for-timestamp=0` configuration settings returned an `Invalid default value for 'SET_TIME' error`.
  - Duplicate `SET` data type values caused a `Duplicated value in SET` error, regardless of the `sql_mode` configuration setting.

(Bug #26944731, Bug #88039, Bug #26948678, Bug #88032)

- Queries with a common table expression and a derived table or view that contained a window function produced incorrect results. (Bug #26907753, Bug #87939)
- The deprecation warnings were clarified regarding use of the `--symbolic-links` and `--skip-symbolic-links` server options (and their equivalents). The server no longer warns about a missing data directory when invoked with the `--help` option. (Bug #26898576, Bug #87913)
- `VALUES()` was not handled correctly in some cases. (Bug #26881946)

  References: See also: Bug #19601973, Bug #17458914.
- The tablespace discovery mechanism in MySQL Cluster was disabled, which prevented serialized dictionary information from being imported into the data dictionary in cases where the table exists in the storage engine dictionary but not in the MySQL data dictionary. (Bug #26867488)
- For debug builds, validation checks on relevant generated columns could be missed for `UPDATE` statements, leading to a server exit. (Bug #26838771)
- For window functions, an error could be produced that a window was not defined, when it was defined. (Bug #26813454, Bug #87780)
- When a materialized derived table was determined to be superfluous, the routine that deleted the corresponding object left it in an inconsistent state. Now in such cases, the derived table's `TABLE_LIST` object left in a consistent state after its materialized object is deleted, by setting its table pointer to `NULL`. (Bug #26798989)
- Incorrect results were obtained for a query with `MAX()` and a `HAVING` clause used inside a view. (Bug #26781725)
- Metadata locking for definition changes to tables underlying a view could be inconsistent with metadata locking for other statements on the tables. (Bug #26770836)
- In MySQL 8.0, view column names are restricted to 64 characters. A MySQL 5.7 view with longer column names was marked as invalid during an upgrade to 8.0 was marked invalid. Now such views produce an error during the upgrade and must be altered to have legal column names before the upgrade will succeed. (Bug #26743291, Bug #87650)
- Concurrent calls to `GET_LOCK()` could cause deadlock, even with a wait time of 0. (Bug #26739438, Bug #87620)
- `FROM_UNIXTIME()` did not always work correctly with `LAG()`. (Bug #26739028)
- Following an `INSERT` statement with `BLOB` values in the `ON DUPLICATE KEY UPDATE` clause that failed with a constraint violation, a similar statement with no reason to return an error could cause a server exit. (Bug #26734162)
- The Performance Schema now stores rewritten rather than raw SQL statement text when available. (Bug #26732229)
- Re-executing a prepared statement that used window functions could cause a server exit. (Bug #26730020)
- An in-place upgrade to MySQL 8.0 caused a server exit if tables contained columns with a pre-5.0 `DECIMAL` data type. This data type is not supported, so upgrades now detect such columns and warn

that in-place upgrade cannot be done. Affected tables must be upgraded by dumping and restoring them. (Bug #26727481)

- A comparison operator used to order keys in a data dictionary cache hash map incorrectly determined that two storage-engine private IDs used by different storage engines were equal. (Bug #26723442)
- Values in the `XID_GTRID` column of the Performance Schema `events_transactions_current` table were displayed incorrectly for XA transactions. (Bug #26710081, Bug #87559)
- Incorrect results or a server exit could result when `SHA2()` was passed a user-defined variable in some character sets. (Bug #26704451)
- Incorrect `NULL` handling by `LAG()` and `LEAD()` could cause a server exit. (Bug #26703246, Bug #26703156)
- If an error occurred while setting up the temporary table for duplicate weedout in a semijoin (for example, because the disk was full), the server did not terminate gracefully with an appropriate error message.

  Now in the event that the temporary table is not successfully created, the query is aborted. (Bug #26679983)
- The server did not always clean up correctly after executing an `IN` subquery that used a hash semijoin. (Bug #26679495)
- Building with the `-DWITHOUT_SERVER=ON` `CMake` option failed due to attempting to link the `authentication_ldap_sasl_client` client-side plugin against the embedded server library. (Bug #26665217)
- During data directory creation or upgrade from MySQL 5.7 to 8.0, server startup would fail due to a Performance Schema initialization failure if the server was started in `read_only` mode. Additionally, Information Schema metadata was not updated at startup, and Performance Schema and Information Schema version information was stored without verifying that schema tables were created.

  The versioning scheme used for the data dictionary and Information Schema system views is now consistent with the Performance Schema versioning scheme. (Bug #26636238, Bug #87436)
- The `CREATE_OPTIONS` column in the `INFORMATION_SCHEMA.TABLES` table did not show correct information. (Bug #26634507)
- Incorrect results could be returned for queries that used an outer join and a derived table referenced a `const` value from an inner table of the outer join. (Bug #26627181)
- `AFTER UPDATE` triggers were not invoked for `INSERT ... ON DUPLICATE KEY UPDATE` when the value to be updated and the new value were the same. (Bug #26626277, Bug #87371)
- Assignment of anonymous roles to the `mandatory_roles` system variable was incorrectly permitted. Additionally, assigning a value to `mandatory_roles` now requires the `ROLE_ADMIN` privilege, in addition to the `SYSTEM_VARIABLES_ADMIN` or `SUPER` privilege normally required to set a global system variable. (Bug #26576989)
- The server fell back to using the built-in error messages if the `lc_messages_dir` value was invalid at server startup, but not if `lc_messages` or `lc_time_names` were invalid. Now the server uses the built-in messages if any of those variables are invalid at startup. (Bug #26576922)
- `SET DEFAULT ROLE ALL` did not include roles named in the `mandatory_roles` system variable. (Bug #26571995)
- On Windows, `CMake` did not automatically add x64 toolchain support for some Visual Studio versions. (Bug #26566360)

- Problems could occur when a derived table with an `ORDER BY` clause was merged into an outer query, and when the columns from the `ORDER BY` were not also referenced in the outer query. (Bug #26542829)
- Parallel inserts of schema SDI into the SDI B-tree could raise an assertion when creating tables in the same schema in parallel. (Bug #26539665, Bug #87225)
- Changing the `UMASK` and `UMASK_DIR` environment variables from their default values had no effect on database directory and table file access. (Bug #26529942)
- For debug builds, incorrect nullability assessment of derived table column references could cause `CONCAT()` to raise an assertion. (Bug #26524721)
- A server exit could result from simultaneous attempts by multiple threads to register and deregister metadata Performance Schema objects, or to acquire and release metadata locks. (Bug #26502135)
- MSI packages for Windows failed to detect when Microsoft Visual C++ 2010 Redistributable Package was installed. (Bug #26501092, Bug #87139)
- Queries that used window functions for tables that contained a `BLOB` column could cause a server exit. (Bug #26496880)
- Persisted variables belonging to plugins were not always handled properly at server startup. (Bug #26495619)
- A server exit could occur for queries that used `DISINCT` and `ORDER BY ... LIMIT` and were executed using range access and a temporary table for the `ORDER BY`. (Bug #26483909)
- LDAP authentication plugins could fail if their associated system variables were set to invalid values. (Bug #26474964)
- The Linux RPM spec file for RHEL6 and higher is updated with comments that recommend installing the `redhat-rpm-config` package to add convenience macros that make rebuilding the RPM package easier. Thanks to Simon Mudd for the patch. (Bug #26474153, Bug #87098)
- If the error log was misconfigured and the server could not start, no output describing the problem was produced. (Bug #26447825, Bug #87087)
- When a materialized semijoin operation was evaluated more than once, and one of the tables in the materialization was a `const` table (that is, with join type `JT_CONST`), invalid data was accessed during the second materialization when referencing the `const` table. (Bug #26436185)
- Password-expiration options did not work correctly for authentication plugins that use external authentication methods. (Bug #26435766)
- Adding an `ORDER BY` to a query that included an outer join and a subquery caused a constant value defined for a column in the subquery to be incorrectly promoted to a constant value in the case when the subquery returns 0 rows. (Bug #26432173)
- For the `autocommit` system variable, the Performance Schema `variables_info` table always reported the `VARIABLE_SOURCE` column as `COMPILED`. (Bug #26428017)
- For debug builds, `INSERT IGNORE` statements that tried to insert `NULL` into a `GEOMETRY NOT NULL` column raised an assertion because there is no valid value to convert the `NULL` to. This is now handled as a nonignorable `ER_BAD_NULL_ERROR_NOT_IGNORED` error. (Bug #26412713)
- `SET PERSIST_ONLY` changed the `VARIABLE_SOURCE` column of the Performance Schema `variables_info` table when it should not have. (Bug #26395134)
- The server failed to check the maximum path length for partition names. (Bug #26390632)

- Problems occurred when a window with buffering followed an equi-join on a unique index, due to the fact that the window modified the input record with the assumption that, the next time control passes back to the join, a new record was read to the input record. This problem is addressed by reinstating the input record in such cases.

  **Note**

  This fix was reverted in MySQL 8.0.27.

  (Bug #26389508)

  References: See also: Bug #32820802.

- Identifiers containing a mixt of backslashes and backticks could be parsed incorrectly. (Bug #26372491)

- `audit_log` plugin `THD` objects could be created with incorrect thread ID information, leading to assertion failure. (Bug #26362452)

- The `HISTOGRAM` column in the `column_statistics` data dictionary table used a key named `charset-id` to indicate collation numbers. This key has been renamed to `collation-id`. (Bug #26330090, Bug #86794)

- Starting the server with the `skip_name_resolve` system variable enabled could cause `localhost` in account entries to match non-`localhost` hosts. (Bug #26328274, Bug #26202411, Bug #86546)

- if a configured error log service existed but could not be initialized, log information was lost. Now if a log service is configured at startup but cannot be initialized, diagnostics about the problem are sent to the default log service (or if that fails, directly to the error stream), and the server exits. If a log service is configured at runtime but cannot be initialized, diagnostics are sent to the client. (Bug #26286871, Bug #86728)

- Installing and uninstalling Performance Schema example plugins concurrently with deletes from tables associated with those plugins could cause a server exit. (Bug #26281359)

- When `HASH_SCAN` was specified as one of the values for the `slave_rows_search_algorithms` system variable, which is the default from MySQL 8.0.2, and row-based replication was in effect, updates to a table containing virtual generated fields could raise an assertion. The issue was caused by an error when generating string representations of the virtual generated fields in order to create hashes for use in searches. To remove the issue, MySQL no longer creates hashes for virtual generated fields. (Bug #26280724)

- The server permitted `SHOW CREATE TABLE` on nontable files created for full-text searching. (Bug #26271244)

- For debug builds, `CREATE OR REPLACE VIEW` for an existing view raised an assertion for column names greater than 64 characters. Now an appropriate error is reported. (Bug #26266789)

- `SET PERSIST` could be ineffective due to sorting variables written to `mysqld-auto.cnf`. Variables are now written in the order persisted. (Bug #26259671)

- Attempting a partial backup with `mysqlpump` on a GTID-enabled server failed and produced an error message suggesting incorrectly that this was not possible. (It is possible using the `--set-gtid-purged` option.) (Bug #26199978)

- `GRANT GRANT OPTION ON *.* TO` *`user`* granted `GRANT OPTION` for static but not dynamic privileges. `REVOKE ALL ... FROM CURRENT_USER()` revoked static but not dynamic privileges. (Bug #26191109, Bug #25658967)

- Error logging could attempt to log freed messages, resulting in a server exit. (Bug #26188656, Bug #86562)
- A `HAVING` condition was optimized away for an alias on an aggregate column where there was no `GROUP BY` clause. (Bug #26188578)
- `ST_Crosses()` could return an incorrect result when at least one parameter is a geometry collection and multiple elements of the geometry collection must be taken into account in order to determine whether the geometries cross. (Bug #26188208, Bug #86559)
- `MBROverlaps()` incorrectly returned false for two crossing perpendicular lines. (Bug #26188118, Bug #86558)
- `ALTER USER user DEFAULT ROLE ALL` produced an error. (Bug #26174169)
- `mysqldump` exited abnormally for large `--where` option values. (Bug #26171967, Bug #86496, Bug #27510150)
- A query using a window function with a window which partitioned or ordered on the result of an aggregate function where this evaluated as `NULL` returned incorrect results. (Bug #26164633)
- The Performance Schema could leak memory due to nondeletion of file instances created for `ALTER TABLE` operations that used the table-copy algorithm. (Bug #26152751, Bug #86482)
- Failed creation of a temporary table using `REPLACE` could under some circumstances result in an assertion in a later statement. (Bug #26126789, Bug #86422)
- `mysqlpump` did not properly parse `TABLESPACE` clauses in the result from `SHOW CREATE TABLE` statements it executed to determine table structure. (Bug #26116415)
- The binary file for the `udf_example` loadable function was omitted from binary distributions. (Bug #26115002, Bug #29178542)
- Long `SET PERSIST` statements could cause a server exit. (Bug #26100122)
- An incorrect formula was used to calculate maximum length of result strings for a few string functions: `QUOTE()`, `AES_DECRYPT()`, and `WEIGHT_STRING()`. This could affect, for example, the length of character columns created for `CREATE TABLE ... AS SELECT ... QUOTE()`. (Bug #26049942, Bug #86305)
- Schema creation and removal operations could fail due to checking for schema directories under the data directory rather than checking the data dictionary. (Bug #26043994, Bug #86282)
- `SHOW PLUGINS` did not handle plugins that were terminating, resulting in a server exit. The statement now displays the status for such plugins as `DELETING`. (Bug #26029765, Bug #86243)
- Some statements could cause a buffer overflow in the digest code. Thanks to Laurynas Biveinis and Roel van de Paar for the patch. (Bug #26021187)
- Previously, when the Performance Schema failed to initialize, it wrote a nonspecific `init failed` warning to the error log. Now it prints more specific messages about which memory allocation failed. (Bug #25996291)
- Incorrect results could occur on a table with a unique index when the optimizer chose a Loose Index Scan even though the unique index had no index extensions. (Bug #25989915, Bug #86165, Bug #26532061, Bug #87207)

  References: This issue is a regression of: Bug #21749123, Bug #78244.

- For `XA COMMIT`, precommit handling could set an error in the diagnostics area that was not reported correctly on the calling side, causing an assertion to be raised. (Bug #25978684, Bug #86142)
- The `MIN_VALUE` column of the Performance Schema `variables_info` table displayed incorrect values on 32-bit big-endian platforms. (Bug #25967079)
- A memory leak occurred when the optimizer excluded a subquery associated with a temporary table. (Bug #25951134)
- An assertion could be raised for updates when a view or derived table considered read only had nested references not seen as read only. (Bug #25832861, Bug #85796)
- Certificate and key files automatically generated by the server could have an incorrect access mode. (Bug #25832856)
- Queries on the `INFORMATION_SCHEMA TABLES` and `STATISTICS` tables, if evaluated using Index Condition Pushdown, could push down internal data dictionary functions, resulting in an assertion being raised. (Bug #25820175, Bug #85765)
- `ST_AsText()` could read freed memory. (Bug #25818451)
- `CREATE USER IF NOT EXISTS` was not written to the binary log if the user existed. This could result in inconsistent replication behavior if the user did not exist on slave servers. A similar issue occurred for `ALTER USER IF EXISTS`. To avoid inconsistencies, these statements now are written to the binary log. (Bug #25813089, Bug #85733)
- An invalid `utf8` input string caused a heap buffer overflow error. (Bug #25811623, Bug #25946444)
- A race condition made it possible to cause a server exit by persisting variables from multiple sessions simultaneously. (Bug #25768813)
- Plugins can create or drop `INFORMATION_SCHEMA` tables, but views that reference `INFORMATION_SCHEMA` tables were not validated when plugins were unloaded or unloaded. (Bug #25761992, Bug #85579)
- `mysql` wrote some password-related statements to the `.mysql_history` file. (Bug #25750609)
- Incorrect handling of internal memory buffers could cause a server exit. (Bug #25737271)
- On a read-only server with GTIDs enabled, a `DROP TEMPORARY TABLE IF EXISTS` statement relating to a nonexistent or filtered table could write an unnecessary transaction to the binary log and create an unnecessary GTID. In this situation, the missing temporary table was previously assumed to be transactional, leading to the statement being split. Now, MySQL checks that the temporary table exists and that its `DROP TEMPORARY TABLE` statement is recorded in the binary log. If this is not the case, no GTID is created. Thanks to Laurynas Biveinis for the patch. (Bug #25656992, Bug #85258)
- The MeCab full-text parser plugin failed to load on Windows. (Bug #25633175)
- The `SET_TIME` column of the Performance Schema `variables_info` table was initialized incorrectly. (Bug #25608115)
- Executing a stored procedure containing a statement that created a table from the contents of certain `SELECT` statements could result in a memory leak. (Bug #25586773)
- The Performance Schema failed to check the maximum host length for client connections. (Bug #25510805)
- For spatial functions, some set operations produced a result with SRID 0 when given arguments in a different SRID. (Bug #25510403)

- Large `--ssl-cipher` values could cause client programs to exit. (Bug #25483593)
- A missing argument-count check during preparation of a stored procedure call could result in a server exit. (Bug #25398451, Bug #84512)
- Temporary tables used in processing a recursive common table expression with `UNION DISTINCT` and a great many columns now use the `MEMORY` engine instead of `InnoDB`. (Bug #25190109)
- If the MySQL `root` user account was renamed, a query that accessed an `INFORMATION_SCHEMA` view returned an error stating that the user specified as the definer does not exist. To avoid this error, a new reserved account, `'mysql.infoschema'@'localhost'`, is now the `DEFINER` for `INFORMATION_SCHEMA` views. (Bug #25185947, Bug #84027)
- When an `UPDATE` required a temporary table having a primary key larger than 1024 bytes and that table was created using `InnoDB`, the server could exit. (Bug #25153670)
- `SET DEFAULT ROLE` was not transactional like other account-management statements. (Bug #25122897)
- `mysqlpump` included the `gtid_executed` table in dumps of the `mysql` system database, causing the `gtid_executed` position to be lost upon server restart after the dump was reloaded. `mysqlpump` no longer dumps the `gtid_executed` table. (Bug #25109007)
- For geometry calculations, invalid input parameters could lead to an incorrect result buffer and cause an assertion to be raised or a server exit. (Bug #25062396)
- `IFNULL(decimal, int)` could lose a digit after the decimal point when used in a query that included `GROUP BY` and was executed using a temporary table. (Bug #25051195, Bug #83699)
- For some queries, such as those involving `UNION`, column width for `GROUP_CONCAT()` could be calculated incorrectly, leading to incorrect application of `group_concat_max_len`. (Bug #25050090, Bug #83667)
- Audit logging of events for the Performance Schema `global_variables` table was improved so as to not report events for rows materialized but not reported to the SQL layer. (Bug #24970428)
- `ST_Buffer()` could return an invalid result or raise an error for some inputs that should have produced a valid output geometry. (Bug #24947868, Bug #26735293, Bug #25662426)
- For builds with AddressSanitizer or Undefined Behavior Sanitizer enabled, division by zero could occur during Performance Schema timer initialization. (Bug #24785784)
- Operations that rely heavily on the metadata locking (MDL) subsystem caused a performance degradation. Traversal of MDL ticket lists was time consuming in cases where there were large number of MDL tickets. (Bug #24734971, Bug #83143)
- When `binlog_format` is `ROW` or `MIXED`, operations on temporary tables are not logged. Previously, the exception to this rule was that when the connection was terminated at the end of the session, the statement `DROP TEMPORARY TABLE IF EXISTS` was logged for any temporary tables that had been opened in the session. For row-based replication, this behavior caused an unnecessary write to the binary log, and added a transaction sequence number for the GTID where these were enabled.

  Now, when a temporary table is created in a session, the binary logging format is tracked. The `DROP TEMPORARY TABLE IF EXISTS` statement is only logged at the end of the session if statement-based format was in effect when the temporary table was created, so the `CREATE TEMPORARY TABLE` statement was logged. If row-based or mixed-format binary logging was in use when the table was created, the `DROP TEMPORARY TABLE IF EXISTS` statement is not logged.

Thanks to Laurynas Biveinis for the patch. (Bug #24670909, Bug #83003, Bug #28606948)

- Concurrent `INSERT`, `ALTER TABLE`, and `DROP DATABASE` operations could result in deadlock. (Bug #24510948, Bug #82704)
- Under some conditions, the `audit_log` plugin could recursively lock a mutex, resulting in an unresponsive server. (Bug #24437533)
- In some cases, the optimizer chose a Loose Index Scan (`QUICK_GROUP_MIN_MAX_SELECT`) for a `GROUP BY` query even when there was a predicate with a disjunction. This is fixed by not performing a range scan when the condition in the `WHERE` clause results in more than one disjoint range tree. (Bug #24423143)
- Incorrect results could occur when the optimizer chose an index on a generated column to fetch values. (Bug #24345509, Bug #29451999)
- Debug symbol packages are now included for all `apt` platforms (previously, they were only available on Debian 9). (Bug #24008883, Bug #27990381)
- With `SQL_MODE=''`, `UNIX_TIMESTAMP(COUNT(1))` returned `NULL` instead of 0 as expected. (Bug #23529242)
- When deleting rows from a table that had an indexed virtual `BLOB` column with a `NOT NULL` constraint, and the generated column expression evaluated to `NULL` in one of the rows that were being deleted, conversion of the `NULL` to its `NOT NULL` equivalent was not performed correctly. (Bug #23321196)

  References: See also: Bug #23037025, Bug #21345972.
- Setting the `MYSQL_GROUP_SUFFIX` environment variable had no effect. (Bug #23072792)
- A missing check for error handling during generated column evaluation could result in a server exit. (Bug #23021693)
- To check whether a table was empty, `ALTER TABLE` performed a table scan, which is inefficient. (Bug #22688065)
- Failure to acquire tablespace metadata locks for `ALTER TABLE` when a `LOCK TABLES` was active could cause an assertion to be raised. (Bug #22486020, Bug #79820)
- Constant propagation is no longer performed when a constant expression contains a reference to the column it is meant to replace. (Bug #20964700)
- Queries with many left joins were slow if join buffering was used (for example, using the block nested loop algorithm). (Bug #18898433, Bug #72854)
- `REGEXP` failed to find matches occurring after a `\0` character in the string expression. (Bug #17541193, Bug #70470)
- Selecting from a view that involved aggregation and `WITH ROLLUP` could result in a spurious `Column *col_name* cannot be null` error. (Bug #11755860, Bug #47693)
- `COALESCE()` could change the value of `FLOAT` fields. (Bug #11751705, Bug #42666)
- In some cases, the row estimate used by the server to determine whether to use sampling could be inaccurate. This was because the histogram process assumed that the estimate for the number of rows in the table was current, although it was not updated by (for example) `INSERT` or `DELETE` statements. Now the histogram process requests an updated count of rows. (Bug #88710, Bug #27197709)

- Long-running regular expression matches could not be killed. (Bug #88676, Bug #27183583)
- An optimizer `SET_VAR` hint (see Variable-Setting Hint Syntax) setting `cte_max_recursion_depth` was ignored. (Bug #88594, Bug #27153338)
- When handling range frames, if the first row for a range frame was found, its position was not stored. This could later cause retrieval of the row from the frame buffer to fail. (Bug #88568, Bug #27149369)
- The server did not handle triggers activated by `LOAD DATA` correctly when `--skip-log-bin` was enabled. (Bug #88516, Bug #27128534)

  References: This issue is a regression of: Bug #27041382.
- Stored procedures performing XA transactions and acting on views were not executed correctly. (Bug #88326, Bug #27058931)
- When a table contained a column whose length was zero, the optimizer could in some cases allocate a record buffer that was too small to hold the columns read by the query. (Bug #88283, Bug #27041288)
- A trigger containing invalid syntax, followed by an `INSERT` that did not specify a column list, attempted to insert a new row regardless. (Bug #88274, Bug #27041382)
- Window functions did not always produce correct results with `LAST_VALUE()` and frames having multiple `ORDER BY` expressions. (Bug #88186, Bug #27010574)
- A fix for a previous issue caused the aggregated data type to be set to `VARCHAR` whenever the result type was string and the column size was larger than 255 characters (maximum length for `CHAR`). This caused problems for data types such as `JSON` whose result type is a string, but which support field values longer than 255 characters. Now in such cases the data type is set explicitly to `VARCHAR` when an aggregated column is of type `CHAR` or `BINARY` (both represented internally as strings) but its size exceeds the maximum for `CHAR`. (Bug #88073, Bug #26960106)

  References: This issue is a regression of: Bug #83895, Bug #25123839.
- A prepared statement containing an `ORDER BY` list that referred to a parameter was not always handled correctly. (Bug #87863, Bug #26867652)
- `DENSE_RANK()` did not work correctly for the first row in a partition when buffering was in use, due to premature initialization of the cache comparator for `ORDER BY`. (Bug #87760, Bug #26802696)
- The optimizer chose a composite index for `ref` access where only the first part of the key could be used. The composite key was suitable but was seen as a higher cost. This was because, when choosing between `ref` access and `range` access on the same index, we prefer `range` if certain criteria are fulfilled, one of these being to choose to avoid `ref`-access if it has an overly-optimistic or unrealistically low cost as can happen when `records_per_key` is very low. This was done even if the estimate of the number of rows for `range` access was more reliable than the estimate for `ref` access. (Bug #87613, Bug #26727773)

  References: See also: Bug #23259872.
- When a stored function was used with a table column value as an argument in a `WHERE` predicate, its internal `not_null_tables` property was falsely set to a nonempty value. If this predicate was applied to an outer join operation and one of the arguments was from an inner table of the outer join, the predicate was sometimes used (incorrectly) to convert the outer join to an inner join. According to the SQL standard, only functions that have the `RETURNS NULL ON NULL INPUT` property should behave in that manner. Since MySQL does not currently implement this property, stored functions are changed such that they no longer implement the `RETURNS NULL ON NULL INPUT` behavior. (Bug #86922, Bug #26389402)

- A view or derived table contained incorrect data when defined using a `SELECT` that performed aggregation of a column, and whose result was filtered with `HAVING`. (Bug #86840, Bug #26360114)
- The server handled triggers and generated columns incorrectly. (Bug #86637, Bug #26251621)
- A query that grouped results on a subquery which returned a `BLOB` (or a type based on `BLOB` such as `JSON`) sometimes failed to find the group boundaries, and so returned incorrect results. (Bug #78787, Bug #21974696)

# Changes in MySQL 8.0.3 (2017-09-21, Release Candidate)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

**Note**

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward. (Making a backup before the upgrade is a prudent precaution in any case.)


- Account Management Notes
- Atomic DDL Notes
- C API Notes
- Character Set Support
- Compilation Notes
- Configuration Notes
- Data Dictionary Notes
- Deprecation and Removal Notes
- InnoDB Notes
- Logging Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Plugin Notes
- Security Notes
- Server Administration
- Spatial Data Support

- SQL Syntax Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Account Management Notes

- `CREATE USER` now permits a `DEFAULT ROLE` clause enabling the account default roles to be specified.

  `SHOW CREATE USER` now displays the account default roles if the default is not `NONE`. (Bug #24670738, Bug #82987)

- MySQL now maintains information about password history, which makes it possible to enable restrictions on reuse of previous passwords. DBAs can require that new passwords not be selected from previous passwords for some number of password changes or period of time. It is possible to establish password-reuse policy globally using the `password_history` and `password_reuse_interval` system variables, as well as on a per-account basis using the `CREATE USER` and `ALTER USER` statements. Together with existing password-expiration capabilities to require that passwords be changed periodically, the new reuse-restriction capabilities provide DBAs more complete control over password management. For more information, see Password Management.

  **Important**

  The implementation of password-reuse restrictions involves a change to the structure of the `mysql.user` system table and a new `mysql.password_history` system table. If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these system database changes. Until this is done, *password changes are not possible.*

  (WL #6595)

## Atomic DDL Notes

- MySQL now supports atomic data definition statements (*atomic DDL*). An atomic DDL statement combines the data dictionary updates, storage engine operations, and binary log writes associated with a DDL operation into a single, crash-safe, transaction that is either fully committed or rolled back.

  Both table and nontable DDL statements are supported. Table-related DDL operations require storage engine support, whereas nontable DDL operations do not. Currently, the `InnoDB` storage engine supports atomic DDL.

  - Supported table DDL statements include `CREATE`, `ALTER`, and `DROP` statements for databases, tablespaces, tables, and indexes, and the `TRUNCATE TABLE` statement.
  - Supported nontable DDL statements include:
    - `CREATE` and `DROP` statements, and, if applicable, `ALTER` statements for stored programs, triggers, views, and loadable functions. Atomic DDL support for `CREATE TRIGGER` and `DROP TRIGGER` was added in MySQL 8.0.0.
    - Account management statements: `CREATE`, `ALTER`, `DROP`, and, if applicable, `RENAME` statements for users and roles, as well as `GRANT` and `REVOKE` statements. Atomic DDL support for account management statements was added in MySQL 8.0.1.

For table-related DDL operations, `InnoDB` writes DDL logs to the `mysql.innodb_ddl_log` data dictionary table. Enabling the `innodb_print_ddl_logs` configuration option prints DDL recovery logs to `stderr`.

The atomic DDL feature changes the behavior of some statements:

- `DROP VIEW` fails with an error if a named view does not exist, and no changes are made. Previously, the statement returned an error indicating which views did not exist, but also dropped the views that did exist.
- `DROP TABLE` fails with an error if a named table does not exist, and no changes are made. Previously, the statement returned an error indicating which tables did not exist, but also dropped the tables that did exist.
- `DROP TABLE` is fully atomic if all named tables use an atomic DDL-supported storage engine.
- `DROP DATABASE` is atomic if all tables use an atomic DDL-supported storage engine. However, removal of the database directory from the file system occurs last and is not part of the atomic transaction. If removal of the database directory fails due to a file system error or server halt, the `DROP DATABASE` transaction is not rolled back.
- Interrupted DDL operations on tables that use an atomic DDL-supported storage engine no longer introduce discrepancies between the storage engine, data dictionary, and binary log, or leave behind orphan files.
- Partial execution of account management statements is no longer permitted. Account management statements either succeed for all named users or roll back and have no effect if an error occurs.

Changes to `DROP TABLE`, `DROP VIEW`, and account management statement behavior have implications for cross-version replication configurations.

For more information, see Atomic Data Definition Statement Support. (Bug #24620918, WL #9173, WL #9536, WL #7896, WL #9045, WL #7743)

## C API Notes

- The MySQL C API now enables clients to specify that metadata transfer for result sets is optional. Suppression of metadata transfer can improve performance, particularly for sessions that execute many queries that return few rows each. For more information, see Optional Result Set Metadata. (WL #8134)

## Character Set Support

- MySQL now supports Russian collations for the `utf8mb4` Unicode character set:
  - `utf8mb4_ru_0900_ai_ci` is accent insensitive and case insensitive.
  - `utf8mb4_ru_0900_as_cs` is accent sensitive and case sensitive.

  (WL #10753)

## Compilation Notes

- For debug builds, the `SAFE_MUTEX` compilation flag was disabled if the `memcached` plugin was included in the build. This no longer occurs; `SAFE_MUTEX` is always enabled for debug builds. Some code issues found as a result of this change were corrected. (Bug #26442367, Bug #87068)

- Binary packages on EL6 and EL7 now are compiled using Devtoolset 6 rather than Devtoolset3 and GCC 6.2.1 rather than 4.9.2. (Bug #26436968, Bug #87061)
- MySQL now compiles for SPARC on Oracle Linux. (Bug #26306331, Bug #86745)
- MySQL compilation on macOS using Clang now requires a Clang version different from 8.0, which has problems with certain inline constructs. (Bug #26279510, Bug #86711)
- Work was done to clean up the source code base, including: Removing unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

## Configuration Notes

- The `performance_schema_max_mutex_classes` system variable default value has been increased from 220 to 250. The `performance_schema_max_thread_classes` system variable default value has been increased from 50 to 100. (Bug #26193630)
- The new `cte_max_recursion_depth` system variable implements a common table expression (CTE) maximum recursion depth. The server terminates execution of any CTE that recurses more levels than the value of this variable. For more information, see Limiting Common Table Expression Recursion. (Bug #26136509, Bug #86444, WL #10972)
- The `back_log` system variable default value is now the value of `max_connections`, which enables the permitted backlog to adjust to the maximum permitted number of connections. (WL #9704)
- To enable the Event Scheduler by default, the `event_scheduler` system variable default value was changed from `OFF` to `ON`. (WL #9644)
- The `max_allowed_packet` system variable default value has been increased from 4194304 (4M) to 67108864 (64M). (WL #8393)
- The `max_error_count` system variable default variable has been increased from 64 to 1024. (WL #9686)

## Data Dictionary Notes

- These `INFORMATION_SCHEMA` tables have been reimplemented as views on data dictionary tables:

  ```
  FILES
  PARTITIONS
  REFERENTIAL_CONSTRAINTS
  ```

  Queries on those tables are now more efficient because they obtain information from data dictionary tables rather than by other, slower means. For example, the server no longer must create a temporary table for each query of the `INFORMATION_SCHEMA` table.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes. (WL #9814, WL #11059)

## Deprecation and Removal Notes

- **Replication:** The following obsolete `mysqlbinlog` options are now deprecated and will be removed in a future MySQL version:
  - `--short-form`, which could be used for testing to limit the output to statements alone.

- `--stop-never-slave-server-id`, which provided a server ID for connections using the `--stop-never` option. If you require this function, use the `--connection-server-id` option instead.

The deprecation warnings for these options are sent to standard error, rather than to standard output, so that they do not interfere with the operation of tools that use the output of `mysqlbinlog`. (WL #9632, WL #9633)

- **Replication:** The deprecated global scope for the `sql_log_bin` system variable has been removed. `sql_log_bin` now has session scope only. Applications that rely on accessing `@@GLOBAL.sql_log_bin` should be adjusted. (WL #10922)
- These encryption-related deprecated items have been removed:
  - The `ENCODE()` and `DECODE()` functions.
  - The `ENCRYPT()` function.
  - The `DES_ENCRYPT()`, and `DES_DECRYPT()` functions, the `--des-key-file` option, the `have_crypt` system variable, the `DES_KEY_FILE` option for the `FLUSH` statement, and the `HAVE_CRYPT` `CMake` option.

  In place of the removed encryption functions: For `ENCRYPT()`, consider using `SHA2()` instead for one-way hashing. For the others, consider using `AES_ENCRYPT()` and `AES_DECRYPT()` instead. (Bug #26493987, WL #10788, WL #10789)
- The deprecated `tx_isolation` and `tx_read_only` system variables have been removed. Use `transaction_isolation` and `transaction_read_only` instead. (WL #9636)
- The deprecated query cache has been removed. Removal includes these items:
  - The `FLUSH QUERY CACHE` and `RESET QUERY CACHE` statements.
  - These system variables: `query_cache_limit`, `query_cache_min_res_unit`, `query_cache_size`, `query_cache_type`, `query_cache_wlock_invalidate`.
  - These status variables: `Qcache_free_blocks`, `Qcache_free_memory`, `Qcache_hits`, `Qcache_inserts`, `Qcache_lowmem_prunes`, `Qcache_not_cached`, `Qcache_queries_in_cache`, `Qcache_total_blocks`.
  - These thread states: `checking privileges on cached query`, `checking query cache for query`, `invalidating query cache entries`, `sending cached result to client`, `storing result in query cache`, `Waiting for query cache lock`.
  - The `SQL_CACHE` `SELECT` modifier.

  These deprecated query cache items remain deprecated, but have no effect, and will be removed in a future MySQL version:
  - The `SQL_NO_CACHE` `SELECT` modifier.
  - The `ndb_cache_check_time` system variable.

  The `have_query_cache` system variable remains deprecated, always has a value of `NO`, and will be removed in a future MySQL version. (WL #10824)
- The deprecated `EXTENDED` and `PARTITIONS` keywords for the `EXPLAIN` statement have been removed. These keywords are unnecessary because their effect is always enabled. (WL #9678)

- The unused `date_format`, `datetime_format`, `time_format`, and `max_tmp_tables` system variables have been removed. (WL #9680)

- The deprecated `multi_range_count` system variable has been removed. (WL #10908)

- The deprecated `log_warnings` system variable and `--log-warnings` server option have been removed. Use the `log_error_verbosity` system variable instead. (WL #9676)

- The deprecated `secure_auth` system variable and `--secure-auth` client option have been removed. The `MYSQL_SECURE_AUTH` option for the `mysql_options()` C API function was removed. (WL #9674)

- The deprecated `ignore_builtin_innodb` system variable has been removed. (WL #9675)

- In MySQL 8.0.2, the system variables for the slave status logs, `master_info_repository` and `relay_log_info_repository`, were set to `TABLE` instead of `FILE` by default. In MySQL 8.0.3, the `FILE` setting for both these system variables is deprecated, and a warning is issued if it is used. The `FILE` setting will be removed in a future MySQL version.

  The `TABLE` setting ensures that replication repository information is stored in InnoDB tables, rather than in files in the data directory. The use of tables makes replication resilient to unexpected halts.

  The default names for the slave status logs when stored as files were `master.info` and `relay-log.info`. The names could be changed using the `--master-info-file` and `--relay-log-info-file` options, respectively. As InnoDB tables, the slave status logs are named `mysql.slave_master_info` and `mysql.slave_relay_log_info`.

  To modify an existing replication slave that is using a `FILE` repository for the slave status logs to use `TABLE` repositories, convert the existing replication repositories dynamically by running the following commands:

  ```
  STOP SLAVE;
  SET GLOBAL master_info_repository = 'TABLE';
  SET GLOBAL relay_log_info_repository = 'TABLE';
  ```

  The master info log table `mysql.slave_master_info` should be protected because it contains the password for connecting to the master. When you back up the replication slave's data, ensure that you back up the `mysql.slave_master_info` and `mysql.slave_relay_log_info` tables containing the slave status logs, because they are needed to resume replication after you restore the data from the slave. (WL #6959)

## InnoDB Notes

- Renaming of columns in a parent foreign key is temporarily disabled due to ongoing work on foreign key locking. This restriction will be lifted in MySQL 8.0.4. (Bug #26334071)

  References: See also: Bug #26659110.

## Logging Notes

- The binary log is now enabled by default at server startup. The `log_bin` system variable is set to ON by default, instead of OFF, even if the `--log-bin` option has not been specified. Binary logging is standard practice for production installations, so enabling it by default removes configuration and planning steps that were usually required.

  To disable binary logging, you can specify the `--skip-log-bin` or `--disable-log-bin` option at startup.

The `server_id` system variable is now set to 1 by default, instead of 0. For servers in a replication topology, you must still change this setting to specify a unique server ID for each replication server. Previously, the server could not start with `log_bin=ON` if no server ID was specified. Now, the server can start, but a warning message is issued if you did not set an explicit server ID. (From MySQL 8.0.4, the message is only informational.)

With binary logging enabled for a server, all statements that change data are logged to the server's binary log, which is a sequence of files with a base name and numeric extension. By default, the server creates binary log files and an index file in the data directory. In MySQL 8.0.3, the default base name of these files is *`host_name`*`-bin`, using the name of the host machine. From MySQL 8.0.4, the default base name is `binlog` if you do not supply the `--log-bin` option, and *`host_name`*`-bin` if you supply the option with no string or an empty string.

You can choose the names and locations of the binary log files and index file by specifying the `--log-bin` and `--log-bin-index` options. You are recommended to specify a base name explicitly, so that if the host name changes, you can easily continue to use the same binary log file names. The `log_bin_basename` system variable holds the base name and any specified path for the binary log files.

The relay log and relay log index on a replication slave, whose names are specified by the `--relay-log` and `--relay-log-index` options, cannot be given the same names as the binary log and binary log index. From MySQL 8.0.3, the server issues an error message and does not start if the binary log and relay log file base names would be the same.

The server creates a new binary log file in the series each time it starts or flushes the logs. The server also creates a new binary log file automatically after the current file's size reaches `max_binlog_size`, which defaults to the maximum permitted value of 1GB. In MySQL 8.0.3, binary log files expire by default after 30 days, and can then be automatically removed at startup or when the binary log is flushed. You can purge binary log files manually using the `PURGE BINARY LOGS` statement, or specify a different binary log expiration period using the `binlog_expire_logs_seconds` system variable.

Many other options are available to modify the behavior of binary logging. For more information, see The Binary Log and Binary Logging Options and Variables. (WL #10470)

References: See also: Bug #26730000.

## Optimizer Notes

- The optimizer now supports a `SET_VAR` hint that sets the session value of a system variable for the duration of a single statement. Examples:

```
SELECT /*+ SET_VAR(sort_buffer_size = 16M) */ name FROM people ORDER BY name;
INSERT /*+ SET_VAR(foreign_key_checks=OFF) */ INTO t2 VALUES(2);
```

  For more information, see Optimizer Hints. (Bug #22906815, WL #681)

- The optimizer now uses column-value histogram statistics stored in the `column_statistics` data dictionary table to construct query execution plans. Histogram use applies to predicates involving comparison of a column to a constant. See Optimizer Statistics. (WL #9223)

- Previously, there was no way of skipping the use of index dives to estimate index usefulness, except by using the `eq_range_index_dive_limit` system variable. Now index dive skipping is possible for single-table queries under certain query conditions (see Range Optimization). (WL #6526)

- The `optimizer_switch` system variable has a new flag named `use_invisible_indexes` to control whether the optimizer uses invisible indexes for query execution plan construction. If the flag is `off` (the

default), the optimizer ignores invisible indexes (the same behavior as prior to the introduction of this flag). If the flag is `on`, invisible indexes remain invisible but the optimizer takes them into account for execution plan construction. (WL #10891)

## Packaging Notes

- `mysqlcheck` was missing in the MySQL Server Docker image, which prevented `mysql_upgrade` from running. (Bug #26400146, Bug #86968)
- For Debian, non-debug binaries were moved from the `mysql-server` package to the `mysql-server-core` package. (Bug #26382333, Bug #86899)
- The Debian/Ubuntu `mysql-community-source` package is no longer produced because the MySQL source tarball it contained is provided by other packages at `dev.mysql.com`. (Bug #26201482)
- The zlib library bundled with MySQL has been upgraded from version 1.2.3 to version 1.2.11. MySQL implements compression with the help of the zlib library.

  The zlib `compressBound()` function in zlib 1.2.11 returns a slightly higher estimate of the buffer size required to compress a given length of bytes than it did in zlib version 1.2.3. The `compressBound()` function is called by `InnoDB` functions that determine the maximum row size permitted when creating compressed `InnoDB` tables or inserting and updating rows in compressed `InnoDB` tables. As a result, `CREATE TABLE ... ROW_FORMAT=COMPRESSED`, `INSERT`, and `UPDATE` operations with row sizes very close to the maximum row size that were successful in earlier releases could now fail. To avoid this issue, test `CREATE TABLE` statements for compressed `InnoDB` tables with large rows on a MySQL 8.0 test instance prior to upgrading. (WL #10551)

## Performance Schema Notes

- As of MySQL 8.0.2, Performance Schema table definitions are maintained internally to the server. In consequence of that change, `CREATE TABLE` and `DROP TABLE` are no longer possible for Performance Schema tables. (Bug #26136994)
- The `events_statements_summary_by_digest` table now provides, for each row, a sample statement that produces the digest value in the row. Applications can use this information as a more efficient means of capturing statement samples than alternatives such as probing the *`xxx`*`_history_long` tables. The latter approach requires enabling the corresponding *`xxx`*`_history_long` consumers, which is additional overhead for applications that do not otherwise need those tables. For more information, see Performance Schema Statement Digests and Sampling, and Statement Summary Tables.

  Additionally, the `FIRST_SEEN` and `LAST_SEEN` timestamp columns of the `events_statements_summary_by_digest` table now have a fractional seconds part. (WL #9830)
- The Performance Schema `setup_instruments` table now has columns for instrument metadata: Instrument properties, instrument volatility, and a documentation string describing the instrument purpose. Also, the `TIMED` column now can be `NULL`, indicating that the instrument does not support timing. See The setup_instruments Table.

  The new Performance Schema `setup_threads` table exposes instrumented thread class names and attributes. See The setup_threads Table. (WL #7801)

## Plugin Notes

- The new `get_sysvar_source` plugin service enables plugins to retrieve the source of system variable settings.

**Note**

As part of this work, the compound-part structured-variable syntax used for referring to multiple `MyISAM` key caches is deprecated. See Multiple Key Caches.

(WL #9424)

## Security Notes

- A new `caching_sha2_password` authentication plugin is available. Like the `sha256_password` plugin, `caching_sha2_password` implements SHA-256 password hashing, but uses caching to address latency issues at connect time. It also supports more connection protocols and does not require linking against OpenSSL for RSA password-exchange capabilities. See Caching SHA-2 Pluggable Authentication. (WL #9591)
- Because starting the server with `--skip-grant-tables` disables authentication checks, the server also disables remote connections in that case by enabling `skip_networking`. (WL #4321)

## Server Administration

- MySQL now supports creation and management of resource groups, and permits assigning threads running within the server to particular groups so that threads execute according to the resources available to the group. Group attributes enable control over its resources, to enable or restrict resource consumption by threads in the group. DBAs can modify these attributes as appropriate for different workloads. Currently, CPU time is a manageable resource, represented by the concept of “virtual CPU” as a term that includes CPU cores, hyperthreads, hardware threads, and so forth. The server determines at startup how many virtual CPUs are available, and database administrators with appropriate privileges can associate these CPUs with resource groups and assign threads to groups. For more information, see Resource Groups. (WL #9467)

## Spatial Data Support

- **Incompatible Change:** Previously, these functions that test geometry relationships supported only Cartesian spatial reference systems (SRSs): `ST_Contains()`, `ST_Crosses()`, `ST_Disjoint()`, `ST_Equals()`, `ST_Intersects()`, `ST_Overlaps()`, `ST_Touches()`, `ST_Within()`, `MBRContains()`, `MBRCoveredBy()`, `MBRCovers()`, `MBRDisjoint()`, `MBREquals()`, `MBRIntersects()`, `MBROverlaps()`, `MBRTouches()`, `MBRWithin()`.

  These functions now detect geometry arguments in a geographic SRS and return geographic results. Calculations for projected SRSs and SRID 0 remain the same. For more information, see Spatial Relation Functions That Use Object Shapes, and Spatial Relation Functions That Use Minimum Bounding Rectangles.

  **Note**

  If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

  (WL #10827)

- **InnoDB:** Spatial reference identifier (SRID) support was added for `InnoDB` spatial indexes. (WL #10439)
- Spatial data types now permit an `SRID` attribute, to explicitly indicate the spatial reference system (SRS) for values stored in the column. See Spatial Data Types.

To indicate each column's `SRID` attribute value, if there is one, the `INFORMATION_SCHEMA.COLUMNS` table now has an `SRS_ID` column.

A spatial column with an explicit `SRID` attribute is SRID-restricted: The column takes only values with that ID, and `SPATIAL` indexes on the column become subject to use by the optimizer. The optimizer ignores `SPATIAL` indexes on spatial columns with no `SRID` attribute. See SPATIAL Index Optimization. (WL #8592)

## SQL Syntax Notes

- `ALTER TABLE` now supports easier column renaming using `RENAME COLUMN old_name TO new_name` syntax. See ALTER TABLE Statement. For changing a column name but not its definition, `RENAME COLUMN` is more convenient than `CHANGE`, which requires respecifying the current column definition. With `CHANGE`, you must look up the definition if you do not know it, and if you do not respecify it exactly, there is a possibility of data change or loss. (Bug #11746522, Bug #26949, Bug #11747473, Bug #32497, Bug #11765084, Bug #58006, Bug #14031617, WL #10761)

## X Plugin Notes

- The X Plugin could not be installed when the server was started with the `--skip-grant-tables` option. (Bug #26516678)
- X Plugin socket connections were not working correctly. (Bug #26427112, Bug #87019)
- When compiling MySQL from source, certain infrequently used `CMake` arguments caused issues for the X Plugin build. (Bug #26141933)
- A `Mysqlx.Connection.CapabilitiesGet` request using X Protocol did not return the complete list of available authentication mechanisms. (Bug #26044113)
- For mixed case or uppercase schema names, the statement `list_objects` could incorrectly report a collection as a table. (Bug #25769683)
- The X Plugin was omitted from the list of plugins to include for testing data directory permissions. (Bug #24823999)

## Functionality Added or Changed

- **InnoDB:** The new `--innodb-dedicated-server` server option, can be used to have `InnoDB` automatically set the following options according to the amount of memory detected on the server:
  - `innodb_buffer_pool_size`
  - `innodb_log_files_in_group`
  - `innodb_flush_method`

  This option is intended for MySQL server instances that run on a dedicated server. For more information, see Enabling Automatic InnoDB Configuration for a Dedicated MySQL Server. (WL #9193)
- **InnoDB:** Renaming a general tablespace is now supported by `ALTER TABLESPACE ... RENAME TO` syntax.

  The `ALTER TABLESPACE` and `DROP TABLESPACE ENGINE` clause is deprecated and will be removed in a future MySQL version. (WL #8972)

- **InnoDB:** Code related to obsoleted `InnoDB` system tables was removed. `INFORMATION_SCHEMA` views based on `InnoDB` system tables were replaced by internal system views on data dictionary tables. Affected `InnoDB` `INFORMATION_SCHEMA` views were renamed:

  **Table 1 Renamed InnoDB Information Schema views. The first column shows the old name. The second column shows the new name.**

  | Old Name | New Name |
  |---|---|
  | `INNODB_SYS_COLUMNS` | `INNODB_COLUMNS` |
  | `INNODB_SYS_DATAFILES` | `INNODB_DATAFILES` |
  | `INNODB_SYS_FIELDS` | `INNODB_FIELDS` |
  | `INNODB_SYS_FOREIGN` | `INNODB_FOREIGN` |
  | `INNODB_SYS_FOREIGN_COLS` | `INNODB_FOREIGN_COLS` |
  | `INNODB_SYS_INDEXES` | `INNODB_INDEXES` |
  | `INNODB_SYS_TABLES` | `INNODB_TABLES` |
  | `INNODB_SYS_TABLESPACES` | `INNODB_TABLESPACES` |
  | `INNODB_SYS_TABLESTATS` | `INNODB_TABLESTATS` |
  | `INNODB_SYS_VIRTUAL` | `INNODB_VIRTUAL` |

  After upgrading to MySQL 8.0.3 or later, update any scripts that reference previous `InnoDB` `INFORMATION_SCHEMA` view names.

  The new `INFORMATION_SCHEMA.INNODB_TABLESPACES_BRIEF` view provides space, name, path, flag, and space type data for `InnoDB` tablespaces. (WL #9535)

- **InnoDB:** When `InnoDB` was integrated with the global data dictionary, file-per-table tablespace names in the data dictionary were created in the form of innodb_file_per_table_*x*, where *x* is the `InnoDB` tablespace ID. For ease of use, file-per-table tablespace names in the data dictionary are once again the same as the table name.

  Upgrading from MySQL 5.7 to MySQL 8.0 appends MySQL 5.7 `innodb_table_stats` and `innodb_index_stats` tablespace names in the data dictionary with "`_backup57`" to differentiate them from their MySQL 8.0 counterparts. (WL #10436)

- **InnoDB:** The default `innodb_autoinc_lock_mode` setting was changed from 1 (consecutive) to 2 (interleaved). Interleaved lock mode permits the execution of multi-row inserts in parallel, which improves concurrency and scalability. The new `innodb_autoinc_lock_mode` default setting reflects the change from statement-based replication to row based replication as the default replication type in MySQL 5.7. Statement-based replication requires the consecutive auto-increment lock mode to ensure that auto-increment values are assigned in a predictable and repeatable order for a given sequence of SQL statements, whereas row-based replication is not sensitive to the execution order of SQL statements. For more information, see InnoDB AUTO_INCREMENT Lock Modes.

  For systems that use statement-based replication, the new `innodb_autoinc_lock_mode` default setting may break applications that depend on sequential auto-increment values. To restore the previous default, set `innodb_autoinc_lock_mode` to 1. (WL #9699)

- **InnoDB:** Serialized dictionary information (SDI) is now present in all `InnoDB` tablespace files except for temporary tablespace and undo tablespace files. SDI is serialized metadata for table and tablespace objects. The presence of SDI data provides metadata redundancy. For example, dictionary object metadata may be extracted from tablespace files if the data dictionary becomes unavailable. SDI extraction is performed using the `ibd2sdi` tool. SDI data is stored in `JSON` format.

The inclusion of SDI data in tablespace files increases tablespace file size. An SDI record requires a single index page, which is 16KB in size by default. However, SDI data is compressed when it is stored to reduce the storage footprint. (WL #9538)

- **InnoDB:** The `innodb_flush_neighbors` default value was changed from 1 to 0, which disables flushing of neighboring pages from the buffer pool. A setting of 0 is optimal for non-rotational storage (SSD) devices where seek time is not a significant factor. For systems that use rotational storage (HDD), it is recommended to change the setting back to the previous default value of 1. (WL #9631)
- **InnoDB:** Default values for configuration options that affect buffer pool preflushing and flushing behavior were modified:
  - The `innodb_max_dirty_pages_pct_lwm` default value was changed to 10. The previous default value of 0 disables buffer pool preflushing. A value of 10 enables preflushing when the percentage of dirty pages in the buffer pool exceeds 10%. Enabling preflushing improves performance consistency.
  - The `innodb_max_dirty_pages_pct` default value was changed from 75 to 90. `InnoDB` attempts to flush data from the buffer pool so that the percentage of dirty pages does not exceed this value. The increased default value permits a greater percentage of dirty pages in the buffer pool.

  (WL #9707, WL #9630)
- **InnoDB:** The minimum `innodb_undo_tablespaces` value changed from 0 to 2. In previous releases, the system tablespace is used for rollback segments if `innodb_undo_tablespaces` is set to 0. A minimum value of 2 ensures that rollback segments are created in undo tablespaces instead of the system tablespace. For more information, see Undo Tablespaces. (WL #10583)
- **Replication; JSON:** Added the `binlog_row_value_options` system variable. Currently this variable can be unset, or set to the value `PARTIAL_JSON`. This causes MySQL's row-based replication to use a compact binary log format for each update modifying only a small portion of a JSON document and using any combination of `JSON_SET()`, `JSON_REPLACE()`, and `JSON_REMOVE()`. The compact format includes only the modified parts of the JSON document, not the full document, in the after-image used for the update in the binary log. If the modification requires more space than the full document, or if it is not possible to generate a partial update, the full document is used instead.

  See the description of the variable as well as Partial Updates of JSON Values, for more information. (WL #2955)
- **Replication:** The `IGNORE_SERVER_IDS` option of the `CHANGE MASTER TO` statement is now deprecated when using GTID-based replication (`gtid_mode=ON`). With GTIDs, transactions that have already been applied are automatically ignored, so this function is not needed.

  Before starting GTID-based replication, check for and clear all ignored server ID lists that have previously been set on the servers involved. The `SHOW_SLAVE_STATUS` statement, which can be issued for individual channels, displays the list of ignored server IDs if there is one. If there is no list, the `Replicate_Ignore_Server_Ids` field is blank.

  If `gtid_mode=ON` is set for the server, a deprecation warning is now issued if you include the `IGNORE_SERVER_IDS` option in a `CHANGE MASTER TO` statement. A deprecation warning is also issued if you issue a `SET GTID_MODE=ON` statement when any channel has existing server IDs set with `IGNORE_SERVER_IDS`. If you do receive the deprecation warning, you can still clear a list after `gtid_mode=ON` is set by issuing a `CHANGE MASTER TO` statement containing the `IGNORE_SERVER_IDS` option with an empty list. (WL #10963)
- **Replication:** The `log_slave_updates` system variable is now set to ON by default, so you do not need to specify `--log-slave-updates` explicitly when you start a replication slave.

The `log_slave_updates` system variable is read-only. If you need to prevent a replication slave from logging the updates performed by its SQL thread to its own binary log, specify `--log-slave-updates=OFF` at slave server startup. (WL #10479)

- **Group Replication:** This release adds the `group_replication_communication_debug_options` server system variable, making it possible to filter out debugging and tracing messages dynamically, per Group Replication component. (WL #10200)

- **Group Replication:** Group Replication thread states are now shown in the MySQL Performance Schema. (WL #10622)

- **JSON:** The `JSON_MERGE()` function is renamed to `JSON_MERGE_PRESERVE()`.

  This release also adds the `JSON_MERGE_PATCH()` function, an RFC 7396 compliant version of `JSON_MERGE_PRESERVE()`; its behavior is the same as that of `JSON_MERGE_PRESERVE()`, with the following two exceptions:

  - `JSON_MERGE_PATCH()` removes any member in the first object with a matching key in the second object, provided that the value associated with the key in the second object is not JSON `null`.

  - If the second object has a member with a key matching a member in the first object, `JSON_MERGE_PATCH()` *replaces* the value in the first object with the value in the second object, whereas `JSON_MERGE_PRESERVE()` *appends* the second value to the first value.

  This example compares the results of merging the same 3 JSON objects, each having a matching key `"a"`, with each of these functions:

```
mysql> SET @x = '{ "a": 1, "b": 2 }',
    >     @y = '{ "a": 3, "c": 4 }',
    >     @z = '{ "a": 5, "d": 6 }';

mysql> SELECT  JSON_MERGE_PATCH(@x, @y, @z)    AS Patch,
    ->         JSON_MERGE_PRESERVE(@x, @y, @z) AS Preserve\G
*************************** 1. row ***************************
   Patch: {"a": 5, "b": 2, "c": 4, "d": 6}
Preserve: {"a": [1, 3, 5], "b": 2, "c": 4, "d": 6}
```

  `JSON_MERGE()` is still supported as an alias of `JSON_MERGE_PRESERVE()`, but is now deprecated and subject to removal in a future MySQL version.

  See Functions That Modify JSON Values, for more information. (Bug #81283, Bug #23255346, WL #9692)

- `InnoDB` now uses Variance-Aware Transaction Scheduling (VATS) for scheduling the release of transaction locks when the system is highly loaded, which helps reduce lock sys wait mutex contention. Lock scheduling uses VATS when >= 32 threads are suspended in the lock wait queue.

  For more information about VATS, see Identifying the Major Sources of Variance in Transaction Latencies: Towards More Predictable Databases.

  Thanks to Jiamin Huang for the contribution. (Bug #25290971, Bug #84266, WL #10793)

- MySQL now extends metadata locks, as necessary, to tables that are related by a foreign key constraint. Extending metadata locks prevents conflicting DML and DDL operations from executing concurrently on related tables. This feature also enables updates to foreign key metadata when a parent table is

modified. Previously, foreign key metadata, which is owned by the child table, could not be updated safely.

If a table is locked explicitly with `LOCK TABLES`, any tables related by a foreign key constraint are now opened and locked implicitly. For foreign key checks, a shared read-only lock (`LOCK TABLES READ`) is taken on related tables. For cascading updates, a shared-nothing write lock (`LOCK TABLES WRITE`) is taken on related tables that are involved in the operation.

If `LOCK TABLES` is active for a table in a foreign key relationship, `ALTER TABLE ... RENAME` is not permitted for that table. This is a temporary restriction, lifted in MySQL 8.0.4 by the patch for Bug #26647340. (WL #6049)

- The `expire_logs_days` system variable, which specifies the binary log expiration period in days, is now deprecated and will be removed in a future MySQL version. `expire_logs_days` does not provide sufficient flexibility for defining the binary log expiration period.

  `binlog_expire_logs_seconds` can be used to set the binary log expiration period in seconds. In MySQL 8.0.3, the effects of the two variables are currently cumulative. For example, if `expire_logs_days` is 1 and `binlog_expire_logs_seconds` is 43200, then the binary log is purged every 1.5 days. This produces the same result as setting `binlog_expire_logs_seconds` to 129600 and `expire_logs_days` to 0. Note that the default `expire_logs_days` setting of 30 days is currently added to the binary log expiration period if `expire_logs_days` is not specified. To use `binlog_expire_logs_seconds` alone, set `expire_logs_days=0` explicitly.

  To disable automatic purging of the binary log, you must set both `expire_logs_days` and `binlog_expire_logs_seconds` explicitly to 0. (WL #10924)

  References: See also: Bug #26483363.

- A new type of backup lock permits DML during an online backup while preventing operations that could result in an inconsistent snapshot. The new backup lock is supported by `LOCK INSTANCE FOR BACKUP` and `UNLOCK INSTANCE` syntax. The `BACKUP_ADMIN` privilege is required to use these statements. (WL #9451)

## Bugs Fixed

- **Incompatible Change; JSON:** If a JSON object contained multiple members with the same key name, MySQL kept the first member and discarded the remainder. This contradicts RFC 7159, which suggests that duplicate key names can be handled in one of the ways listed here:

  - Report an error (or otherwise fail to parse the object)
  - Report all of the name-value pairs, including duplicates
  - Report the last name-value pair only

  When a JSON text is evaluated in JavaScript, the last name-value pair is kept if multiple pairs with the same name are specified. MySQL now does likewise, and implements the last of the three options just listed, as shown here:

```
mysql> CREATE TABLE t1 (c1 JSON);

mysql> INSERT INTO t1 VALUES ('{"x": 17, "x": "red", "x": [3, 5, 7]}');

mysql> SELECT c1 FROM t1;
+------------------+
| c1               |
+------------------+
```

```
| {"x": [3, 5, 7]} |
+------------------+
```

The fix for this issue also corrects a failure in the MySQL 8.0 server to handle insertion into a JSON column of data containing JSON arrays as the values for multiple identical keys. (Bug #86620, Bug #86866, Bug #26238736, Bug #26369555)

- **Performance; JSON:** Creating a representation of a JSON string now optimizes for the most common case—that the string to be processed contains no special characters that need to be escaped—scanning for the first special character in the string, and copying each sequence of characters which do not require escaping in a single `memcpy()` call, rather than checking each character in turn to determine whether it needed to be escaped, escaping it if so, and then copying it, one by one, as was done previously.

  This fix also corrects a failure to escape the control character `\u001f`, or unit separator character. (Bug #86898, Bug #26388690, Bug #87722, Bug #26780307)

  References: See also: Bug #25977595.

- **InnoDB:** The default value for `ON UPDATE` and `ON DELETE` foreign key clauses was changed from `RESTRICT` to `NO ACTION`. `NO ACTION` is a standard SQL keyword that is equivalent to the `RESTRICT` keyword in MySQL. (Bug #30186407, Bug #96466)

- **InnoDB:** Foreign keys records were missing from the `INFORMATION_SCHEMA.INNODB_SYS_FOREIGN` table due to file name encoding of the internal parent and child table names, which caused the names to exceed the permitted length. (Bug #27020089)

- **InnoDB:** A long semaphore wait occurred when executing `ALTER TABLE`, `DROP TABLE`, and `DROP DATABASE` operations. (Bug #26779650)

- **InnoDB:** Invalid error handling code was removed from a function related to tablespace import. (Bug #26595476)

- **InnoDB:** File-per-table tablespaces created prior to MySQL 5.6 caused a failure during an in-place upgrade to MySQL 8.0.2. The tablespaces were not registered with the `InnoDB SYS_TABLESPACES` system table, as required.

  Tables with decimal columns created prior to MySQL 5.5 also caused a failure during an in-place upgrade to MySQL 8.0.2, due to a precision type mismatch. (Bug #26542296, Bug #87229)

- **InnoDB:** During MySQL installation, an unnecessary warning was reported about creation of foreign key constraint system tables. (Bug #26483335)

- **InnoDB:** A query was interrupted during concurrent `ALTER TABLE` operations due to a secondary index entry count mismatch. (Bug #26381213)

- **InnoDB:** A segmentation fault occurred when attempting to open a table that was altered while strict mode was disabled to include conflicting `TABLESPACE` and `COMPRESSION` attributes. (Bug #26375851)

- **InnoDB:** A segmentation fault occurred during a DML operation that used the TempTable storage engine. (Bug #26363837)

- **InnoDB:** Implicit row format conversion during an `ALTER TABLE ... REORGANIZE PARTITION` operation raised an invalid assertion. (Bug #26326611)

- **InnoDB:** An `ALTER TABLE` operation that rebuilt an encrypted table did not set the encryption attribute properly. (Bug #26243264)

- **InnoDB:** A memory leak was encountered on Windows when using the `TempTable` storage engine for in-memory internal temporary tables. (Bug #26237680)

- **InnoDB:** Misleading errors were produced when running Valgrind tests on a server build that was not enabled for Valgrind testing. (Bug #26037206)
- **InnoDB:** Problematic code related to dropping orphan full-text search tables caused an invalid object ID assertion failure on startup. (Bug #25998362)
- **InnoDB:** A check for discarded partitions during a DML operation only checked the first partition. Failure to check for other discarded partitions caused an assertion failure. (Bug #25942592)
- **InnoDB:** Allocated memory was not initialized before it was written to a file, resulting in a Valgrind error. (Bug #25913151, Bug #85986)
- **InnoDB:** Adding a virtual column and index in the same statement caused an error. (Bug #25899959)
- **InnoDB:** Replication lag occurred on slave instances during large update operations on tables with many partitions. (Bug #25687813, Bug #85352)
- **InnoDB:** A failure occurred during an end range comparison. (Bug #25669686)
- **InnoDB:** Enabling the `innodb_buffer_pool_load_now` setting failed in read-only mode. The event that signals the buffer pool load thread was not initialized. (Bug #25586766)
- **InnoDB:** A cursor position check by a multiversion concurrency control row search function raised an assertion. (Bug #25377592)
- **InnoDB:** The wrong variable was passed to the `row_mysql_handle_errors` routine causing an assertion failure. (Bug #25183130)
- **InnoDB:** A long wait for a dictionary operation lock held by a full-text search synchronization operation caused a server exit. (Bug #24938374, Bug #26376681, Bug #26376239)
- **InnoDB:** Assertion code was modified to account for the possibility of a transaction attempting to acquire an explicit lock on a record while another transaction converts an implicit lock to an explicit lock on the same record prior to a commit operation. (Bug #24344131)
- **InnoDB:** A `FLUSH TABLES ... FOR EXPORT` operation on an encrypted or page-compressed table raised an assertion. (Bug #22916982)
- **InnoDB:** A `DROP TABLE` operation was not permitted with an `innodb_force_recovery` setting greater than 0. (Bug #22392152)
- **InnoDB:** A `CREATE TABLE ... SELECT` operation raised an assertion failure when the newly created table was dropped before the transaction was committed. (Bug #22154768)
- **InnoDB:** An in-place `ALTER TABLE` operation that rebuilt the table and added a foreign key without specifying a foreign key constraint name failed due to a duplicate constraint name. `InnoDB` did not account for existing foreign key constraint names. This issue was addressed in MySQL 8.0.3, when foreign key metadata was moved to the data dictionary, and foreign key constraint name generation was moved to the SQL layer. (Bug #18199504, Bug #71616)
- **InnoDB:** A misplaced function call that locks the `InnoDB` data dictionary during a foreign key check was removed. (Bug #12917178, Bug #62221)
- **InnoDB:** The error messages reported when attempting to use an existing foreign key constraint name were inconsistent and did not always provide sufficient information. (Bug #11925430, Bug #60633)
- **Partitioning:** In certain cases when fetching heap records a partition ID could be set to zero. (Bug #86255, Bug #26034430)

- **Partitioning:** It was possible for a `CREATE TABLE` statement that failed to create a partitioned `InnoDB` table not to be rolled back correctly. This was due to an extraneous commit made while performing a check of foreign key information. Since partitioned tables do not presently support foreign keys, this check is unnecessary, and so is no longer made in such cases. (Bug #85299, Bug #25667278)

- **Partitioning:** Queries involving `NULL` were not always handled correctly on tables that were partitioned by `LIST`. (Bug #76418, Bug #20748521)

  References: See also: Bug #86255, Bug #26034430.

- **Replication; Group Replication:** On a member which had both Group Replication and asynchronous replication running simultaneously, asynchronous replication was not respecting restrictions imposed by Group Replication, such as using only the `InnoDB` storage engine, tables requiring primary keys, and so on. This could also be encountered when running `mysqlbinlog` against the member. Now, members running Group Replication and asynchronous replication do not allow any Group Replication requirements to be broken. (Bug #85164, Bug #25609945)

  References: See also: Bug #85781, Bug #25828806.

- **Replication:** On a multithreaded slave, it was possible for a deadlock state to occur due to the timing of updates to the record of disk space used by the relay log. The timing of the update has now been changed so that the deadlock cannot occur. (Bug #26729635)

- **Replication:** With `slave_preserve_commit_order=1` set, a deadlock could occur between a transaction holding a shared write lock on a table, and a transaction earlier in the commit order that also required a shared write lock. (Bug #26666609)

- **Replication:** The unused variable `opt_reckless_slave` was removed. (Bug #26500285)

- **Replication:** `FLUSH LOGS` attempted to send an OK message after having already sent an error response during the commit phase. Thanks to Laurynas Biveinis for the patch. (Bug #26272158, Bug #25363745, Bug #84437)

- **Replication:** With GTIDs generated for incident log events, MySQL error code 1590 (ER_SLAVE_INCIDENT) could not be skipped using the `--slave-skip-errors=1590` startup option on a replication slave. (Bug #26266758)

- **Replication:** `COUNT_TRANSACTIONS_REMOTE_IN_APPLIER_QUEUE` was set to an incorrect value when `group_replication_recovery_complete_at="transactions_certified"` on a recovering member. (Bug #26180350)

- **Replication:** When replicating a partitioned table with an index, on a replication slave where `HASH_SCAN` was specified as part of the `slave_rows_search_algorithms` setting, the slave I/O thread sometimes stopped with an error `HA_ERR_KEY_NOT_FOUND`. (Bug #26137159)

- **Replication:** The system variable `pseudo_slave_mode`, which is for internal server use, sometimes raised an assertion when it was changed inside a transaction. The server no longer changes this variable inside a transaction. (Bug #26034192, Bug #86250)

- **Replication:** When write sets are used for parallelization by a replication slave (as specified by the `binlog_transaction_dependency_tracking` system variable), empty transactions are now ignored, and the handling of relay log rotation has been optimized. (Bug #25982097)

- **Replication:** The Performance Schema `replication_applier_status_by_worker` table sometimes incorrectly displayed a value for APPLYING_TRANSACTION for an inactive worker, because the table was being populated before the worker thread stopped. (Bug #25896166, Bug #85951)

- **Replication:** Attempting to uninstall the plugin while `START GROUP_REPLICATION` executed could result in unexpected behavior. (Bug #25423650, Bug #91042, Bug #28088177)
- **Replication:** In case of a failure while creating multiple slave applier worker threads, some threads would be left orphaned and their resources would not be collected. Thanks to Laurynas Biveinis for his contribution to fixing this bug. (Bug #24679056, Bug #82980)
- **Replication:** The `binlog_checksum` option cannot be changed within a transaction. MySQL cannot log this statement, as would be required inside a transaction, while the requested function is being performed on the binary log. (Bug #22914463)
- **Replication:** `group_replication_force_members` could be used in situations where the group was working properly, in other words a majority was reachable. This incorrect use could cause instability in the group. Therefore, its use has been restricted to the scenario for which it was created, for forming a new membership from a subset of a previous group's membership when a majority of the members are unreachable. (Bug #86359, Bug #26093967)
- **Replication:** Joining a member running a lower version to a group running a higher version resulted in the members running the higher version becoming unreachable. (Bug #85026, Bug #25568493)
- **Group Replication:** Group Replication flow control variables now correctly permit you to have some members in a group that do not affect the minimum throughput of the flow-control mechanism, effectively ignoring those members in case they become blocked. (Bug #26537497)
- **Group Replication:** When the Group Replication plugin delayed initialization thread failed to start due to unavailable resources, a locked mutex was left behind. Now we ensure that this mutex is unlocked when the thread does not start. (Bug #26394678)
- **Group Replication:** In the case of delayed initialization of the Group Replication plugin deployed in single-primary mode, secondaries were able to get writes through an asynchronous replication channel, which is not allowed in normal initialization of the Group Replication plugin. (Bug #26314756)
- **Group Replication:** If the options file contained Group Replication related settings the server could stop unexpectedly on startup. (Bug #26314472)
- **Group Replication:** The values for `group_replication_recovery_use_ssl` and `group_replication_recovery_ssl_verify_server_cert` were not updated when configured for the Group Replication recovery channel. (Bug #26142801)
- **Group Replication:** It was possible to set `server_uuid` to the same value as `group_replication_group_name`, which resulted in unexpected behavior because GTIDs are identified by UUIDs. To fix this problem, setting `server_uuid` to the same value as `group_replication_group_name` is no longer allowed. (Bug #26035931)
- **Group Replication:** It was possible to start the server with invalid values for Group Replication flow control options. Now, the `--group-replication-flow-control-min-quota`, `--group-replication-flow-control-max-quota`, and `--group-replication-flow-control-min-recovery-quota` options are validated on server startup. (Bug #87206, Bug #26531899)
- **Group Replication:** If hostname resolution did not work for a Group Replication member, a misleading error returned when attempting to connect referred to credentials rather than the actual problem with hostname resolution. (Bug #86858, Bug #26368004)
- **Group Replication:** The Group Replication plugin no longer sets `auto_increment_increment` and `auto_increment_offset` when single primary mode is active. (Bug #86669, Bug #26263155)
- **Group Replication:** Group Replication partition threads were not visible in the Performance Schema. (Bug #86626, Bug #26241008)

- **JSON:** Containers in the internal representations of JSON objects and arrays (`Json_object` and `Json_array`) have been changed to use smart pointers rather than raw pointers to `Json_dom`, so that orphaned DOM objects are now automatically destroyed. (Bug #26161264)
- **JSON:** ASCII character 31 (`\u001f`, the unit separator) in a string literal within a JSON document was not quoted when the JSON document was formatted as a string (for example, by `CAST('"\u001f"' AS JSON)`). (Bug #25977959)
- **JSON:** When a *`path_expression`* identified a nonarray value, the `JSON_INSERT()` and `JSON_ARRAY_INSERT()` functions failed to evaluate *`path_expression`*[0] as equal to *`path_expression`*. (Bug #86213, Bug #26022576)
- **JSON:** Searches with `JSON_EXTRACT()` that used wildcards took an inordinate amount of time. (Bug #84523, Bug #25418534)

  References: See also: Bug #83959, Bug #25151440.
- Under heavy load, an infinite loop occurred in Performance Schema buffer container code. (Bug #26666274)
- MySQL-specific typedefs such as `uchar` and `my_bool` were inadvertently reintroduced into the client namespace if the `mysql.h` header file was included. (Bug #26588846, Bug #26582752, Bug #87337)
- `uint8korr()` and related macros were fixed so that they explicitly do unaligned accesses, even on x86. (Bug #26568748, Bug #87298)
- The `main.mysql_upgrade_grant`, `main.roles-upgrade`, and `auth_sec.secure_file_priv_warnings`, test cases mishandled the error log. The `sys_vars.innodb_redo_log_encrypt_basic` test case output was unstable. Thanks to Laurynas Biveinis for the patches. (Bug #26562401, Bug #87279, Bug #26575150, Bug #87313, Bug #26575142, Bug #87314, Bug #26582158, Bug #87303)
- For debug builds, with `sql_buffer_result` enabled, recursive common table expressions caused a server exit. (Bug #26556025)
- Incorrect resolution of a window function as a constant function could result in a server exit. (Bug #26500442)
- For window functions, use of `GROUP BY ... WITH ROLLUP` could cause a server exit. (Bug #26497353, Bug #26497247)
- For window functions, `JSON` columns raised an assertion. (Bug #26496733)
- For window functions, a value less than the argument for a `RANGE` frame caused a server exit. (Bug #26496645)
- Compiling with `-DWITHOUT_SERVER=1` resulted in `my_symlink.c` compilation failure due to missing `#include` for `my_dir.h`. Thanks to Christian Hesse for the patch. (Bug #26495816, Bug #87137)
- Compiling with `-DWITH_SSL=system` `-DWITH_ZLIB=system` assumed that the system `openssl` `zlib` command was available, which might not be the case. Now availability of that command is checked, and if unavailable, the `zlib_decompress` utility is built. (Bug #26494495, Bug #87123)
- yaSSL could incorrectly perform TLS cipher negotiation. (Bug #26482173)
- Some `thread_stack` settings could result in a server exit. (Bug #26438067)
- For window functions, a `ROW` frame accepted noninteger arguments for the row count. (Bug #26411055, Bug #86990)

- `REPLACE(UUID(),...)` expressions could be cached (improperly) and return the same value for each row of a result set. (Bug #26395601)
- When building MySQL within the source tree, `make install` installed some `CMake` files into the `mysql-test` directory within the tree. (Bug #26385175, Bug #86905)
- The `PROCESS_ID` column in the Performance Schema `session_connect_attrs` and `session_account_connect_attrs` tables was changed from `INT` to `BIGINT UNSIGNED` to accommodate larger process ID values. Thanks to Daniël van Eeden for the patch. (Bug #26357806, Bug #86835)
- The index on a generated column, whose value was generated from `JSON_EXTRACT()`, was sometimes not used by the optimizer when it should have been. (Bug #26352119)
- Setting the `log_error_services` system variable to `NULL` caused a server exit. (Bug #26331795)
- For `tar` file packages, some test suite shared libraries were installed in the server package rather than the test package. (Bug #26329850)
- `SHOW COLUMNS` for a valid view could fail. (Bug #26322203, Bug #86778)
- An operation that caused renaming or removal of histogram statistics could cause a server exit. (Bug #26303972)
- For a `VARCHAR` column, sorting using an explicit collation (`ORDER BY col_name COLLATE collation_name`) was much slower than with an implicit collation (no `COLLATE` clause), even if the explicit collation was the same as the implicit collation. (Bug #26286790, Bug #86710)
- `SET binlog_format = ROW` produced a syntax error because `ROW` is now a reserved word. This syntax is now recognized specially to preserve backward compatibility. (Bug #26269280)
- `SET PERSIST_ONLY`, should be permitted only to users who have the `SYSTEM_VARIABLES_ADMIN` and `PERSIST_RO_VARIABLES_ADMIN` privileges, but was incorrectly also permitted to users with the `SUPER` privilege. (Bug #26247864)
- The `information_schema_stats` configuration option, introduced in MySQL 8.0.0, was removed and replaced by `information_schema_stats_expiry`.

  `information_schema_stats_expiry` defines an expiration setting for cached `INFORMATION_SCHEMA` table statistics. For more information, see Optimizing INFORMATION_SCHEMA Queries.

  The `TABLES_DYNAMIC` and `STATISTICS_DYNAMIC` internal system views were removed. (Bug #26203731, Bug #83957)
- Source packages for Debian platforms contained prebuilt debug binaries, causing build failures on any architectures other than the one on which those binaries were built. (Bug #26186911)
- The loadable function registration service did not work if used during server startup. (Bug #26173244)
- Disabling table instrumentation by changes to the Performance Schema `setup_objects` table could cause incorrect index names in index statistics. (Bug #26162562)
- A misleading error message was returned when attempting to drop a nonexistent tablespace file. (Bug #26133507, Bug #86438)
- When running `mysqlbinlog` with the `--read-from-remote-server` option, rewrite rules specified using the `--rewrite-db` option were ignored, so data was not written to the target database. (Bug #26117735, Bug #86288)

- Timestamp data copied from the data dictionary cache during a DDL operation was converted using a `time_zone` value that was no longer valid. The resulting timestamp data was incorrect, causing an error in release builds and an assertion failure in debug builds. (Bug #26091333, Bug #86290)
- Successful data dictionary updates but failure to write the binary log event could result in an inconsistent state. (Bug #26037355)
- `mysqlbinlog` now prints the full metadata for the event type `Table_map_log_event`. (Bug #26020990)
- Some string functions in an `ALTER EVENT` statement could cause a server exit. (Bug #25942505)
- Uninstalling the `daemon_memcached` plugin caused a serious error. (Bug #25909540)
- Prepared statements that used a common table expression and many `?` parameters could be slow. (Bug #25903274, Bug #85933)
- For `UPDATE` or `DELETE` statements with an `ORDER BY ... LIMIT` clause, the optimizer sometimes failed to identify a cheaper ordering method than `filesort`. (Bug #25899921)
- The `rpl_diff.inc` test case file did not find the data difference between servers. Thanks to Yura Sorokin for the patch. (Bug #25860138, Bug #85838)
- An ngram fulltext parser search query returned incorrect results and raised an assertion. (Bug #25851975)
- The combination of an index virtual generated column, a foreign key, and a trigger could cause an assertion to be raised. (Bug #25817660, Bug #85757)
- Selecting from a view could yield different results with materialization enabled versus materialization disabled. (Bug #25782811, Bug #85622)
- After using `SET PERSIST` to set the `event_scheduler` system variable, the server failed to restart if started with the `--skip-grant-tables` option. (Bug #25776940)
- For debug builds, adding an index to a table that had a foreign key relationship could raise an assertion. (Bug #25739983)
- An assertion could be raised for `MIN()`/`MAX()` access to system tables. (Bug #25738624)
- The `INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS` table requested foreign key information from the `InnoDB` storage engine instead of the data dictionary. (Bug #25730513)
- A failed `DROP VIEW` could be written to the binary log. (Bug #25680097)
- `SHOW CREATE VIEW` sometimes added a database name prefix to table names that was not present in the original view definition. (Bug #25634576, Bug #85176)
- The Performance Schema `variables_info` table displayed incorrect `VARIABLE_SOURCE` and `VARIABLE_PATH` values for variables set within option files specified by `!include` or `!includedir` directives. (Bug #25563891)
- Constant string propagation could fail for UCA-based collations. (Bug #25503965, Bug #84837)
- `mysqlpump` displayed incorrect progress information about the number of tables dumped. (Bug #25432850)
- Calculations for UCA 9.0.0 collations were inefficient for tailoring rules containing contraction characters. (Bug #25426632, Bug #84577, Bug #25426632, Bug #84577)

- `GROUP BY DESC` on `DECIMAL` values could incorrectly group `NULL` with non-`NULL` values. (Bug #25407964, Bug #84537)
- Some `mysqldump` warnings went to the standard output rather than the standard error output and consequently were written to the dump file. (Bug #25380000, Bug #82992)
- `NULL` values generated as a result of `WITH ROLLUP` were replaced with the previous row's value when executing a prepared statement on a view having the `GROUP BY .. WITH ROLLUP` clause. (Bug #25174118)
- A server error occurred when a full text search result exceeded the `innodb_ft_result_cache_limit` setting. The patch for this bug also backports a related patch (Bug #21140111). (Bug #25033538)
- A parser refactoring in MySQL 8.0.1 resulted in incorrect handling of some `INSERT ... ON DUPLICATE KEY UPDATE` statements. These problems have been corrected. (Bug #24716127, Bug #25526439, Bug #25071305)
- For debug builds, a `CREATE TABLE` statement with a `VARBINARY` or `BINARY` column having a default value in hexadecimal format caused a server exit. (Bug #24679166, Bug #83020)
- If a stored function was considered a constant by the optimizer, calling it from a subquery in a `NOT IN` condition in the `WHERE` clause could cause a server exit. (Bug #23577867)
- A `mysqldump` memory leak was fixed. Thanks to Yura Sorokin for the patch. (Bug #23531150, Bug #81714)
- Incorrect results or a server exit could result when a query used Batched Key Access optimization and a virtual generated column was part of the join buffer. (Bug #23169112)
- If a session rolled back to a savepoint and then was killed, the statements up to the point of the savepoint could be committed. (Bug #22350047, Bug #79596)
- MySQL accepted a reference to an alias of an aggregated expression defined in an outer query block even when the reference occurred within a GROUP BY subquery where the reference was meaningless. (Bug #21974346, Bug #78785)
- For clients that used Connector/Python and authenticated using the `sha256_password` plugin, the server could handle connections incorrectly. (Bug #21421642)
- Some `SELECT DISTINCT` queries with `GROUP BY` could return incorrect results. (Bug #20692219, Bug #76283)

# Changes in MySQL 8.0.2 (2017-07-17, Development Milestone)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

**Note**

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward. (Making a backup before the upgrade is a prudent precaution in any case.)

- Account Management Notes
- Character Set Support
- Compilation Notes
- Component Notes
- Configuration Notes
- Data Dictionary Notes
- Deprecation and Removal Notes
- SQL Function and Operator Notes
- Logging Notes
- Optimizer Notes
- Packaging Notes
- Parser Notes
- Performance Schema Notes
- Plugin Notes
- Security Notes
- Spatial Data Support
- Test Suite Notes
- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed


## Account Management Notes

- During data directory initialization or upgrade, MySQL now creates a `'mysql.session'@'localhost'` reserved account. This account is used internally by plugins to access the server. It is locked so that it cannot be used for client connections. (Bug #25642343)
- These system variables now are available to define mandatory roles and to enable granted roles to be automatically activated at client connection time:
  - `mandatory_roles` takes a value listing roles the server should treat as automatically granted to all users.
  - `activate_all_roles_on_login` enables control over automatic activation of all granted roles when users log in to the server.

  For more information, see Using Roles. (WL #924)

## Character Set Support

- For Unicode data that uses `NO PAD` collations, sorting of multibyte and variable-length values has been improved:

- `NO PAD` collations are those based on UCA 9.0.0 and higher, such as `utf8mb4_0900_ai_ci` or `utf8mb4_ja_0900_as_cs`.
- The performance improvement is greatest for key values with short weight strings; that is, strings where the weight strings do not fill their entire permitted length. For a `VARCHAR(10)` column that uses the `utf8mb4_0900_ai_ci` collation, values may take up to 160 bytes. The string `'a'` uses only 2 bytes out of a possible 160 and is more sparse than `'abcdefghij'`. But even `'abcdefghij'` uses only 20 bytes of a possible 160 and is more sparse than a string of, for example, ten copies of U+337F SQUARE CORPORATION (㍿), whose weight string requires the entire 160 bytes available.

(Bug #25750527, Bug #85546, WL #9554)

- MySQL now supports a new collation, `utf8mb4_0900_as_ci`, for the `utf8mb4` Unicode character set. This collation is accent sensitive and case insensitive. It is similar to the default `utf8mb4` collation (`utf8mb4_0900_ai_ci`) except that the default collation is accent insensitive.

  MySQL also now supports a new Japanese collation, `utf8mb4_ja_0900_as_cs_ks`, for the `utf8mb4` Unicode character set. This collation is like `utf8mb4_ja_0900_as_cs` in that it is accent sensitive and case sensitive, but `utf8mb4_ja_0900_as_cs_ks` is also kana sensitive and distinguishes Katakana characters from Hiragana characters. `utf8mb4_ja_0900_as_cs` treats Katakana and Hiragana characters as equal for sorting. Applications that require a Japanese collation but not kana sensitivity may use `utf8mb4_ja_0900_as_cs` for better sort performance. `utf8mb4_ja_0900_as_cs` uses three weight levels for sorting; `utf8mb4_ja_0900_as_cs_ks` uses four.

  `utf8mb4_ja_0900_as_cs_ks` is the first collation to use the `_ks` collation suffix. Japanese collations without this suffix are not kana sensitive.

  For more information, see Unicode Character Sets. (WL #10480, WL #10818)

- These character set changes were made to MySQL client support:
  - These clients use a default character set of `utf8mb4` rather than `latin1`: `mysql`, `mysql_upgrade`, `mysqladmin`, `mysqlcheck`, `mysqldump`, `mysqlimport`, `mysqlpump`, `mysqlslap`, `mysqlshow`, and `mysqltest`. (This change does not affect character set autodetection performed by `mysql`, `mysqladmin`, `mysqlcheck`, `mysqlimport`, and `mysqlshow`, as described at Connection Character Sets and Collations).
  - Client connections created using the `libmysqlclient` library use a default character set of `utf8mb4` rather than `latin1`.
  - The `mysqltest` program supports a `--default-character-set` option for specifying the character set explicitly.

  (WL #10353)

## Compilation Notes

- **Performance:** The strict aliasing optimization for GCC is no longer disabled, which results in a minor improvement for MySQL single-threaded performance. (WL #10344)
- **Linux:** MySQL now compiles on Alpine Linux. (Bug #25945568, Bug #80322)
- **Solaris:** On Solaris, server builds now use `std::atomic`, so they must be linked against `libstatomic`. (Bug #25957991)
- MySQL now compiles using Visual Studio 2017. (Bug #25788406)

- The minimum version of the Boost library for server builds is now 1.64.0. (Bug #25772329, Bug #85593)
- These changes were made with respect to client program development:
  - Client programs should only need to `#include` the `<mysql.h>` header file. In particular, `<my_config.h>` should not be needed, and is no longer installed.
  - The `my_init()` function is no longer included in the list of symbols exported from `libmysqlclient`. It need not be called explicitly by client programs because it is called implicitly by other C API initialization functions.

  (Bug #25732787)
- For the Xcode IDE, header files are added to MySQL project sources so they can be searched. (Bug #25636986)
- Work was done to clean up the source code base, including: Removing unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

## Component Notes

- New services are available to enable components to register and unregister system variables (`component_sys_variable_register`, `component_sys_variable_unregister`) and to register status variables (`status_variable_registration`). (WL #8020)
- Two new services are available to enable components and plugins to register and unregister loadable functions: `mysql_service_udf_registration` and `mysql_service_udf_registration_aggregate` provide registration services for scalar and aggregate loadable functions, respectively. These services enable components and plugins to manage loadable functions for themselves, without the need for `CREATE FUNCTION` and `DROP FUNCTION` statements.

  Loadable functions registered using either these services or `CREATE FUNCTION` are listed in the new Performance Schema `user_defined_functions` table. See The user_defined_functions Table. Such loadable functions are *not* listed in the `mysql.func` system table, so the `user_defined_functions` table is preferable for checking which loadable functions are installed. (WL #8020)
- The `mysql_string` string-manipulation service has been converted from a plugin service to a component service: a set of string service APIs for use by components. The password validation component has been updated to use the revised service implementation. (WL #9503)

## Configuration Notes

- Previously, `LOCAL` capability for `LOAD DATA` operations was enabled by default in some contexts: The client library in MySQL binary distributions was compiled with client-side `LOCAL` capability enabled, and the `local_infile` system variable was enabled on the server side. `LOCAL` capability is now disabled by default in all contexts. Client programs must be configured explicitly to enable `LOCAL`, and the server must be run with `local_infile` enabled. See Security Considerations for LOAD DATA LOCAL. (Bug #24511108)
- `mysqld` startup behavior has been modified as follows. In general, these changes enable more consistent or easier server startup when not using `mysqld_safe`.
  - The server executable determines its own full path name at startup and uses the parent of the directory in which it is located as the default `basedir` value. This in turn enables the server to use that

basedir when searching for server-related information such as the share directory containing error messages.

- If error log output cannot be redirected to the --log-error option value, error output could be lost. This is now less likely.
- If the server is started using the --daemonize option and is not connected to a tty device, a default error logging option of --log-error="" is used in the absence of an explicit logging option, to direct error output to the default log file. Previously, an error could occur under these circumstances.
- The -D and -I options now are synonyms for --daemonize and --initialize, respectively.

(Bug #20398088, Bug #75343, WL #10441)

- MySQL now supports a SET PERSIST_ONLY variant of SET statement syntax, for making configuration changes at runtime that also persist across server restarts. Like SET PERSIST, SET PERSIST_ONLY writes the variable setting to an option file named mysqld-auto.cnf in the data directory. However, unlike PERSIST, PERSIST_ONLY does not modify the runtime global system variable value. This makes PERSIST_ONLY suitable for configuring read-only system variables that only be set can at server startup. For more information, see Using Option Files, and SET Syntax for Variable Assignment.

  Use of SET PERSIST_ONLY requires the new PERSIST_RO_VARIABLES_ADMIN privilege, in addition to the SYSTEM_VARIABLES_ADMIN privilege necessary for persisting system variables. (WL #9787)

## Data Dictionary Notes

- The column_stats system table has been removed and replaced by the column_statistics data dictionary table. (WL #8943)
- The version data dictionary table was renamed to dd_properties.

  Metadata for INFORMATION_SCHEMA tables created by dynamic plugins now is recorded in the data dictionary and visible through the INFORMATION_SCHEMA TABLES table.

  These tables are no longer visible in INFORMATION_SCHEMA: SHOW_STATISTICS, SHOW_STATISTICS_DYNAMIC, STATISTICS_BASE, STATISTICS_DYNAMIC, TABLES_DYNAMIC. User impact is minimal because these tables are for internal use by the server. (WL #9495)
- The InnoDB storage engine now uses the MySQL data dictionary rather than its own storage engine-specific data dictionary. For information about the data dictionary, see MySQL Data Dictionary.

  The following list briefly describes the main implications of this change:

  - Upgrade and downgrade implications:
    - To upgrade from MySQL 5.7 to MySQL 8.0, you must perform the upgrade procedure described at Upgrading MySQL.
    - Downgrading from MySQL 8.0 to MySQL 5.7 is only supported using the logical downgrade method (a mysqldump downgrade). In-place downgrades are not supported.
  - Metadata updates associated with exporting and importing tablespaces using the transportable tablespace feature are now performed on global data dictionary tables instead of InnoDB data dictionary tables.
  - InnoDB in-memory metadata is now instantiated from global data dictionary objects. This metadata was previously read from InnoDB system tables.

- Table options that signify tablespace encryption and transparent page compression are now retrieved from the global data dictionary.
- Data dictionary support was added for `InnoDB` `FULLTEXT` indexes. Auxiliary index table names were changed to lowercase.
- `InnoDB` metadata created or modified during DDL operations is now written to the global data dictionary.

(WL #9461, WL #9537, WL #9534, WL #9531, WL #9530, WL #9525, WL #9357)

- `mysql` system tables and data dictionary tables are now created in a single `InnoDB` tablespace file named `mysql.ibd` in the MySQL data directory. Previously, these tables were created in individual `InnoDB` tablespace files in the `mysql` database directory. Associated changes include:
  - The `InnoDB` data dictionary buffer table, which stores fast changing `InnoDB` metadata, was renamed to `innodb_dynamic_metadata` and moved from the `InnoDB` system tablespace to the data dictionary tablespace.
  - Undo tablespace metadata now resides in the data dictionary tablespace.
  - Temporary tablespace metadata now resides in the data dictionary tablespace.
  - Table definitions for tables created in the `mysql` tablespace, the `InnoDB` system tablespace (`innodb_system`), or general tablespaces now include a `TABLESPACE` attribute. This change has the following implications:
    - You cannot use `CREATE TEMPORY TABLE ... LIKE` to create an empty table based on the definition of a table that resides in any of the aforementioned tablespaces, as these tablespaces do not support temporary tables.
    - `CREATE TABLE ... LIKE` preserves the `TABLESPACE` attribute of the original table and creates a new table in the defined tablespace regardless of the `innodb_file_per_table` setting. This is a temporary regression.

    For more information and workarounds, see CREATE TABLE ... LIKE Statement, and CREATE TEMPORARY TABLE Statement.

  (WL #9532)

## Deprecation and Removal Notes

- The `mysql` client by default strips comments in statements sent to the server, and this behavior is controlled using `--skip-comments` (strip comments), and `--comments` (preserve comments).

  Comment stripping is now deprecated. This feature and the options to control it will be removed in a future MySQL version. (WL #10944)

- The `explicit_defaults_for_timestamp` system variable is now enabled by default (previously disabled by default), and a warning occurs if you disable it. This means that the nonstandard (and deprecated) behaviors for default values and `NULL`-value handling in `TIMESTAMP` columns are now disabled by default. (WL #9687)

- Support for these deprecated syntax constructs for table and column references has been removed and their use now results in an error. Instances of these constructs should be changed to remove the leading period.

- *.col_name*
- *.tbl_name*
- *.tbl_name.col_name*

(WL #8662)

- Symbolic link support as described at Using Symbolic Links for MyISAM Tables on Unix, along with the `--symbolic-links` option that controls it, is now deprecated and will be removed in a future MySQL version. In addition, the option is now disabled by default. The related `have_symlink` system variable also is deprecated and will be removed in a future MySQL version. (WL #8392)

## SQL Function and Operator Notes

- MySQL now supports window functions that, for each row from a query, perform a calculation using rows related to that row. These include functions such as `RANK()`, `LAG()`, and `NTILE()`. In addition, most existing aggregate functions now can be used as window functions; for example, `SUM()` and `AVG()`. For more information, see Window Functions.

  **Note**

  Each of the following words now is a reserved word and cannot be used as an identifier without identifier quoting: `CUME_DIST`, `DENSE_RANK`, `FIRST_VALUE`, `GROUPS`, `LAG`, `LAST_VALUE`, `LEAD`, `NTH_VALUE`, `NTILE`, `OVER`, `PERCENT_RANK`, `RANK`, `ROW_NUMBER`, `WINDOW`.

  (WL #9236, WL #9603, WL #9727)

## Logging Notes

- **Incompatible Change:** These error-logging changes have been made:
  - The server is more forgiving if it cannot find the configured error-message file (specified using the `lc_messages_dir` and `lc_messages` system variables). Previously, the server wrote a message to the error log to indicate the problem, then aborted the startup process and exited. Now the server writes a message, but continues startup and defaults to built-in English messages. This applies to messages the server writes to the error log and sends to clients. See Setting the Error Message Language.
  - Error logging was rewritten to use the MySQL component architecture. Traditional error logging is implemented using built-in components, and logging using the system log is implemented as a loadable component. In addition, a loadable JSON-format log sink is available. To control which log components to enable, use the `log_error_services` system variable. For more information, see The Error Log.
  - Incompatibility: To enable logging to the system log, you must load the `log_sink_syseventlog` log component and list it in the `log_error_services` value (see Error Logging to the System Log). This differs from MySQL 5.7 and earlier, for which logging to the system log is enabled by default on Windows, and on all platforms requires no component loading.

    A consequence of this configuration change is that the `log_syslog` system variable previously used to control logging to the system log is obsolete and changes to its value have no effect. `log_syslog` is now deprecated and will be removed in a future MySQL version.

(WL #9323, WL #9342, WL #9343, WL #9344)

## Optimizer Notes

- MySQL now enables management of histogram statistics for table column values:
  - The `ANALYZE TABLE` statement supports `UPDATE HISTOGRAM` and `DROP HISTOGRAM` clauses for generating and removing column histogram statistics.
  - The server stores histogram information in the `column_statistics` data dictionary table. Histograms are viewable using the `INFORMATION_SCHEMA.COLUMN_STATISTICS` table.
  - The `histogram_generation_max_mem_size` system variable controls the amount of memory available for histogram generation.
  - The Performance Schema has a `memory/sql/histograms` instrument for monitoring memory allocations performed for histogram generation.
  - `mysqldump` and `mysqlpump` have a `--column-statistics` option to add `ANALYZE TABLE` statements to the output to generate histogram statistics for dumped tables when the dump file is reloaded.

  For more information, see ANALYZE TABLE Statement, Optimizer Statistics, and The INFORMATION_SCHEMA COLUMN_STATISTICS Table. (WL #8943)

## Packaging Notes

- `mysqladmin` was added to Docker/Minimal packages because it is needed by InnoDB Cluster. (Bug #25998285)
- For Windows, MSI installer packages now include a check for the required Visual Studio redistributable package, and produce a message asking the user to install it if it is missing. (Bug #25658832)
- Debian/Ubuntu packages now support multiple MySQL instances with systemd. See Managing MySQL Server with systemd. (Bug #24559588, Bug #82785)

## Parser Notes

- The parser rules for `ALTER TABLE` were refactored to be context independent and improve maintainability and extensibility. A resulting effect is that some previously accepted undocumented syntax variants are no longer accepted. For example, `CREATE TABLE` statements were permitted with column names qualified by the table name, or by the current database and table name, as were certain `ALTER TABLE` statements for which only column names are permitted. Such statements now produce an error. (WL #8657)

## Performance Schema Notes

- The Performance Schema `threads` table now contains a `RESOURCE_GROUP` column that indicates resource group labels. (WL #8881)
- The Performance Schema now supports versioning, and maintains the current definitions for its tables internally. At startup, the server compares its supported Performance Schema version with the Performance Schema version stored in the data dictionary. If the versions differ, the server drops any old Performance Schema tables and recreates them using the current definitions. In consequence of this change:

  - For MySQL upgrades, it is no longer necessary to run `mysql_upgrade` to incorporate changes to Performance Schema tables because they are recreated automatically as necessary at server startup.

  - The `mysql_system_tables.sql` support script no longer includes SQL statements for Performance Schema table creation because these table definitions are maintained internally.

  To support dynamic Performance Schema table manipulation, a new component service named `pfs_table_service` is now available. (WL #7900, WL #8879)

- The Performance Schema default instrumentation settings have changed:

  - The `transaction` instrument and the `events_transactions_current` and `events_transactions_history` consumers are now enabled by default. See Performance Schema Transaction Tables.

  - All `memory/%` memory instruments are now enabled by default. See Memory Summary Tables.

  - The `wait/lock/metadata/sql/mdl` metadata lock instrument is now enabled by default. See The metadata_locks Table.

  (WL #9625, WL #9628, WL #9629)

## Plugin Notes

- **Incompatible Change:** Plugins such as Group Replication and X Plugin now use the `mysql.session` account added in this release. If you are upgrading from a previous release which did not include the `mysql.session` account, you must run `mysql_upgrade` to ensure the account is created. If `mysql_upgrade` is not run, plugins fail to start with the error message `There was an error when trying to access the server with user: mysql.session@localhost. Make sure the user is present in the server and that mysql_upgrade was run after a server update`. User accounts previously used for this purpose, such as `mysqlxsys`, are no longer created. (Bug #26042764)

  References: See also: Bug #24311527, Bug #25642343, Bug #25750822, Bug #25103980, Bug #83841.

## Security Notes

- The linked OpenSSL library for the MySQL Commercial Server has been updated to version 1.0.2l. Issues fixed in the new OpenSSL version are described at http://www.openssl.org/news/vulnerabilities.html.

  This change does not affect the Oracle-produced MySQL Community build of MySQL Server, which uses the yaSSL library instead. (Bug #26160962)

- Previously, any user could execute the `XA RECOVER` statement to discover the XID values for outstanding prepared XA transactions, possibly leading to commit or rollback of an XA transaction by a user other than the one who started it. Now `XA RECOVER` is permitted only to users who have the new `XA_RECOVER_ADMIN` privilege, which is expected to be granted only to administrative users who have need for it. This might be the case, for example, for administrators of an XA application if it has crashed and it is necessary to find outstanding transactions started by the application so they can be rolled back. This privilege requirement does not affect normal commit or rollback of an XA transaction because the user who started it knows its XID.

For MySQL uprades, `mysql_upgrade` grants `XA_RECOVER_ADMIN` to users who have the `SUPER` privilege, unless some account is found that already has `XA_RECOVER_ADMIN`. (Bug #17188129, WL #7194)

## Spatial Data Support

- Argument suitability checking was improved for these spatial functions that test geometry relationships: `ST_Contains()`, `ST_Crosses()`, `ST_Disjoint()`, `ST_Equals()`, `ST_Intersects()`, `ST_Overlaps()`, `ST_Touches()`, `ST_Within()`, `MBRContains()`, `MBRCoveredBy()`, `MBRCovers()`, `MBRDisjoint()`, `MBREquals()`, `MBRIntersects()`, `MBROverlaps()`, `MBRTouches()`, `MBRWithin()`. For more information, see Spatial Relation Functions That Use Object Shapes, and Spatial Relation Functions That Use Minimum Bounding Rectangles.

  As a consequence of this work, performance of spatial relation functions was improved for cases where both arguments contain geometries of one dimensionality; that is, where the `GeometryCollection` arguments (or one `GeometryCollection` and another type of geometry) passed to a spatial relation function can be reduced to a `MultiPoint`, `MultiLineString`, or `MultiPolygon`. In those cases, the more complex general `GeometryCollection` can be avoided. (Bug #22285402, Bug #79406, WL #8685)

## Test Suite Notes

- The `--do-test-list` option for `mysql-test-run.pl` accepts an argument for a file containing tests one per line. Lines in the file are now accepted in any of the following formats:

  ```
  test_name
  test_name.test
  suite_name.test_name
  suite_name.test_name.test
  path/to/test/file
  ```

  (Bug #25700362)

- `mysql-test-run.pl` and `mysqltest` are now more restrictive about permitted test case and result file names. Names must consist only of alphanumeric characters (`A-Z`, `a-z`, `0-9`), dash (`-`), or underscore (_), and cannot start with dash or underscore. An error occurs for nonconforming file names. (Bug #25487471)

- These changes were made to the `--xml-report` option for `mysql-test-run.pl`:

  - A `<failure>` tag identifies any test failing on a retry attempt.
  - Separate statistics and fields are included for skipped and disabled tests.
  - The XML report is created in the build directory if no absolute path is given for out-of-source builds.
  - For combination runs, a field named `variation` is included in the `<testcase>` tag.
  - Results for all tests belonging to a suite are aggregated within a single `<testsuite>` tag.
  - Information regarding failures is reported with a brief reason in an attribute named `message`, along with other details within the tag.

  (Bug #25349924)

- `mysql-test-run.pl` now supports a `--report-unstable-tests` option that has these effects:

  - Reports any test that has passed using at least one retry attempt in a separate category called “Unstable tests” in the summary.
  - If all failures are due to unstable tests, `mysql-test-run.pl` produces a warning but exits successfully.
  - Adds a new XML tag to report unstable tests, if the `--xml-report` option is also specified.

  (Bug #24473420, Bug #25984429)

- `mysqltest` now supports a `replace_numeric_round` command that takes an argument value from 0 to 16 indicating the number of decimals to round numeric values to. This can be used to help prevent result content mismatch errors for tests in which slightly different results are obtained across platforms due to precision differences. Thanks to Daniel Black for the patch. (Bug #23280117, Bug #81399)
- `mysql-test-run.pl` now looks for a *`testname`*`-client.opt` file, which is analogous to the *`testname`*`-master.opt` file but for specifying test-specific client options. (Bug #17084918)

## X Plugin Notes

- X Plugin now handles expired SSL certificates correctly. (Bug #25835833)
- The output of clauses such as `GROUP BY` and `HAVING` has been improved by moving the `grouping_criteria` expression to the derived query. (Bug #25549637, Bug #24497007)
- During install of the X Plugin a blank rule was being generated in the Firewall white list. (Bug #24488234)

## Functionality Added or Changed

- **Incompatible Change; InnoDB:** `ADD PARTITION`, `DROP PARTITION`, `COALESCE PARTITION`, `REORGANIZE PARTITION`, and `REBUILD PARTITION` `ALTER TABLE` options are now supported by native partitioning in-place APIs and may be used with `ALGORITHM={COPY|INPLACE}` and `LOCK` clauses.

  `EXCHANGE PARTITION`, which previously supported `ALGORITHM` and `LOCK` clauses, is performed by a new server layer API for compatibility with the MySQL data dictionary. Partition and table exchange is performed internally by the native partition handler.

  `DROP PARTITION` with `ALGORITHM=INPLACE` deletes data stored in the partition and drops the partition. However, `DROP PARTITION` with `ALGORITHM=COPY` or `old_alter_table=ON` rebuilds the partitioned table and attempts to move data from the dropped partition to another partition with a compatible `PARTITION ... VALUES` definition. Data that cannot be moved to another partition is deleted. (WL #9559)

- **InnoDB:** `InnoDB` now uses tablespace map files during recovery to identify tablespaces that require redo log application. This is a change from redo log tablespace discovery that was used previously.

  If tablespace map files are lost or corrupted, the `innodb_scan_directories` startup option may be used to specify tablespace directories when starting MySQL after a server outage.

  For more information, see Tablespace Discovery During Crash Recovery. (Bug #24793413, WL #9499)

- **InnoDB:** In previous releases, `InnoDB` stores `ROW_FORMAT` and `KEY_BLOCK_SIZE` attributes specified in `CREATE TABLE` or `ALTER TABLE` statements even if those attributes are silently ignored when creating or altering a table. In such cases, `SHOW CREATE TABLE` and the `Create_options` column

reported by `SHOW TABLE STATUS` show the specified `ROW_FORMAT` and `KEY_BLOCK_SIZE` attributes rather than the actual attributes of the table. `InnoDB` now stores and reports the actual `ROW_FORMAT` and `KEY_BLOCK_SIZE` attributes of the table. (WL #9525)

- **Replication:** In previous versions issuing `STOP GROUP_REPLICATION` stopped the plugin but the server still accepted transactions. This meant the transactions were not transmitted to the group. To make `STOP GROUP_REPLICATION` safer, now `super_read_only` is set to `ON` immediately upon issuing `STOP GROUP_REPLICATION`, which ensures no transactions are accepted. (Bug #25495067, Bug #84795, WL #10611, WL #10960)

- **Replication:** When there was a network partition and a member was in a minority all queries to that member blocked. To improve this situation, the `group_replication_unreachable_majority_timeout` variable has been added which enables you to configure how long members in a minority wait to regain contact with a member in the majority before leaving the group. (Bug #25473794)

- **Replication:** The `group_replication_transaction_size_limit` variable was added to enable you to protect a group against large transactions causing a failure. (Bug #84785, Bug #25510757)

- **Replication:** Support for binary log files created by versions earlier than MySQL 5.0 has been removed, and binary log Version 1 and Version 3 formats are no longer supported by slaves or `mysqlbinlog`. (WL #9219)

- **Replication:** The Performance Schema `replication_group_members` and `replication_group_member_stats` tables have been extended to provide additional monitoring information for Group Replication. `replication_group_members` now includes information about member roles and the version of MySQL running on a given member. `replication_group_member_stats` now includes information about all members in the group, the applier and local queues, and transactions which have been rolled back. (WL #10380)

- **Replication:** The receiver thread has been improved to no longer block other thread's activities when waiting for disk space. This improves the monitoring of replication ensuring that it reports correctly when the receiver thread is waiting for disk space. If you are not able to free disk space to allow the receiver thread to continue its activity, it can be forcefully stopped without side effects in most cases. (WL #10406)

- **Replication:** A new transaction length field has been added to the `Gtid_log_event` which stores the transaction length in bytes. (WL #10493)

- **Replication:** The following variables have had their defaults changed to ensure replication is as robust and efficient as possible by default:

  - `master_info_repository=TABLE` and `relay_log_info_repository=TABLE` ensure that replication repository information is stored in InnoDB tables, making replication resilient to unexpected halts.

  - `transaction_write_set_extraction=XXHASH64` enables the optimal method for generating write set hashes.

  - `slave_rows_search_algorithms='INDEX_SCAN,HASH_SCAN'` enables the use of a hash table to avoid repeated table scans when no Primary Key or Primary Key Equivalent (non-null unique secondary index) exists for a table

  - `expire_logs_days=30` causes unused binary logs that are older than 30 days to be purged, ensuring storage space is used efficiently.

  (WL #10474, WL #10476, WL #10477, WL #10478)

- **Replication:** The Group Replication plugin now notifies other server components that relevant events have occurred; the plugin informs listeners registered in the service registry and notifies them that an event has occurred. These listeners, including other plugins, can then react to these events. Such events include view changes, recovery state updates, network partitioning, and primary election. (WL #10412)

- **Group Replication:** The `group_replication_member_weight` variable has been added which enables you to control the election of new primaries in single-primary mode. In previous versions primary election was based on the member's UUID, with the lowest UUID elected as the new primary in the event of fail over. Use this variable to assign numeric weights to members to ensure that specific members are elected, for example during scheduled maintenance of the primary or to ensure certain hardware is prioritized. (WL #10433, WL #10959)

- **Group Replication:** The following system variables affecting Group Replication have been added in this release:

  - `group_replication_flow_control_hold_percent`
  - `group_replication_flow_control_max_quota`
  - `group_replication_flow_control_member_quota_percent`
  - `group_replication_flow_control_min_quota`
  - `group_replication_flow_control_min_recovery_quota`
  - `group_replication_flow_control_period`
  - `group_replication_flow_control_release_percent`

  These variables enable you to fine-tune flow control of individual group members, adjusting the quota depending on the task the member performs in the group. (WL #9838)

- **JSON:** Added support for ranges in the XPath expressions used with many MySQL JSON functions, including `JSON_EXTRACT()` and `JSON_REMOVE()`. Such a range is specified using the syntax *`start`* `to` *`end`*, where *`start`* and *`end`* are, respectively, the first and last indexes of a range of elements from a JSON array (always numbered starting with 0). For example, `$[1 to 3]` includes the second, third, and fourth elements, as shown here:

```
mysql> SELECT JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[1 to 3]');
+-----------------------------------------------+
| JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[1 to 3]') |
+-----------------------------------------------+
| [2, 3, 4]                                     |
+-----------------------------------------------+
1 row in set (0.00 sec)
```

  This work also provides support in such expressions for the `last` keyword, which you can use to represent the index of the last (rightmost) element in the current array, like this:

```
mysql> SELECT JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last]');
+-----------------------------------------+
| JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last]') |
+-----------------------------------------+
| 5                                       |
+-----------------------------------------+
1 row in set (0.00 sec)
```

  Indexes relative to the end of the array are also supported, as shown here:

```
mysql> SELECT JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last-4 to last-2]');
```

```
+-------------------------------------------------------+
| JSON_EXTRACT('[1, 2, 3, 4, 5]', '$[last-4 to last-2]') |
+-------------------------------------------------------+
| [1, 2, 3]                                             |
+-------------------------------------------------------+
1 row in set (0.00 sec)
```

For further information and examples, see Searching and Modifying JSON Values. (Bug #79052, Bug #22285926, WL #9831)

- **JSON:** The optimizer now supports partial (in-place) updates of JSON documents when using `JSON_SET()`, `JSON_REPLACE()`, or `JSON_REMOVE()`. (Previously, all updates of `JSON` column values were done by completely removing the previous document and writing the new one in its place.) In general, this optimization can be performed when the replacement value is less than or equal in size to the column's original value, and no new elements are added to the original value. It *cannot* be performed for a simple replacement of the column value of the form `UPDATE tbl SET json_col = json_val`.

  The JSON Data Type provides more information about the requirements for an update to be optimized in this way.

  This work includes the addition of two JSON utility functions, `JSON_STORAGE_SIZE()` and `JSON_STORAGE_FREE()`, in the MySQL Server. `JSON_STORAGE_SIZE()` returns the number of bytes used to store the binary representation of a JSON document, whether the document is presented as a column value in a table, as the value of a user variable, or as a JSON literal. In the case of a `JSON` column, this is the space used to store the JSON document as it was inserted into the column, prior to any partial updates that may have been performed on it since then. `JSON_STORAGE_FREE()` shows the number of bytes in the binary representation of a JSON column value that were freed by the most recent partial update of the column. For a user variable storing a JSON document, `JSON_STORAGE_FREE()` always returns 0; it also returns 0 if the argument is a JSON literal.

  Each of these functions, like many other MySQL functions that act on JSON values, also accepts a string that can be successfully parsed as a JSON document. For more information and examples, see JSON Utility Functions. (WL #9192, WL #8963, WL #10570)

- RPM `.spec` files now include support for running unit tests. (Bug #25814143, Bug #85743)

- The `mysql` client now supports a `--binary-as-hex` option that causes display of binary data using hexadecimal notation (`0x`*`value`*). Thanks to Daniël van Eeden for the patch. (Bug #25340722, Bug #84391)

- The `SHOW TABLES` statement now supports an optional `EXTENDED` keyword that causes statement output to list hidden tables created by failed `ALTER TABLE` statements. (These temporary tables have names beginning with `#sql`.) For more information, see SHOW TABLES Statement. (Bug #24786075, Bug #83241)

- Serialized dictionary information (SDI) files now have a lowercase extension (`.sdi`) instead of uppercase. (WL #7069)

- The `TempTable` storage engine replaces the `MEMORY` storage engine as the default engine for in-memory internal temporary tables. The `TempTable` storage engine provides efficient storage for `VARCHAR` and `VARBINARY` columns. The `internal_tmp_mem_storage_engine` session variable defines the storage engine for in-memory internal temporary tables. Permitted values are `TempTable` (the default) and `MEMORY`. The `temptable_max_ram` configuration option defines the maximum amount

of memory that can be occupied by the `TempTable` storage engine before it starts storing data on disk. The default value is 1073741824 bytes (1GiB).

The `memory/temptable/physical_ram` and `memory/temptable/physical_disk` Performance Schema instruments may be used to monitor `TempTable` memory allocation and disk storage. (WL #8117)

- The MySQL 8.0.2 release introduces a number of undo related changes:

  - The number of undo tablespaces can now be modified at runtime, or when the server is restarted, using the `innodb_undo_tablespaces` configuration option. Previously, the number of undo tablespaces could only be configured when initializing the MySQL instance and could not be changed afterward. This change permits the addition of undo tablespaces and rollback segments as the database grows.

  - `innodb_undo_log_truncate` is enabled by default. When enabled, any undo tablespace that exceeds the threshold value defined by `innodb_max_undo_log_size` is marked for truncation. See Truncating Undo Tablespaces.

  - The `innodb_undo_tablespaces` default value was changed from 0 to 2, which means that rollback segments are created in two separate undo tablespaces instead of the `InnoDB` system tablespace by default. A minimum of two undo tablespaces is required to permit truncation of undo logs.

    Setting `innodb_undo_tablespaces` to 0 is deprecated and will not be supported in a future MySQL version.

  - The naming convention used for undo tablespace files is changed from `undo`*`NNN`* to `undo_`*`NNN`*, where *`NNN`* is the undo space number.

  - The `innodb_rollback_segments` configuration option defines the number of rollback segments per undo tablespace. Previously, `innodb_rollback_segments` was a global setting that specified the total number of rollback segments for the MySQL instance. This change increases the number of rollback segments available for concurrent transactions. More rollback segments increases the likelihood that concurrent transactions use separate rollback segments for undo logs, resulting in less resource contention.

  - The `innodb_undo_logs` configuration option is removed. The `innodb_rollback_segments` configuration option performs the same function and should be used instead.

  - The `Innodb_available_undo_logs` status variable is removed. The number of available rollback segments per tablespace may be retrieved using `SHOW VARIABLES LIKE 'innodb_rollback_segments';`

  (WL #9507, WL #10498, WL #10499, WL #10322)

- The `SHOW COLUMNS` and `SHOW INDEX` statements now support an optional `EXTENDED` keyword that causes statement output to include information about hidden columns and indexes that MySQL uses internally and are not accessible by users. For more information, see SHOW COLUMNS Statement, and SHOW INDEX Statement. (WL #9570)

## Bugs Fixed

- **InnoDB:** Queries run on `INFORMATION_SCHEMA.TABLES` performed more slowly with `information_schema_stats` set to `latest`. (Bug #26197113)

- **InnoDB:** An `ALTER TABLE ... ADD PARTITION` operation that specified a `DATA DIRECTORY` clause failed to ignore the `TABLESPACE` attribute of the table. (Bug #26113652)
- **InnoDB:** When `foreign_key_checks` is disabled, a child table with a foreign key constraint can be created before the parent table, which can result in a foreign key constraint failure, as the parent table is unaware of the constraint. When a table is created, there is now a call to load foreign key constraints for the table and check for child tables. (Bug #25976199)
- **InnoDB:** A parsing error occurred while optimizing a table with a full-text index. (Bug #25914332)
- **InnoDB:** Compiling MySQL on Windows with Microsoft Visual C++ 2015 or macOS 10.12.4 with GCC 4.2.1 or Apple LLVM version 8.0.0 (clang-800.0.38) returned warnings. (Bug #25910531)
- **InnoDB:** In debug builds, an assertion was raised during bootstrap when the system tablespace file (`ibdata1`) ran out of space during creation of doublewrite pages. (Bug #25872368)
- **InnoDB:** Incorrect locking order caused a deadlock when `InnoDB` attempted to persist an auto-increment counter value to disk. (Bug #25833228)
- **InnoDB:** Internal methods for accessing dictionary table object data did not account for virtual columns. (Bug #25822154)

  References: This issue is a regression of: Bug #23748128.
- **InnoDB:** The length of a virtual column field in a virtual index record was less than the expected template column length. (Bug #25793677)
- **InnoDB:** In debug builds, shutting down the server with `--innodb-fast-shutdown=0` raised an assertion. (Bug #25756224)
- **InnoDB:** The `ibd2sdi` utility exited when run on an unsupported file type. (Bug #25738491)
- **InnoDB:** `InnoDB` did not set the compression algorithm when opening a partitioned table. (Bug #25685868)
- **InnoDB:** An in-place `ALTER TABLE` operation failed to set the encryption type, causing a `FLUSH TABLES ... FOR EXPORT` operation to assert. (Bug #25672779)
- **InnoDB:** A latch that was held while registering a file close caused a hang condition. (Bug #25658467)
- **InnoDB:** During recovery, prepared transactions were rolled back if the `innodb_force_recovery` setting was greater than 0. (Bug #25651042)
- **InnoDB:** A `CREATE TABLE` operation that defined a unique key with an eight character prefix on a NOT NULL `TEXT` field would raise an assertion if a primary key was not defined. (Bug #25647413)
- **InnoDB:** Updates to data dictionary tables combined with updates to `InnoDB` system tables for full-text search auxiliary tables raised a lock-related assertion. (Bug #25610353)
- **InnoDB:** The server allocated memory unnecessarily for an operation that rebuilt the table. (Bug #25573565, Bug #85043)
- **InnoDB:** Test-related code intended to simulate a random read on a nonexistent page raised an invalid assertion. (Bug #25479538)

  References: This issue is a regression of: Bug #25053705.
- **InnoDB:** With `foreign_key_checks` disabled, `InnoDB` incorrectly printed messages to the error log when operations were performed on a table that referenced a nonexistent foreign key. (Bug #25365223)

- **InnoDB:** During a `TRUNCATE TABLE` operation on a file-per-table tablespace, a dictionary operation lock was released before eviction of dirty pages from the buffer pool, causing a latch order violation. (Bug #25357789)

- **InnoDB:** A `DROP TABLE` or `TRUNCATE TABLE` operation could raise an assertion following a failed online `ALTER TABLE .. ADD INDEX` operation. The index root page was dropped prematurely. (Bug #25357789)

- **InnoDB:** When using an index merge optimizer switch, a `SELECT COUNT(*)` operation sometimes returned 0. Partitioning code incorrectly performed a `memcpy` instead of a column copy of columns read by the index, causing the wrong records to be copied. (Bug #25332330, Bug #81031)

- **InnoDB:** After a `RENAME TABLE` operation that moved a table to a different schema, `InnoDB` returned an error on restart indicating that it could not locate the tablespace data file. `InnoDB` failed to update `INNODB_SYS_DATAFILES` data dictionary table during the `RENAME TABLE` operation. (Bug #25189192, Bug #84038)

- **InnoDB:** During an `ALTER TABLE` operation that rebuilt a table containing a virtual column, `InnoDB` failed to apply a concurrent insert log record. (Bug #24961167)

- **InnoDB:** `InnoDB` failed to apply the concurrent delete log for an in-place `ALTER TABLE` operation due to a virtual column validation issue. (Bug #24960450)

- **Partitioning:** Following execution of an `ALTER TABLE ... OPTIMIZE PARTITION` statement, names of tablespaces used by the table were not preserved, and subsequent accesses to the table eventually caused the server to fail. (Bug #25512556)

- **Replication:** When replicating a partitioned table with an index, on a replication slave where `HASH_SCAN` was specified as part of the `slave_rows_search_algorithms` setting, the slave I/O thread sometimes stopped with an error `HA_ERR_KEY_NOT_FOUND`. (Bug #26137159)

- **Replication:** A `USE` statement that followed a `SET GTID_NEXT` statement sometimes had no effect. (Bug #26128931)

- **Replication:** A misleading warning was issued when the command `FLUSH LOGS` or `PURGE LOGS BEFORE` was used on a binary log file with an expiry time set, and the binary log file was in use. The warning related to the file being in use, and implied that a purge attempt had taken place, even if the expiry time had not yet been reached. Now, MySQL checks the expiry time of the binary log file first, and only then checks whether the file is in use. So the warning is only issued for an in-use binary log file that is old enough to be purged. (Bug #25973525)

- **Replication:** Groups can now contain members running different server versions to enable you to do online upgrades of a replication group. The rules for combining members in a group with different versions are:

  - If you have a group with 8.0 members, you cannot add a 5.7 member

  - If you have a group with 5.7 members you can add a 8.0 member, but it remains in read-only mode. Writing to this member is dangerous while the group contains multiple server versions and should be avoided.

  In a single-primary group, if the current primary leaves the group and a new primary must be elected, the primary is first chosen from the lower version members. If no lower version member is found, the primary is chosen from newer version members. (Bug #25876807)

- **Replication:** Now that XA transactions are prepared and committed in two parts, an issue with statement-based replication has been identified. If two XA transactions committed on the master are

being prepared on the slave in the inverse order, locking dependencies can occur that cannot be safely resolved. The issue is not present with row-based replication.

XA transactions are therefore now considered unsafe for statement-based replication.

- When `binlog_format = STATEMENT`, a warning is issued for DML statements inside XA transactions, and replication might fail with deadlock on slaves.
- When `binlog_format = MIXED`, DML statements inside XA transactions are logged using row-based replication.
- When `binlog_format = ROW`, DML statements inside XA transactions are logged as before.

(Bug #25786490, Bug #85639)

- **Replication:** The following Performance Schema replication tables now continue to be populated when the Performance Schema is disabled:
  - `replication_connection_configuration`
  - `replication_connection_status`
  - `replication_applier_configuration`
  - `replication_applier_status`
  - `replication_applier_status_by_coordinator`
  - `replication_applier_status_by_worker`

  The exception is local timing information (start and end timestamps for transactions) in the replication tables `replication_connection_status`, `replication_applier_status_by_coordinator`, and `replication_applier_status_by_worker`. This information is not collected when the Performance Schema is disabled. (Bug #25694813)

- **Replication:** When first starting the MySQL server following an installation from RPM, passwword validation plugin is activated by default (true only for RPM installations). If binary logging was already enabled at this time, the activation was logged, even though plugin activations should not be recorded in the binary log. (Bug #25672750)

- **Replication:** The `XA START`, `XA END`, `XA COMMIT`, and `XA ROLLBACK` statements, used to perform XA transactions, were incrementing the counter for a `replicate-do-db` filter that named the relevant database. The counter is no longer incremented for these statements. (Bug #25669344)

- **Replication:** MySQL 8.0 servers were not able to join a replication group that contained MySQL 5.7 servers, and MySQL 5.7 servers in a replication group could not be upgraded to MySQL 8.0. (Bug #25477979)

- **Replication:** When `mysqlbinlog` output from a MySQL 5.7 server was applied on a MySQL 8.0 server, the MySQL 8.0 server generated its own timestamp for the `original_commit_timestamp`. The fix ensures that the value of `original_commit_timestamp` is correctly set to 0, meaning that the timestamp is not known. (Bug #25316086)

- **Replication:** If the server was started with the `--log-bin` option, discovery of one Performance Schema table with invalid structure caused all subsequently checked Performance Schema tables to be marked invalid as well. (Bug #25041396)

- **Replication:** Interleaved transactions could sometimes deadlock the slave applier when the transaction isolation level was set to `REPEATABLE-READ`. (Bug #25040331)

- **Replication:** The `slave_skip_errors` system variable did not permit error numbers larger than 3000. Thanks to Tsubasa Tanaka for the patch. (Bug #24748639, Bug #83184)

- **Replication:** The SQL state of `ER_TRANSACTION_ROLLBACK_DURING_COMMIT` was `HY00`, which is generic and not suitable for rollback. The SQL state has been modified to 40000. (Bug #24658431)

- **Replication:** `mysqlbinlog`, if invoked with the `--raw` option, does not flush the output file until the process terminates. But if also invoked with the `--stop-never` option, the process never terminates, thus nothing is ever written to the output file. Now the output is flushed after each event. (Bug #24609402)

- **Replication:** A memory leak in `mysqlbinlog` was fixed. The leak happened when processing fake rotate events, or when using `--raw` and the destination log file could not be created. The leak only occurred when processing events from a remote server. Thanks to Laurynas Biveinis for his contribution to fixing this bug. (Bug #24323288, Bug #82283)

- **Replication:** A slave server could lose events not yet applied when `MASTER_AUTO_POSITION=0`, both replication threads were stopped, and the applier delay was changed using `CHANGE MASTER TO MASTER_DELAY=N`. (Bug #23203678, Bug #81232)

  References: See also: Bug #25340185, Bug #84375.

- **Replication:** Transmission of large GCS messages could take so long the sender appeared to have died. (Bug #22671846)

- **Replication:** Multithreaded slaves could not be configured with small queue sizes using `slave_pending_jobs_size_max` if they ever needed to process transactions larger than that size. Any packet larger than `slave_pending_jobs_size_max` was rejected with the error ER_MTS_EVENT_BIGGER_PENDING_JOBS_SIZE_MAX, even if the packet was smaller than the limit set by `slave_max_allowed_packet`.

  With this fix, `slave_pending_jobs_size_max` becomes a soft limit rather than a hard limit. If the size of a packet exceeds `slave_pending_jobs_size_max` but is less than `slave_max_allowed_packet`, the transaction is held until all the slave workers have empty queues, and then processed. All subsequent transactions are held until the large transaction has been completed. The queue size for slave workers can therefore be limited while still allowing occasional larger transactions. (Bug #21280753, Bug #77406)

- **Replication:** An incident event that broke replication was not written to the binary log with a GTID, so that it was not possible to skip the event using `SET gtid_next=value`. Instead, it was necessary to set the relay log file and relay log positions directly; this meant that, when autopositioning was enabled, it was necessary first to disable it, then to set the relay log file and position, and finally to re-enable autopositioning.

  Now in such cases MySQL writes the incident event into the statement cache, so that a GTID is generated and written for it prior to flushing, and that the slave applier works with the change. Then users can skip the event using the SQL statement `SET gtid_next=value`, followed by `BEGIN` and `COMMIT`. (Bug #19594845)

- **Replication:** Setting an empty filter rule using `CHANGE REPLICATION FILTER` caused issues when running tests with UBSan. (Bug #85405, Bug #25702297)

- **Replication:** When the receiver thread held a lock on the relay log while queuing an event, clients executing `SHOW SLAVE STATUS` or `SHOW RELAYLOG EVENTS` were blocked. (Bug #85084, Bug #25584734)

- **Replication:** Indexes for the Performance Schema `replication_applier_status_by_worker` table worked correctly only for single threaded replication. This was due to the fact that the indexing relied exclusively on the SQL thread ID regardless of the existence of any workers. Now in such cases, the index also considers worker thread ID when multithreading is enabled on the slave. (Bug #84646, Bug #25444610)

- **Replication:** In certain cases, the master could write to the binary log a `last_committed` value which was smaller than it should have been. This could cause the slave to execute in parallel transactions which should not have been, leading to inconsistencies or other errors. (Bug #84471, Bug #25379659)

- **Replication:** When using `group_replication_ip_whitelist=AUTOMATIC`, IPs in the private network are permitted automatically, but some class C IP addresses were not being permitted correctly. (Bug #84329, Bug #25503458)

- **Replication:** The replication applier thread returns Error 3002 `ER_INCONSISTENT_ERROR` when there is a difference between an expected error number and the actual error number. It is now possible to ignore this error by using 3002 with `slave_skip_errors`. (Bug #83186, Bug #24753281)

- **Replication:** MySQL lost its GTID position following a restart when a dump from `mysqldump` had been used to load data.

  To keep this problem from occurring, the `mysql.gtid_executed` table is now excluded automatically from dumps made by `mysqldump`. (Bug #82848, Bug #24590891)

  References: See also: Bug #87455, Bug #26643180.

- **Replication:** It was possible to set `binlog_format` without causing an error when there were open temporary tables. (Bug #82467, Bug #24411680)

- **Replication:** Corruption of relay logs for one channel in multi-source replication caused good channels not to be initalized during a server restart. In addition, when run with `--skip-slave-start=false`, the server also failed to start slave threads for those channels which were in good condition, despite the fact that it should have started the slave threads for all good channels.

  Now, regardless of any errors on other channels, the server attempts to create and initialize channels that are in good condition, and starts slave threads for the good channels if `--skip-slave-start` is disabled. As part of this fix, `START SLAVE` and `STOP SLAVE`, which are intended to operate on all channels, are also modified such that they continue executing on all good channels even if they find bad channels among them. (Bug #82209, Bug #24285104)

- **Replication:** The SQL thread was unable to GTID skip a partial transaction. (Bug #81119, Bug #25800025)

- **Replication:** It was possible for `FLUSH LOGS` to write a `Rotate_log_event` into an uninitialized log file. (Bug #80368, Bug #22732184)

  References: See also: Bug #23531998, Bug #81734.

- **Group Replication:** When `binlog_checksum` was `NONE` following startup, when Group Replication was started and an error occurred, the server remained in the `RECOVERING` state and could not be shut down. (Bug #25793366, Bug #85667)

- **Group Replication:** In a Group Replication setup where circular asynchronous replication was implemented between members of different replication groups, view change log events were repeatedly

replicated between the groups with new generated GTIDs each time. This fix ensures that view change log events are ignored outside the named replication group in which they occur, and never generate new GTIDs. (Bug #25674926)

References: See also: Bug #26049695, Bug #25928854, Bug #25721175.

- **Group Replication:** In a setup where single-primary Group Replication was combined with asynchronous replication, for example with servers `S1` and `S2` forming a group, and `S2` and `S3` functioning as master and slave, secondaries such as `S2` accepted transactions and were then able to enter the group. This fix prevents a secondary from creating an asynchronous replication channel when it belongs to a single-primary group, and Group Replication from starting while asynchronous replication is running. (Bug #25574200, Bug #85047)

  References: See also: Bug #86325, Bug #26078602.

- **Group Replication:** Group Replication failed to start if the `super_read_only` system variable was enabled. (Bug #25481287, Bug #84733)

- **Group Replication:** In the event that a member failed to join a group the member was not stopping and continued to accept transactions. To avoid this set your members to use `super_read_only=1` in the `my.cfg` file. Group Replication now checks for this setting upon successful startup, and sets `super_read_only=0`. This ensures that members which do not successfully join a group cannot accept transactions. (Bug #25474736, Bug #84728)

- **Group Replication:** Loading and initialization of the Group Replication plugin failed if no `root` account was present. (Bug #24311527)

  References: See also: Bug #25750822, Bug #25103980, Bug #83841.

- **Group Replication:** Issuing `SHOW SLAVE STATUS FOR CHANNEL 'group_replication_recovery'` following restart of a server using group replication led to an unplanned shutdown. (Bug #85739, Bug #25813258)

- **Group Replication:** When an existing `GTID_NEXT` transaction was assigned a conflicting GTID by the server, Group Replication generated an assert upon detecting two transactions with same GTID. This was because Group Replication generates the GTID after conflict detection, which is later than with asynchronous replication. When the transaction has been committed, the server issues a warning that the GTID has already been used, and the transaction is rolled back. (Bug #84153, Bug #25232042)

- **Microsoft Windows:** On Windows, the `Docs/INFO_SRC` file was missing the `build-date` entry. (Bug #25799855)

- **JSON:** `NULLIF()` failed with an assertion error in debug builds if the result from this function was used in a JSON context. This result can now be used safely as a JSON value.

  This fix includes using `DBUG_ASSERT()` for handling this error instead of `DBUG_ABORT()`, which caused debugging problems on some platforms. (Bug #25818544)

  References: See also: Bug #21383497, Bug #21383530.

- **JSON:** The internal `Json_array` now uses a `std::vector` instead of a `Prealloced_array` to store its elements, which reduces the amount of heap space required by them. (Bug #85877, Bug #25867454)

- When the character set of one string comparison operand was a superset of the character set of the other operand, some comparisons were disallowed that should be permitted by converting the operand with the “smaller” character set to the “larger” character set. `utf8mb4` and `utf32` are considered to be a superset of any other encoding. (Bug #27897053, Bug #25642319, Bug #85224)

- The `mysqld_pre_systemd` script in RPM packages found the error log setting in option files if specified as `log-error` but not as `log_error`, though both are permitted. (Bug #26148391, Bug #86466)
- `ALTER TABLE ... ADD PARTITION` statements could cause a server exit. (Bug #26132947)
- `SET PERSIST` did not work for X Plugin system variables. (Bug #26115672)
- `REFERENCES` privilege checking could use the incorrect database in some cases. (Bug #26106655)
- With the `gtid_mode` system variable not set in a `.cnf` option file or on the command-line, the server failed to restart after using `SET PERSIST` or `SET GLOBAL` to set `gtid_mode` to `ON_PERMISSIVE`. (Bug #26085712)
- During prepared statement execution, too many bytes of a buffer could be read. (Bug #26042934, Bug #85937)
- Dynamic privileges were not loaded when the server was started using the `--initialize` option. (Bug #26005645)
- *Linux:* The generic Linux build for MySQL 8.0 now supports Non-Uniform Memory Access (NUMA) on its host system. Systems that use the build need to have `libnuma` installed on them. See Installing MySQL on Unix/Linux Using Generic Binaries for more details. (Bug #26005558)
- Debian client packages were missing information about conflicts with akonadi-backend-mysql packages. (Bug #26002288)
- `mysqldump` could write database names in `USE` statements incorrectly. (Bug #25998635)
- If the `mysql_stmt_close()` C API function was called, it freed memory that later could be accessed if `mysql_stmt_error()`, `mysql_stmt_errno()`, or `mysql_stmt_sqlstate()` was called. To obtain error information after a call to `mysql_stmt_close()`, call `mysql_error()`, `mysql_errno()`, or `mysql_sqlstate()` instead. (Bug #25988681)
- Upgrading from MySQL 5.7 to MySQL 8.0 failed during data dictionary creation due to table names that exceeded the 64 character length limit. (Bug #25973237, Bug #86120)
- Queries could be cached incorrectly, leading to incorrect query results, under these circumstances: `InnoDB` table; rows are being inserted but have not yet been committed; a query uses the table as a base table in a derived table; the optimizer chooses to materialize the derived table. (Bug #25943038, Bug #86047)
- `mysqld` misbehaved if compiled with AddressSanitizer enabled and invoked with `--basedir=` (that is, with an empty option value). (Bug #25914296, Bug #85994)
- The `ST_LatFromGeohash()`, `ST_LongFromGeohash()`, and `ST_PointFromGeohash()` functions did not allow data that originated from a `CHAR` column. (Bug #25912557, Bug #85981)
- Attempting to drop a data file from a tablespace with multiple data files returned a duplicate file name error. (Bug #25858461)
- An attribute was added to data dictionary client method declarations to generate compilation warnings when return values are ignored. (Bug #25840927)
- The data dictionary class hierarchy was simplified. (Bug #25835968, Bug #85811)
- A restriction that prevented the data dictionary object update function (`Dictionary_client::update`) from being called twice on the same object was removed. (Bug #25833932, Bug #85800)

- On Ubuntu platforms, the MySQL service script did not terminate correctly if the user running the script did not have permission to access the data directory or PID file. (Bug #25825833)
- String comparison queries on the `INFORMATION_SCHEMA.CHARACTER_SETS.DESCRIPTION` field were case sensitive, which is a regression from earlier releases. (Bug #25824297)
- Man pages for a few utilities were missing from Debian/Ubuntu packages. (Bug #25811814)
- An `INFORMATION_SCHEMA` view executed in `LOCK TABLES` mode, using the `SERIALIZABLE` isolation level, and with `autocommit` disabled, failed to use non-locking reads, which could block DDL operations or cause deadlocks. (Bug #25811413)
- Table locking failures could occur if tables were used by a trigger for which a trigger of the same name existed in another database and the database names differed only in lettercase. (Bug #25807393)
- An in-place MySQL upgrade failed if a `.TRG` file was missing the `created` line for the trigger definition. (This problem is unlikely to occur for upgrades to MySQL 8.0 from 5.7 because the `created` line is present for MySQL 5.7.2 and higher.) (Bug #25805260, Bug #85704)
- The `field-t` unit test failed to run with AddressSanitizer enabled. Thanks to Laurynas Biveinis for the patch. (Bug #25803823, Bug #85678)
- Debian client packages were missing information about conflicts with native packages. (Bug #25799475)
- The `CREATE_OPTIONS` column of the `INFORMATION_SCHEMA.TABLES` table was treated as having the `binary` collation, so functions such as `UPPER()` and `LOWER()` did not have the intended result. (Bug #25793429)
- After importing a table with `DECIMAL` column, accessing the table raised an assertion. (Bug #25792649)
- `ALTER TABLE` could fail when the default character set changed to `utf8mb4` due to incorrect column length calculations. (Bug #25779239, Bug #85614)
- If a component `deinit()` method failed, the component was still unloaded. (Bug #25764325)
- Stored program execution could fail for `DEFINER` accounts with expired passwords even if they were locked and could not be used to connect to the server. Now `DEFINER` accounts are not checked for expired passwords if they are locked. (Bug #25741966)
- `DISTINCT` operations on temporary tables could produce incorrect results due to allocation of too-small comparison keys. (Bug #25740550, Bug #85518)
- In optimizer trace output, `num_tmp_files` did not actually indicate number of files. It has been renamed to `num_initial_chunks_spilled_to_disk` and indicates the number of chunks before any merging has occurred. (Bug #25733784, Bug #85487)
- The Perl path in `#!` lines at the beginning of Perl scripts has been adjusted to `/usr/local/bin/perl` for FreeBSD 11. (Bug #25719975)
- With the `IGNORE_SPACE` SQL mode enabled, syntax error messages always reported line 1 as the line number. (Bug #25717617)
- `mysqldump` failed to properly quote certain identifiers in SQL statements written to the dump output. (Bug #25717383)
- Dropping a tablespace without that associated tablespace file failed with an error stating that the tablespace dictionary object is invalid. (Bug #25717019)
- Client preauthorization by the server was missing a length check for a length-encoded string. (Bug #25714674)

- In debug builds, a `CREATE TABLESPACE` operation raised an invalid assertion when using the NDB storage engine. A validation function that checked for zero-length data files did not apply to NDB tablepaces and was removed. (Bug #25700242)
- For debug builds, `EXPORT_SET()` operations could raise an assertion for some arguments. (Bug #25688192)
- For debug builds, the assertion added for Bug#59686 was too strict and could be raised when it should not have been. (Bug #25685958)

  References: See also: Bug #59686.
- A code refactoring in MySQL 8.0.1 caused several assertions to be raised in debug builds. (Bug #25669590, Bug #25669606, Bug #25669580, Bug #25688504)

  References: This issue is a regression of: Bug #25221172, Bug #84103.
- For some double-precision calculations, overflow could occur when calculating the exponent part. (Bug #25664323, Bug #85290)
- `SHOW INDEX` output was inconsistent for a `FULLTEXT` index defined on multiple columns. (Bug #25659276)
- During a group commit, the stack could incorrectly be reported exhausted, leading to stack overflow. (Bug #25656875)
- An in-place (binary) upgrade could change the default `character_set_client` and `character_set_connection` variables for stored routines, possibly resulting in different routine behavior. (Bug #25633041)
- For debug builds, the partitioning handler could inappropriately evaluate generated column expressions in `UPDATE` statements, resulting in a raised assertion. (Bug #25615803, Bug #85179)
- In strict SQL mode, an `ER_TRUNCATED_WRONG_VALUE` error could be converted from a warning to an error but then ignored, leading to a raised assertion. (Bug #25586959, Bug #25586673)
- Failure occurred for `ALTER TABLE` on an `ARCHIVE` table containing a `NOT NULL` column having a geometry data type. For debug builds, an assertion was raised. For non-debug builds, an error occurred. (Bug #25582178, Bug #85059)
- For debug builds, an assertion could be raised for `DROP TRIGGER` of a trigger for a table used by a view. (Bug #25581925)
- For upgrades from MySQL 5.7 to MySQL 8.0, the server wrote unnecessary parsing warnings to the error log. These messages are now suppressed. (Bug #25518436, Bug #84889)
- The server exited abnormally attempting to access invalid memory. (Bug #25501659)
- The maximum length of the `name` field in the `column_type_elements` table in the data dictionary was extended from 255 bytes to 1020 bytes to accommodate long `ENUM` and `SET` values that could result from using a multibyte character set.

  When using a multibyte character set, a single `ENUM` or `SET` element occupies a maximum of *`M`* x *`w`* bytes in the column type definition in the data dictionary, where `M` is the element literal length and *`w`* is the number of bytes required for the maximum-length character in the character set.

  In prior releases, the maximum supported length of an individual `ENUM` or `SET` element depended on the number of elements in the type. Thus, there could be a single element with (*`M`* x *`w`*) = 64K, or 64K elements with (*`M`* x *`w`*) = 1.

The maximum supported length of an individual `ENUM` or `SET` element is now *`M`* <= 255 and (*`M`* x *`w`*) <= 1020, regardless of the number of elements in the type. (Bug #25481355)

- With `mysqld` secured by TCP wrappers and the `hosts.allow` and `hosts.deny` files configured to restrict access from an IP address, connection attempts from that address resulted in too many messages to the error log. (Bug #25476479, Bug #84708)
- `mysqlpump` no longer includes the `slave_master_info` and `slave_relay_log_info` tables in dumps of the `mysql` system database. Restoring a dump file containing these tables caused problems by changing the replication state improperly. (Bug #25469190)
- Changes made by calling `mysql_options()` to set `MYSQL_OPT_SSL_MODE` could be affected by later `mysql_options()` calls. Now setting `MYSQL_OPT_SSL_MODE` is unaffected by later `mysql_options()` calls. (Bug #25452210)
- A race condition could occur for `CREATE TABLE` statements with `DATA DIRECTORY` or `INDEX DIRECTORY` clauses. (Bug #25451091)
- With AddressSanitizer enabled, compiling the `keyring_file` plugin produced One-Definition Rule violations. (Bug #25448205)
- `ALTER TABLE ... MODIFY` on a `DATETIME NOT NULL` column using an `AFTER` clause resulted in an `ER_INVALID_USE_OF_NULL` error. (Bug #25385334)
- The range optimizer could create an incorrect query tree, resulting in a server exit. (Bug #25369742, Bug #25586531)
- `mysqld_failed` to start the server if the `--datadir` option was specified with a relative path name. (Bug #25364806)
- `XA PREPARE`, `XA ROLLBACK`, and `XA COMMIT` for a transaction from a disconnected session did not take a global commit lock and modified the binary log and `InnoDB` redo log even when `FLUSH TABLES WITH READ LOCK` was in effect. This could lead to inconsistent backups when backup tools assumed that the server was in a read-only state. (Bug #25364178, Bug #84442)
- `GROUP_CONCAT(DISTINCT)` returned nonunique values if the data size was greater than the value of the `tmp_table_size` system variable. (Bug #25331425, Bug #84320)
- The fix for Bug #78777 had different effects depending on whether the Performance Schema is enabled. (Bug #25309017, Bug #84305)

  References: This issue is a regression of: Bug #78777.
- An aggregate function in some nested queries could cause a server exit. (Bug #25303711)
- Virtual generated column expressions that used the `BIN_TO_UUID()`, `CONV()`, or `HEX()` functions could cause problems if the connection character set was changed. In this context, the table character set is now used for these functions regardless of connection character set. (Bug #25287633)
- The `Rewriter` plugin did not perform locking properly if the `read_only` system variable was enabled. (Bug #25264253)
- With `read_only` enabled, creation of non-`TEMPORARY` tables by non-`SUPER` users was permitted under certain conditions. (Bug #25250768)
- For a table having a `TIMESTAMP` or `DATETIME` column having a default of `CURRENT_TIMESTAMP`, the column could be initialized to `0000-00-00 00:00:00'` if the table had a `BEFORE INSERT` trigger. (Bug #25209512, Bug #84077)

- On Windows, `Time` values in `SHOW PROCESSLIST` output drifted higher over time. (Bug #25101724, Bug #83019)
- An assertion was raised when a create function at server bootstrap attempted to evaluate non-resolved expressions. (Bug #24961932)
- The number of data dictionary cache lookups was reduced, and `ALTER TABLE` handling of triggers was improved help ensure trigger persistence in case of a server exit during `ALTER TABLE` processing. (Bug #24930129, Bug #83473)
- Certain queries against `InnoDB` tables that used a primary key and a subquery could return incorrect results if the `index_merge_intersection` flag of the `optimizer_switch` system variable was enabled. (Bug #24829050, Bug #79675)
- On x86 machines, the `uint3korr()` macro read 4 bytes of data instead of the intended 3 bytes. (Bug #24807826, Bug #83264)
- An assertion was raised during a fetch operation by the memcached plugin. (Bug #24605783)
- Queries that contained `UNION` in a subquery and `GROUP BY` could return incorrect results. (Bug #24595639)
- Some syntactically incorrect `CREATE INDEX` statements could cause a server exit rather than a syntax error. (Bug #24593992)
- An in-place MySQL upgrade failed if the `innodb_table_stats` or `innodb_index_stats` tables were not present in the `mysql` system database. (This problem should not occur for upgrades to MySQL 8.0 from 5.7 because those tables are present in MySQL 5.7.) (Bug #24557143)
- Incorrect behavior could occur for `INSERT` statements executed in stored-program or prepared-statement context, if the `VALUES` part of an `ON DUPLICATE KEY UPDATE` clause referred to a `BLOB` value in the `INSERT` column list. (Bug #24538207, Bug #25361251, Bug #25530880, Bug #25684790)
- systemd support scripts in Debian packages contained hardcoded references to the data directory, making it difficult to change the data directory using `--datadir`. (Bug #24398446, Bug #82417)
- MySQL failed to compile under macOS 10.10.5 using Clang. (Bug #24352163, Bug #82340)
- If a `REPLACE` statement tried to update a row in a table containing a virtual generated column of type `BLOB`, subsequent DML statements could behave incorrectly. (Bug #23573575)
- Some `PROXY` grants were not replicated to slaves, causing incorrect replication. (Bug #23289541, Bug #81424, Bug #23623115)
- `EXPLAIN` for single-table `UPDATE` or `DELETE` statements could raise an assertion attempting to optimize away subqueries. (Bug #23209903)
- The help output from `mysqlxtest` has been improved. (Bug #23107137, Bug #81086)
- For builds with AddressSanitizer enabled, the `ST_Simplify()` function could attempt to use already freed memory. (Bug #23023817)
- Compiler flags were adjusted to eliminate numerous warnings that occurred when compiling the `keyring_file` plugin using Clang. (Bug #22834591, Bug #80524)
- If enabling the Event Scheduler caused an event defined as `ON COMPLETION NOT PRESERVE` to be dropped because its execution time had passed, the drop event was not written to the binary log, causing slaves not to replicate it and replication failure if an event of the same name was created later. (Bug #22150112)

- For some `CREATE TABLE ... SELECT` statements, adding an `ORDER BY` clause changed column data types, or caused an assertion to be raised for debug builds. (Bug #16833464)
- `LOAD XML` performance became noticeably slower when the XML file being read contained a great many spaces, such as those introduced by indenting or pretty-printing. Now all leading whitespace is trimmed from each such value before reading it into memory. (Bug #16212207)

# Changes in MySQL 8.0.1 (2017-04-10, Development Milestone)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

**Note**

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward. (Making a backup before the upgrade is a prudent precaution in any case.)

**Note**

This release makes several important changes in Unicode character set support. In particular, the default character set has changed from `latin1` to `utf8mb4`.

- Account Management Notes
- C API Notes
- Character Set Support
- Compilation Notes
- Configuration Notes
- Connection Management Notes
- Data Dictionary Notes
- Deprecation and Removal Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Plugin Notes
- Security Notes
- Spatial Data Support
- SQL Syntax Notes
- Test Suite Notes

- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed

## Account Management Notes

- **Incompatible Change:** Components and plugins now can define privileges dynamically (at runtime), which enables them to create privileges specifically associated with the capabilities they implement.

  Implementation of dynamic privileges enables DBAs to begin migrating away from the `SUPER` privilege. In the past, `SUPER` has been associated with a wide and growing variety of operations, not all related to each other. Many operations covered by `SUPER` now are also associated with a dynamic privilege of more limited scope, and that privilege is registered by the component or plugin that implements the operation. Each such operation that previously required the `SUPER` privilege can be permitted to an account by granting the associated dynamic privilege rather than `SUPER`. For example, a user who must be able to modify global system variables can be granted `SYSTEM_VARIABLES_ADMIN` rather than `SUPER`.

  This change improves security by enabling DBAs to avoid granting `SUPER` and tailor user privileges more closely to the operations permitted. `SUPER` is now deprecated and will be removed in a future MySQL version.

  Dynamic privileges apply only at the global level. The server stores information about current assignments of dynamic privileges to user accounts in `mysql.global_grants`, a new table in the `mysql` system database.

  For more information, see Static Versus Dynamic Privileges. That discussion includes instructions for migrating accounts away from `SUPER` to dynamic privileges.

  Incompatibility: `SHOW GRANTS` no longer displays `ALL PRIVILEGES` in its global-privileges output because the meaning of `ALL PRIVILEGES` at the global level varies depending on which dynamic privileges are defined. Instead, `SHOW GRANTS` explicitly lists each granted global privilege. For details, see SHOW GRANTS Statement. Applications that process `SHOW GRANTS` output should be adjusted accordingly.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `mysql` system database. (WL #8131)

## C API Notes

- The C API implemented by `libmysqlclient` now includes a client interface for reading a stream of replication events from a MySQL server binary log. For more information, see C API Binary Log Interface (WL #7672)

## Character Set Support

- **Important Change:** The default character set has changed from `latin1` to `utf8mb4`. These system variables are affected:

  - The default value of the `character_set_server` and `character_set_database` system variables has changed from `latin1` to `utf8mb4`.
  - The default value of the `collation_server` and `collation_database` system variables has changed from `latin1_swedish_ci` to `utf8mb4_0900_ai_ci`.

As a result, the default character set and collation for new objects differ from previously unless an explicit character set and collation are specified. This includes databases and objects within them, such as tables, views, and stored programs. One way to preserve the previous defaults is to start the server with these lines in the `my.cnf` file:

```
[mysqld]
character_set_server=latin1
collation_server=latin1_swedish_ci
```

(WL #7554)

- Performance of UCA 9.0.0-based collations (for example, `utf8mb4_0900_ai_ci`) was improved. These collations are now faster than any other UCA collations.

  Additionally, the `max_length_for_sort_data` system variable default value has been increased from 1024 to 4096. (Bug #24823885, Bug #83319)

- The default collation for the `utf8mb4` character set has changed from `utf8mb4_general_ci` to `utf8mb4_0900_ai_ci`; `utf8mb4_general_ci` does not handle characters outside the Basic Multilingual Plane (BMP) correctly. (Bug #24742157)

- The pad attribute for Unicode 9.0.0 collations was changed from `PAD SPACE` to `NO PAD`. Consequently, these collations now treat spaces at the end of strings as significant, like any other character. The affected collations have names that contain the string `_0900_`.

  Comparisons of nonbinary strings (`CHAR`, `VARCHAR`, and `TEXT` strings) that have a 9.0.0 (`NO PAD`) collation differ from `PAD SPACE` collations with respect to trailing spaces. For example, `'a'` and `'a '` compare as different strings, not the same string. Example:

```
mysql> SET NAMES 'latin1' COLLATE 'latin1_swedish_ci';
mysql> SELECT 'a' = 'a ';
+------------+
| 'a' = 'a ' |
+------------+
|          1 |
+------------+
mysql> SET NAMES 'utf8mb4' COLLATE 'utf8mb4_0900_ai_ci';
mysql> SELECT 'a' = 'a ';
+------------+
| 'a' = 'a ' |
+------------+
|          0 |
+------------+
```

  See also Trailing Space Handling in Comparisons.

  The `INFORMATION_SCHEMA COLLATIONS` table now has a `PAD_ATTRIBUTE` column that indicates the pad attribute for each collation.

  A problem with the `latin1_de` collation involving early weight string truncation has been corrected. The only likely effect is for `WEIGHT_STRING()` function results. (WL #10354)

- Complementing earlier work in MySQL 8.0.0 to add case-insensitive and accent-insensitive collations for the `utf8mb4` Unicode character set, new case-sensitive and accent-sensitive collations have been added. The general collation is named `utf8mb4_0900_as_cs`, and there are language-specific collations with characteristics similar to `utf8mb4_0900_as_cs` except that language-specific rules take precedence where applicable. The language-specific collations are indicated by ISO 639-1 language codes in the collation name, as shown in the following table. In two cases the language code has an additional item that denotes a variant (German phone book order, Traditional Spanish).

**Note**

`utf8mb4_ja_0900_as_cs` is the first Japanese language-specific collation available for Unicode in MySQL.

**Table 2 utf8mb4 UCA 9.0.0 Language-Specific Collations**

| Language | Collation |
|---|---|
| Croatian | `utf8mb4_hr_0900_as_cs` |
| Czech | `utf8mb4_cs_0900_as_cs` |
| Danish | `utf8mb4_da_0900_as_cs` |
| Esperanto | `utf8mb4_eo_0900_as_cs` |
| Estonian | `utf8mb4_et_0900_as_cs` |
| German phone book order | `utf8mb4_de_pb_0900_as_cs` |
| Hungarian | `utf8mb4_hu_0900_as_cs` |
| Icelandic | `utf8mb4_is_0900_as_cs` |
| Japanese | `utf8mb4_ja_0900_as_cs` |
| Latvian | `utf8mb4_lv_0900_as_cs` |
| Lithuanian | `utf8mb4_lt_0900_as_cs` |
| Polish | `utf8mb4_pl_0900_as_cs` |
| Classical Latin | `utf8mb4_la_0900_as_cs` |
| Romanian | `utf8mb4_ro_0900_as_cs` |
| Slovak | `utf8mb4_sk_0900_as_cs` |
| Slovenian | `utf8mb4_sl_0900_as_cs` |
| Modern Spanish | `utf8mb4_es_0900_as_cs` |
| Traditional Spanish | `utf8mb4_es_trad_0900_as_cs` |
| Swedish | `utf8mb4_sv_0900_as_cs` |
| Turkish | `utf8mb4_tr_0900_as_cs` |
| Vietnamese | `utf8mb4_vi_0900_as_cs` |

`utf8mb4_0900_as_cs` also works as an accent-sensitive, case-sensitive collation for the languages in the following table.

**Table 3 Languages for Which utf8mb4_0900_as_cs is Suitable**

| Language Name | Language Code |
|---|---|
| German (dictionary order) | de |
| English | en |
| French (locale fr_FR) | fr |
| Irish Gaelic | ga |
| Indonesian | id |
| Italian | it |
| Luxembourgian | lb |

| Language Name | Language Code |
|---|---|
| Malay | ms |
| Dutch | nl |
| Portuguese | pt |
| Swahili | sw |
| Zulu | zu |

`utf8mb4_0900_as_cs` is suitable for French French (locale `fr_FR`) but not for Canadian French (locale `fr_CA`). For Canadian French, `utf8mb4_0900_ai_ci` is suitable. The reason for the difference between `ai_ci` and `as_cs` collations is that Canadian French has a different order of accented characters than French French.

`utf8mb4_da_0900_as_cs` also works as an accent-sensitive, case-sensitive collation for the languages in the following table.

**Table 4 Languages for Which utf8mb4_da_0900_as_cs is Suitable**

| Language Name | Language Code |
|---|---|
| Norwegian | no |
| Norwegian Bokmål | nb |
| Norwegian Nynorsk | nn |

The nonlanguage-specific `utf8mb4_0900_as_cs` and language-specific `utf8mb4_LANG_0900_as_cs` Unicode collations each have these characteristics:

- The collation is based on Unicode Collation Algorithm (UCA) 9.0.0 and Common Locale Data Repository (CLDR) v30, is accent sensitive, and case sensitive. These characteristics are indicated by `_0900`, `_as`, and `_cs` in the collation name. Exception: `utf8mb4_la_0900_as_cs` is not based on CLDR because Classical Latin is not defined in CLDR.
- The collation works for all characters in the range [U+0, U+10FFFF].
- If the collation is not language specific, it sorts all characters, including supplemental characters, in default order (described following). If the collation is language specific, it sorts characters of the language correctly according to language-specific rules, and characters not in the language in default order.
- By default, the collation sorts characters having a code point listed in the DUCET table (Default Unicode Collation Element Table) according to the weight value assigned in the table. The collation sorts characters not having a code point listed in the DUCET table using their implicit weight value, which is constructed according to the UCA.

  The collation sorts on all three DUCET weight levels, including the tertiary level. This compares with accent-insensitive, case-insensitive collations, which sort only on the primary and secondary levels.
- For non-language-specific collations, characters in contraction sequences are treated as separate characters. For language-specific collations, contractions might change character sorting order.

For more information, see Unicode Character Sets. (WL #9109, WL #9751)

## Compilation Notes

- **Incompatible Change:** The `my_bool` type is no longer used in MySQL source code. Any third-party code that used this type to represent C boolean variables should use the `bool` or `int` C type instead.

  **Note**

  The change from `my_bool` to `bool` means that the `mysql.h` header file now requires a C++ or C99 compiler to compile.

  (Bug #25597667)

- **InnoDB:** A LOB page deletion function failed to delete LOB pages. (Bug #24480254)
- Windows builds now use the default runtime libraries (builds use the `/MD` flag). (Bug #25611609)
- `CMake` support was added for compiling with Developer Studio 12.6. (Bug #25384295)
- The rapidjson library included in MySQL distributions was upgraded to version 1.1.0. (Bug #24947436, Bug #83515)
- The minimum version of the Boost library for server builds is now 1.63.0. (Bug #24579061, Bug #82834, Bug #25126144, Bug #83905)
- For GCC versions higher than 4.4, `-fno-expensive-optimizations` was replaced with `-ffp-contract=off`, which has the effect of enabling more optimizations. Thanks to Alexey Kopytov for the patch. (Bug #24571672, Bug #82760)
- For building MySQL 8.0, the minimum required version of `CMake` is now 3.2.3 on Windows, 3.4.0 on Solaris, and 2.8.12 otherwise. (Bug #24481181, Bug #82628)

  References: See also: Bug #24687701.

- Work was done to clean up the source code base, including: Removing unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.

## Configuration Notes

- In source distributions, several utilities previously in the `extra` directory have been moved to the new `utilities` directory. (Bug #25416084)
- Two new `CMake` options are available for debugging. `LINK_RANDOMIZE` indicates whether to randomize the order of symbols in the `mysqld` binary (default `OFF`), and `LINK_RANDOMIZE_SEED` specifies a seed value for `LINK_RANDOMIZE`. (Bug #25336715)
- MySQL failed to compile if `-DENABLE_DEBUG_SYNC=OFF` AND `-DWITH_DEBUG=ON` were both given. The `ENABLE_DEBUG_SYNC` option has been removed and enabling `WITH_DEBUG` enables Debug Sync. (Bug #18374703)
- These changes were made with respect to persisted system variables:
  - A new statement, `RESET PERSIST`, enables removal of variable settings that were persisted with `SET PERSIST` to the `mysqld-auto.cnf` file that stores persisted global system variable settings.
  - Using `SET PERSIST` (or `@@PERSIST.`) to set a global variable to `DEFAULT` or to the variable literal default value previously also added a setting for the variable to the `mysqld-auto.cnf` file if was is

not present, or removed it from `mysqld-auto.cnf` if it was present. Now the assignment always adds a setting for the variable to the `mysqld-auto.cnf` file; to remove the setting from the file, use `RESET PERSIST`.

- A new Performance Schema table, `persisted_variables`, provides an SQL interface to the `mysqld-auto.cnf` file, enabling the file contents to be inspected at runtime using `SELECT` statements.
- The Performance Schema `variables_info` table has new columns showing when and by which user each system variable was most recently set.

For more information, see Persisted System Variables, and RESET PERSIST Statement.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the Performance Schema. (WL #9720, WL #9763)

References: See also: Bug #24522064.

- The default value of the `query_cache_size` system variable has been reduced from 1M to 0. Consequently no query cache buffer is allocated by default. (WL #9817)

## Connection Management Notes

- MySQL Server now includes a plugin library that enables administrators to introduce an increasing delay in server response to clients after a certain number of consecutive failed connection attempts. This capability provides a deterrent that slows down brute force attacks that attempt to access MySQL user accounts. For more information, see Connection Control Plugins. (WL #8885)

## Data Dictionary Notes

- SQL-layer code and the storage engine API were extended to provide data dictionary support for:
  - Storing information about auxiliary columns and keys that `InnoDB` adds to tables implicitly, storage engine-private data and object identifiers for tables and tablespaces, and `InnoDB` full-text search auxiliary tables and other similar, implicitly created objects.
  - Combining data dictionary updates, storage engine changes, and binary log writes for DDL operations into atomic transactions.

  A behavior change for `DROP TABLE` occurs if any named tables do not exist. Previously, the statement returned an error indicating which tables did not exist and it was unable to drop, but also dropped the tables that did exist. Now the statement still indicates which tables did not exist, but fails with an error and no changes are made.

  For more information, see Atomic Data Definition Statement Support, and WL#7743. (WL #7743)

- These `INFORMATION_SCHEMA` tables have been reimplemented as views on data dictionary tables:

```
EVENTS
PARAMETERS
ROUTINES
TRIGGERS
```

  Queries on those tables are now more efficient because they obtain information from data dictionary tables rather than by other, slower means. For example, the server no longer must create a temporary table for each query of the `INFORMATION_SCHEMA` table.

Improvements for those tables also apply to `SHOW` statements that display information corresponding to the `INFORMATION_SCHEMA` tables. For example, `SHOW TRIGGERS` displays the same information as the `TRIGGERS` table.

If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes. (WL #9494)

## Deprecation and Removal Notes

- **Incompatible Change; InnoDB; Partitioning:** The `ALTER TABLE ... UPGRADE PARTITIONING` statement is no longer supported.
- The deprecated `PROCEDURE ANALYSE()` syntax has been removed. (WL #4745)
- The deprecated `libmysqld` embedded server library has been removed, along with:
  - The `mysql_config --libmysqld-libs`, `--embedded-libs`, and `--embedded` options
  - The `CMake WITH_EMBEDDED_SERVER`, `WITH_EMBEDDED_SHARED_LIBRARY`, and `INSTALL_SECURE_FILE_PRIV_EMBEDDEDDIR` options
  - The (undocumented) `mysql --server-arg` option
  - The `mysqltest --embedded-server`, `--server-arg`, and `--server-file` options
  - The `mysqltest_embedded` and `mysql_client_test_embedded` test programs

  (WL #9721, WL #9722, WL #9816)
- The deprecated `--temp-pool` server option has been removed. (WL #8396)
- The deprecated `replace` utility has been removed. If you wish to continue using this utility, be sure to retain a copy from an installed version of MySQL before upgrading to the current version. (WL #9874)
- The parser no longer considers `\N` as a synonym for `NULL` in SQL statements. Use `NULL` instead.

  This change does not affect text file import or export operations performed with `LOAD DATA` or `SELECT ... INTO OUTFILE`, for which `NULL` continues to be represented by `\N`. See LOAD DATA Statement. (WL #7247)
- Support for DTrace has been removed. (WL #9581)

## Optimizer Notes

- **InnoDB:** MySQL now supports descending indexes: `DESC` in an index definition is no longer ignored but causes storage of key values in descending order. Previously, indexes could be scanned in reverse order but at a performance penalty. A descending index can be scanned in forward order, which is more efficient. Descending indexes also make it possible for the optimizer to use multiple-column indexes when the most efficient scan order mixes ascending order for some columns and descending order for others. For more information, see Descending Indexes.

**Note**

Previously, `GROUP BY` implicitly sorted by default (that is, in the absence of `ASC` or `DESC` designators for `GROUP BY` columns), but relying on implicit `GROUP BY` sorting was deprecated. `GROUP BY` no longer sorts by default, so query results

> may differ from previous MySQL versions. To produce a given sort order, use explicit `ASC` or `DESC` designators for `GROUP BY` columns or provide an `ORDER BY` clause.

(WL #1074, WL #7737)

- Optimizer trace output now includes more information about `filesort` operations, such as key and payload size and why addon fields are not packed. (Bug #25246184, Bug #84180)
- Previously, invisible indexes were supported only for the `InnoDB` storage engine. Invisible indexes are now storage engine neutral (supported for any engine). (Bug #23541244)
- A `GROUPING()` function has been introduced for use in `GROUP BY` queries that include a `WITH ROLLUP` modifier. This function enables you to distinguish `NULL` values for super-aggregate rows in the query result from `NULL` values in regular grouped rows. For more information, see Miscellaneous Functions, and GROUP BY Modifiers. Thanks to Zhe Dong for a patch that was partially used to implement this feature.

  **Note**

  `GROUPING` now is a reserved word and cannot be used as an identifier without identifier quoting.

  (Bug #11754449, Bug #46053, Bug #11745963, Bug #21728, WL #1979)

- The optimizer now supports `INDEX_MERGE` and `NO_INDEX_MERGE` hints to affect use of the Index Merge access method. Examples:

  ```
  SELECT /*+ INDEX_MERGE(t1 f1, f2, f3) */ f2 FROM t1
    WHERE f1 = 'o' AND f2 = f3 AND f3 <= 4;
  SELECT /*+ NO_INDEX_MERGE(t1 f2, f3) */ f1 FROM t1
    WHERE (f2 = 5 OR f3 = 'c') AND (f1 = 4 OR f4 = 'f');
  ```

  For more information, see Index Merge Optimization, and Optimizer Hints. (WL #9167)

- The `server_cost` and `engine_cost` optimizer cost model tables in the `mysql` system database now include a `default_value` column that contains the default value for each cost table estimate. This column is a read-only generated column that retains its value even if the associated cost estimate is changed. For more information, see The Optimizer Cost Model.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `mysql` system database. (WL #10128)

- In the optimizer cost model `engine_cost` table, the cost for memory access was decreased relative to the cost for disk access. An implication of this change is that the execution plan may change between two runs of the same query. For example, at server startup before data has been read into the buffer pool, you may get a different plan than after the query has been run because then the data will be in memory. (WL #8737)
- MySQL now supports common table expressions, both nonrecursive and recursive. Common table expressions enable use of named temporary result sets, implemented by permitting a `WITH` clause preceding `SELECT` statements and certain other statements. For more information, see WITH (Common Table Expressions).

  **Note**

  `RECURSIVE` now is a reserved word and cannot be used as an identifier without identifier quoting.

(WL #883, WL #3634)

- The optimizer now supports hints that enable specifying the order in which to join tables. For more information, see Optimizer Hints. (WL #9158)

## Packaging Notes

- **Microsoft Windows:** *Reminder:* MySQL 8.0 requires the Microsoft Visual C++ 2015 Redistributable Package to run on Windows platforms. Users should make sure the package has been installed on the system before starting the server. The package is available at the Microsoft Download Center.
- Changes in RPM package structure require a larger set of packages to be removed to install MySQL Server cleanly. (Bug #25603087)
- To avoid potential race conditions, Debian packages now use the GNU `install` utility rather than a combination of `mkdir`, `touch`, and `chown`. (Bug #25258829)
- New Debian/Ubuntu packages named `server-core` and `client-core` are now available. These packages contain binaries only, without configuration or service scripts, directory setup, man pages, and so forth. They are installed automatically with the standard packages, but when installed by themselves do not result in a functioning MySQL setup. They may be of interest for user which wish to customize their own installation setup. (Bug #25146364)
- `scripts/mysql_security_commands.sql` and `support-files/mysql.server-sys5.sh` are no longer used and have been removed from MySQL distributions. (Bug #24756400, Bug #24756442)
- RPM packages now are built with `-DWITH_NUMA=ON` for platforms with NUMA support: OEL higher than EL5, Fedora, SLES, Docker. (Bug #24689078)
- The `my-default.cnf.sh` file (used to produce a default `my-default.cnf` or `my-default.ini` file) is no longer included in source distributions and `my-default.cnf` and `my-default.ini` are no longer included in or installed by distribution packages. (Bug #22525354)
- The unused and undocumented `archive_reader` and `archive_test` utilities have been removed from MySQL source distributions. (Bug #12818207, Bug #62014)

## Performance Schema Notes

- The Performance Schema maintains statement event summary tables that contain information about minimum, maximum, and average statement latency. Those tables permit high-level assessment of system performance. To permit assessment at a more fine-grained level, the Performance Schema now also collects histogram data for statement latencies. These histograms provide additional insight into latency distributions and are available in these tables:

  - `events_statements_histogram_by_digest`: Statement histograms per schema and digest value
  - `events_statements_histogram_global`: Statement histogram summarized globally

  For more information, see Statement Histogram Summary Tables.

  In addition, the `events_statements_summary_by_digest` table now has columns that provide percentile information about statement latencies. For more information, see Statement Summary Tables.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the Performance Schema. (WL #5384)

- System and status variable information is no longer maintained in the `INFORMATION_SCHEMA`. These tables have been removed: `GLOBAL_VARIABLES`, `SESSION_VARIABLES`, `GLOBAL_STATUS`, `SESSION_STATUS`. Use the corresponding Performance Schema tables instead. See Performance Schema System Variable Tables, and Performance Schema Status Variable Tables. In addition, the `show_compatibility_56` system variable has been removed. It was used in the transition period during which system and status variable information in `INFORMATION_SCHEMA` tables was moved to Performance Schema tables, and is no longer needed. These status variables have been removed: `Slave_heartbeat_period`, `Slave_last_heartbeat`, `Slave_received_heartbeats`, `Slave_retried_transactions`, `Slave_running`. The information they provided is available in Performance Schema tables; see Migrating to Performance Schema System and Status Variable Tables. (WL #8057)

- The deprecated `INFORMATION_SCHEMA` `INNODB_LOCKS` and `INNODB_LOCK_WAITS` tables have been removed. To obtain `InnoDB` data lock information, use the Performance Schema `data_locks` and `data_lock_waits` tables instead. Alternatively, use the `sys` schema `innodb_lock_waits` and `x$innodb_lock_waits` views, which have been reimplemented to use the new Performance Schema tables rather than the removed `INFORMATION_SCHEMA` tables.

  The Performance Schema tables expose what data locks exist, which transaction holds each lock, and which transaction lock requests are blocked by other locks. For `InnoDB`, the `data_locks` table reports all data locks, in contrast to the `INNODB_LOCKS` table, which reported only waited-for data locks, For more information, see The data_locks Table, and The data_lock_waits Table. For information about mapping old table columns to new table columns, see The INFORMATION_SCHEMA INNODB_LOCKS Table, and The INFORMATION_SCHEMA INNODB_LOCK_WAITS Table.

  The tables differ in the privileges required: The `INFORMATION_SCHEMA` tables require the global `PROCESS` privilege. The Performance Schema tables require the usual Performance Schema privilege of `SELECT` on the table to be selected from.

  The `sys` schema `innodb_lock_waits` and `x$innodb_lock_waits` views can be used without change, with the exception that the `locked_table` column containing combined schema/table name values has been removed and replaced with `locked_table_schema` and `locked_table_name` columns that contain separate schema and table name values. In addition, these views now contain `locked_table_partition` and `locked_table_subpartition` columns. For more information, see The innodb_lock_waits and x$innodb_lock_waits Views,

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the Performance Schema and the `sys` schema. (WL #9275, WL #6657)

## Plugin Notes

- MySQL Server 8.0 includes a component-based infrastructure for improving server extensibility; see MySQL Components. However, MySQL plugins use an interface that predates the component interface. A new plugin service, `plugin_registry_service`, enables plugins to access the component registry and its services. (WL #4989)

## Security Notes

- **Incompatible Change:** For `STANDALONE` and `WIN` builds, the default `secure_file_priv` value has changed from the empty string to `NULL`. This is a secure-by-default setting because it disables import and export operations. To permit those operations, set `secure_file_priv` to the path name of the directory to use for those operations. (Bug #24679907, Bug #24695274, Bug #24707666, WL #6782)

- **Incompatible Change:** These changes were made to `mysqld_safe`:

- Unsafe use of `rm` and `chown` in `mysqld_safe` could result in privilege escalation. `chown` now can be used only when the target directory is `/var/log`. An incompatible change is that if the directory for the Unix socket file is missing, it is no longer created; instead, an error occurs. Due to these changes, `/bin/bash` is required to run `mysqld_safe` on Solaris. `/bin/sh` is still used on other Unix/Linux platforms.
- The `--ledir` option now is accepted only on the command line, not in option files.
- `mysqld_safe` ignores the current working directory.

Other related changes:

- Initialization scripts that invoke `mysqld_safe` pass `--basedir` explicitly.
- Initialization scripts create the error log file only if the base directory is `/var/log` or `/var/lib`.
- Unused systemd files for SLES were removed.

(Bug #24483092, Bug #25088048, Bug #25378439, Bug #25378565)

References: See also: Bug #24464380, Bug #24388753, Bug #24619033, Bug #82920.

- yaSSL was upgraded to version 2.4.2. This upgrade corrects issues with: Potential AES side channel leaks; DSA padding for unusual sizes; the `SSL_CTX_load_verify_locations()` OpenSSL compatibility function failing to handle long path directory names. (Bug #24512715, Bug #24740291)
- OpenSSL is ending support for version 1.0.1 in December 2016; see https://www.openssl.org/policies/releasestrat.html. Consequently, MySQL Commercial Server builds now use version 1.0.2 rather than version 1.0.1, and the linked OpenSSL library for the MySQL Commercial Server has been updated from version 1.0.1 to version 1.0.2j. For a description of issues fixed in this version, see https://www.openssl.org/news/vulnerabilities.html.

  This change does not affect the Oracle-produced MySQL Community build of MySQL Server, which uses the yaSSL library instead. (WL #9136)

## Spatial Data Support

- **Incompatible Change:** Previously, `ST_Distance()` supported only Cartesian spatial reference systems (SRSs). `ST_Distance()` now detects geometry arguments in a geographic SRS and returns the geodetic distance on the ellipsoid in meters. The only permitted geographic argument types are `Point` and `Point`, or `Point` and `MultiPoint` (in any argument order). Calculations for projected SRSs and SRID 0 remain the same. For more information, see Spatial Relation Functions That Use Object Shapes.

  **Note**

  If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

  (WL #9347)

- **Incompatible Change:** These functions previously accepted either WKB strings or geometry arguments. Geometry arguments are no longer permitted and produce an error.
  - `ST_GeomCollFromWKB()`, `ST_GeometryCollectionFromWKB()`

- `ST_GeomFromWKB()`, `ST_GeometryFromWKB()`
- `ST_LineFromWKB()`, `ST_LinestringFromWKB()`
- `ST_MLineFromWKB()`, `ST_MultiLinestringFromWKB()`
- `ST_MPointFromWKB()`, `ST_MultiPointFromWKB()`
- `ST_MPolyFromWKB()`, `ST_MultiPolygonFromWKB()`
- `ST_PointFromWKB()`
- `ST_PolyFromWKB()`, `ST_PolygonFromWKB()`

For information about migrating queries that refer to those functions away from using geometry arguments to using WKB arguments, see Functions That Create Geometry Values from WKB Values. (WL #10166)

- **Incompatible Change:** These functions now interpret latitude and longitude coordinates as in the order specified by the spatial reference system. The functions also accept an optional argument to override the default axis order.

  - `ST_GeomCollFromText()`, `ST_GeometryCollectionFromText()`, `ST_GeomCollFromTxt()`
  - `ST_GeomFromText()`, `ST_GeometryFromText()`
  - `ST_LineFromText()`, `ST_LinestringFromText()`
  - `ST_MLineFromText()`, `ST_MultiLinestringFromText()`
  - `ST_MPointFromText()`, `ST_MultiPointFromText()`
  - `ST_MPolyFromText()`, `ST_MultiPolygonFromText()`
  - `ST_PointFromText()`
  - `ST_PolyFromText()`, `ST_PolygonFromText()`
  - `ST_GeomCollFromWKB()`, `ST_GeometryCollectionFromWKB()`
  - `ST_GeomFromWKB()`, `ST_GeometryFromWKB()`
  - `ST_LineFromWKB()`, `ST_LinestringFromWKB()`
  - `ST_MLineFromWKB()`, `ST_MultiLinestringFromWKB()`
  - `ST_MPointFromWKB()`, `ST_MultiPointFromWKB()`
  - `ST_MPolyFromWKB()`, `ST_MultiPolygonFromWKB()`
  - `ST_PointFromWKB()`
  - `ST_PolyFromWKB()`, `ST_PolygonFromWKB()`

  For more information, see Functions That Create Geometry Values from WKT Values, and Functions That Create Geometry Values from WKB Values.

  **Note**

  If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

  (WL #9434, WL #9435)

- **Incompatible Change:** These functions now return latitude and longitude coordinates in the order specified by the spatial reference system that applies to the geometry value argument. The functions also accept an optional argument to override the default axis order.

  - `ST_AsBinary()`, `ST_AsWKB()`
  - `ST_AsText()`, `ST_AsWKT()`

  For more information, see Geometry Format Conversion Functions.

  **Note**

  If spatial data contains SRID values that refer to a geographic spatial reference system, existing queries using these functions will return different results, compared to previous MySQL versions.

  (WL #9432, WL #9433)

- The `st_spatial_reference_systems` data dictionary table that stores information about spatial reference systems other than SRID 0 has been upgraded from EPSG Dataset 8.7 to 9.0. Also, the table contents are now exposed through the `INFORMATION_SCHEMA ST_SPATIAL_REFERENCE_SYSTEMS` table, which is implemented as a view on the data dictionary For more information, see The INFORMATION_SCHEMA ST_SPATIAL_REFERENCE_SYSTEMS Table.

  MySQL now has an `INFORMATION_SCHEMA ST_GEOMETRY_COLUMNS` table that provides information about table columns that store spatial data. The new table is implemented as a view on the existing `INFORMATION_SCHEMA COLUMNS` table. For more information, see The INFORMATION_SCHEMA ST_GEOMETRY_COLUMNS Table.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these `INFORMATION_SCHEMA` changes. (Bug #25337054, Bug #84384, WL #9348, WL #8582)

- A new spatial function, `ST_SwapXY()`, accepts a geometry argument and swaps the X and Y values of each coordinate pair within the geometry. For more information, see Geometry Format Conversion Functions. (WL #9436)

## SQL Syntax Notes

- Derived table syntax now permits a list of explicit column names following the table name, to override the default column names taken from the select list. For example, the column names for this derived table come from its select list:

```
mysql> SELECT * FROM (SELECT 1, 2, 3, 4) AS dt;
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
```

  To provide column names, follow the derived table name with a parenthesized list of column names:

```
mysql> SELECT * FROM (SELECT 1, 2, 3, 4) AS dt (a, b, c, d);
+---+---+---+---+
| a | b | c | d |
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
```

(WL #883)

## Test Suite Notes

- `mysql-test-run.pl` now has a `--charset-for-testdb=charset_name` option for specifying the default character set for the `test` database. The default value is `latin1`. (Bug #25494068, Bug #84806)
- `mysql-test-run.pl` now has an `--only-big-tests` option, which causes only big tests to run. Normal (non-big) tests are skipped. If both `--big-test` and `--only-big-tests` are given, `--only-big-tests` is ignored. (Bug #25182306)
- `mysql-test-run.pl` has a new `--discover` option. When given, `mysql-test-run.pl` attempts to preload `discover`, the Developer Studio Memory Error Discovery Tool when starting `mysqld`. Reports from `discover` may be found in `log/mysqld.%p.txt` under the directory given by `--vardir`. This option is supported only on SPARC-M7 systems. (Bug #25048971)
- `mysqltest` now supports an `expr` command that enables an expression to be evaluated and assigned to a variable. For details, see the command description in the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html. (Bug #24806741)
- `mysqltest` now supports a `force-cpdir` command that copies a source directory to a destination directory recursively (that is, it copies subdirectories). (Bug #24806681)
- `mysql-test-run.pl` has a new `--summary-report` option. When given, `mysql-test-run.pl` generates a plain text version of the test summary only and writes it to the file named as the option argument. The file is suitable for sending by email. (Bug #24512357, Bug #82708)
- `mysql-test-run.pl` could not be run with --valgrind-option=--tool=custom_tool, for values of *`custom_tool`* such as `massif` or `helgrind`, because it added the options for `memcheck` that might not be understood by other tools. Also, the `mysql-test-run.pl --callgrind` option did not work because it supplied an invalid `--base` option to `callgrind`. Thanks to Daniel Black for the patch on which the fixes were based. (Bug #23713613, Bug #82039)
- The `mysqltest connect()` function now permits `SOCKET` and `TCP` values for the `options` argument, to specify using the socket-file and TCP/IP connection protocols.

  In addition, transport protocols now must match the current system. Previously, passing `PIPE` or `SHM` on non-Windows systems caused the default (socket-file) transport protocol to be used. Now this causes an error, and, similarly, passing `SOCKET` on Windows systems causes an error. (Bug #21046241)
- `mysql-test-run.pl` now supports a `--bootstrap` option that enables `mysqld` options to be designated as bootstrap options. When `mysql-test-run.pl` finds `--bootstrap` options in a `master.opt` file, it removes and reinitializes the data directory and restarts the server with the new option settings. (Bug #18184868)

## X Plugin Notes

- The `MYSQLX_TCP_PORT` `CMake` option was ignored. (Bug #25493867, Bug #84804)
- The protobuf compiler emitted “unused import” warnings for mysqlx.proto. (Bug #25444009, Bug #84641)
- Connections were released which were not initialized when validation occurred. (Bug #25392280)
- The `Mysqlx_notice_warning_sent` and `Mysqlx_notice_other_sent` status variable values did not increment as expected. (Bug #25289949, Bug #84258, Bug #25290001, Bug #84260)

- The manual definition of the `MYSQL_DYNAMIC_PLUGIN` macro was redundant. (Bug #25162590, Bug #83988)
- IPv6 connectivity support defined a new system variable `mysqlx_bind_address` with the default value of '0.0.0.0', but the default value should have been '*' so that connections to X Plugin may be made using an IPv6 address, an IPv4 address, or a hostname. (Bug #25047909, Bug #83688)
- The `stmt` field was marked as `required` in the message:

```
message Mysqlx.Crud.ModifyView {
   required Collection collection = 1;

   optional string definer = 2;
   optional ViewAlgorithm algorithm = 3;
   optional ViewSqlSecurity security = 4;
   optional ViewCheckOption check = 5;

   repeated string column = 6;
   required Mysqlx.Crud.Find stmt = 7;
}
```

  (Bug #24968735, Bug #83595)
- For a table created with `CREATE TABLE foo (doc JSON)`, the statement `list_objects` incorrectly reported the table as a collection. (Bug #24963952)
- Sending `Expect.Open` to a connected socket before authentication led to the following error messages:

```
... Plugin mysqlx reported: '10: Invalid message 24 received during client initialization'
... Plugin mysqlx reported: '10: ERROR reading from socket Bad file descriptor (9) 2'
```

  (Bug #24940144, Bug #83494)
- On Linux, MySQL Shell in JavaScript and Python interactive mode hid collections if the lettercase in the collection name was not uniform. (Bug #24848125)
- If multiple user accounts exist that match the authenticating user, the X Plugin would attempt to authenticate all matching accounts instead of the best matching account. (Bug #24847537)
- Boost classes with corresponding functionality in C++ 11 were moved or wrapped. (Bug #24680856)
- When two X Plugin instances were started on the same port, the resulting error message was split over two lines and contained duplicate information. (Bug #24679018, Bug #83008)
- The `performance_schema.threads` table did not show `processlist` statistics for the X Plugin. (Bug #24638038)
- If the client sent an unknown message type to the server, the server would send back an error message and then disconnect the client. (Bug #24611754, Bug #82868)
- When the client sent a zero length message, the X Plugin did not send a response. For a subsequent message the client received an error `E_X_BAD_MESSAGE` and was disconnected, depending on the header content. (Bug #24595459, Bug #82862)
- X Plugin displayed its version to any connected user, including those not authenticated. (Bug #24562707, Bug #82784)
- X Plugin generated an incorrect query when a duplicate field name existed in an `Update` statement. (Bug #24510083)

- The statement `list_objects` returned incorrect information for some views in `sys` database. (Bug #24472325)
- X Plugin was trying to send result sets to a client, even where the connection had been closed. The plugin now stops execution of a stored procedure if the connection is reset. (Bug #24440344)
- Attempting to list objects using the `list_objects` statement without explicitly setting the database name resulted in the collection being incorrectly classified as a table. (Bug #23205895, Bug #81248)
- X Plugin now supports the `sha256_password` plugin. (WL #9271)

## Functionality Added or Changed

- **Incompatible Change:** Previously, for `INSERT ... SELECT ... ON DUPLICATE KEY UPDATE` statements for which the `UPDATE` referred to `SELECT` columns, the results could be incorrect if the `SELECT` was a `UNION`. Now such statements are prohibited and produce an error. To work around this restriction, write the `UNION` as a derived table so that its rows can be treated as a single-table result set, which is permitted. See INSERT ... ON DUPLICATE KEY UPDATE Statement. (WL #5094)
- **InnoDB:** By default, `InnoDB` reads uncommitted data when calculating statistics. In the case of an uncommitted transaction that deletes rows from a table, `InnoDB` excludes records that are delete-marked when calculating row estimates and index statistics, which can lead to non-optimal execution plans for other transactions that are operating on the table concurrently using a transaction isolation level other than `READ UNCOMMITTED`. To avoid this scenario, a new configuration option, `innodb_stats_include_delete_marked`, can be enabled to ensure that `InnoDB` includes delete-marked records when calculating persistent optimizer statistics. (Bug #23333990)
- **InnoDB:** Geometry parsing and bounding box computational code for R-trees was moved from `InnoDB` to the server. (WL #9439)
- **InnoDB:** `InnoDB` now supports `NOWAIT` and `SKIP LOCKED` options with `SELECT ... FOR SHARE` and `SELECT ... FOR UPDATE` locking read statements. `NOWAIT` causes the statement to return immediately if a requested row is locked by another transaction. `SKIP LOCKED` removes locked rows from the result set. See Locking Read Concurrency with NOWAIT and SKIP LOCKED.

  `SELECT ... FOR SHARE` replaces `SELECT ... LOCK IN SHARE MODE`, but `LOCK IN SHARE MODE` remains available for backward compatibility. The statements are equivalent. However, `FOR SHARE` supports `NOWAIT`, `SKIP LOCKED`, and `OF tbl_name` options. See SELECT Statement.

  `OF tbl_name` applies locking queries to named tables.

  **Note**

  `OF` now is a reserved word and cannot be used as an identifier without identifier quoting.

  `SELECT ... FOR SHARE` and `SELECT ... FOR UPDATE` statements now require the `SELECT` privilege and at least one of the `DELETE`, `LOCK TABLES`, or `UPDATE` privileges. Previously, only the `SELECT` privilege was required. (WL #8919, WL #3597)
- **InnoDB:** The `InnoDB` data-at-rest encryption feature now supports encryption of redo log and undo log data, controlled by the `innodb_redo_log_encrypt` and `innodb_undo_log_encrypt` configuration options. See Redo Log Encryption, and Undo Log Encryption. (WL #9290, WL #9289)
- **InnoDB:** `InnoDB` internal temporary tables that are stored on disk now support multiple cursor positions, permitting single writer and multiple reader access within the same thread. The purpose of this enhancement is to provide support for recursive and non-recursive common table expressions (CTEs).

Additionally, consistent-read access to `InnoDB` on-disk internal temporary tables is replaced by a dirty-read scheme, and row counts for `InnoDB` on-disk internal temporary tables now use row count statistics instead of slower table scans. (WL #9248)

- **InnoDB:** `InnoDB` now compresses large objects into a sequence of smaller `zlib` streams for tables that use `ROW_FORMAT=COMPRESSED`. Previously, large object data was compressed into a single `zlib` stream. (WL #9263)

- **Replication:** When a negative or fractional timeout parameter was supplied to `WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS()`, the server behaved in unexpected ways. With this fix:

  - A fractional timeout value is read as-is, with no round-off.

  - A negative timeout value is rejected with an error if the server is on a strict SQL mode; if the server is not on a strict SQL mode, the value makes the function return NULL immediately without any waiting and then issue a warning.

  (Bug #24976304, Bug #83537)

- **Replication:** Work has been done to reduce contention between receiver (I/O) and applier (SQL) replication threads. This ensures slaves keep up with a higher insert load on their master. (Bug #78246, Bug #21753696, WL #8599)

- **Replication:** Added the `binlog_expire_logs_seconds` system variable, which sets an interval in seconds for purging of the binary log. The effects of this variable and `expire_logs_days` are cumulative, making it possible to set a period such as 1.5 days. To completely disable automatic binary log purging, set both variables equal to 0, which is the default value for both of them. (Bug #71697, Bug #18260088, WL #9237)

- **Replication:** Performance schema tables have been added to monitor replication lags and queues. The `replication_connection_status` table has updated information on the last transaction queued in the relay log, as well as the transaction currently being queued in the relay log. The `replication_applier_status_by_coordinator` table has updated information on the last transaction written to the buffer of a worker, as well as the transaction currently being processed by the coordinator. The `replication_applier_status_by_worker` table has updated information on the last transaction applied by the worker, as well as on the transaction currently being applied. The information presented in each of these tables contains the transaction's GTID, commit timestamps, processing stage timestamp, and the timestamp of the completion.

  In the `replication_applier_status_by_worker` table, the `LAST_SEEN_TRANSACTION` column is replaced by the `APPLYING_TRANSACTION` column. (WL #7374)

- **Replication:** The new binlog_expire_logs_seconds variable is a global server variable in addition to the existing expire_logs_days variable. The purpose is to facilitate finer grained retention policies of binary logs than the current day granularity. (WL #9237)

- **Replication:** Correct recovery of DDL statements or transactions by binary log implements support for correct recovery of DDL transactions based on the binary log data. In the context of the new data-dictionary it becomes possible to make metadata changes and write them to the binary log as a single transaction by using the 2-phase commit protocol. (WL #9175)

- **Replication:** Multi-source replication now supports per-channel replication filters. This makes it possible to filter out the execution of selected replicated data from a specific channel. Until now replication filters have been global, and only applicable to all channels. (WL #7361)

- **Replication:** Row-based replication now uses extended table metadata in the binary log. The extended metadata serves two major purposes: slaves use the metadata to transfer data smoothly when its table structure is different from master's, and external software can use the metadata to decode row events and store the data into external databases. (WL #4618)

- **Replication:** An infrastructure for GTID based delayed replication and replication lag monitoring has been added to enable you to properly monitor replication lag. Two new timestamps that are associated with each transaction (not each event or statement) in the binary log have been introduced. The original_commit_timestamp is in microseconds since the epoch when the transaction was committed on the original master, and the immediate_commit_timestamp is in microseconds since the epoch when the transaction was committed on the immediate master. (WL #7319)

- **Replication:** The `RESET MASTER` statement has been extended to allow specification of a binary log file number. The `RESET MASTER TO` statement accepts an argument which specifies the index of the new binary log file to use. The purpose is to simplify failover procedures by replacing the `FLUSH BINARY LOGS` and `PURGE BINARY LOGS TO` statements with a single `RESET MASTER` statement. (WL #9110)

- **Replication:** The process used for how delay is defined and calculated for delayed replication have been improved. This relies on a new timestamp in the binary log, the immediate_commit_timestamp, which is the number of microseconds since epoch when the transaction was written to the binary log of the immediate master. This means that the delay is no longer computed per-event, it is now applied per-transaction. (WL #7318)

- **Replication:** It is now possible to specify whether information written into the binary log enables replication slaves to parallelize based on commit timestamps, or on transaction write sets.

  Using write sets has a the potential for greater parallelism than using commit timestamps since it does not depend on the commit history. When applying binary logs in this fashion on a replication slave, it may be able to leverage capabilities of the underlying computing hardware (such as CPU cores) and thus speed up this process.

  The interface for choosing the source of parallelization is implemented as a new server system variable `binlog_transaction_dependency_tracking` which can take any one of the values `COMMIT_ORDER`, `WRITESET`, or `WRITESET_SESSION`. `COMMIT_ORDER` (the default) causes parallelization information to be logged using commit timestamps; `WRITESET` causes this information to be logged using write sets in such a way that any transactions not updating the same row can be parallelized; and `WRITESET_SESSION` acts in the same fashion as `WRITESET`, except that updates originating with the same session cannot be reordered. The size of the row hash history that is kept in memory for tracking transaction dependencies can be set using `binlog_transaction_dependency_history_size`, also introduced in this release. (WL #9556, WL #10750)

- **Group Replication:** View change events from a group can now be replicated to an external multithreaded slave (MTS) of type `DATABASE`. (Bug #25170698, Bug #84008)

- **Group Replication:** Group Replication now supports `SAVEPOINT` SQL transactions. (WL #9837, WL #10520)

- **JSON:** Added the JSON utility function `JSON_PRETTY()`, which prints an existing `JSON` value, or any string that can successfully be parsed as a JSON document, in a format that can be easily read by humans. Each JSON object member or array value is displayed on a separate line of the output; each child object or array is intended 2 spaces with respect to its parent.

  Examples:

```
mysql> SELECT JSON_PRETTY('123');
+--------------------+
```

```
| JSON_PRETTY('123') |
+--------------------+
| 123                |
+--------------------+

mysql> SELECT JSON_PRETTY("[1,3,5]");
+-----------------------+
| JSON_PRETTY("[1,3,5]") |
+-----------------------+
| [
  1,
  3,
  5
]      |
+-----------------------+

mysql> SELECT JSON_PRETTY('{"a":"10","b":"15","x":"25"}');
+-------------------------------------------+
| JSON_PRETTY('{"a":"10","b":"15","x":"25"}') |
+-------------------------------------------+
| {
  "a": "10",
  "b": "15",
  "x": "25"
}   |
+-------------------------------------------+
```

(WL #9191)

- **JSON:** Previously, when sorting `JSON` values in a query using `ORDER BY`, each value was represented by a part having a fixed length in the sort key, each such part requiring 1K of memory. In many cases this usage was excessive—for example, an integer scalar value requires only a relatively very few bytes, so that the remainder of the 1K (90% or more) was taken up by padding.

  Variable length parts for JSON sort keys have been introduced to address this issue, with each key part now taking up only the space that is needed to store the value. This has the following benefits for performance:

  - Because sort buffer space is used more effectively, filesorts need not flush to disk as early or often, which means that more data can be sorted in memory.
  - Shorter keys can be compared more quickly than longer ones, providing a noticeable improvement in performance for in-memory sorts as well as sorts requiring disk usage.

  (WL #8741)

- The `WEIGHT_STRING()` debugging function no longer supports the `LEVEL` clause. (Bug #25469683, Bug #84723)
- For Developer Studio 12.5, 32-bit builds are now disabled unless the `FORCE_UNSUPPORTED_COMPILER` option is given. (Bug #25267157, Bug #84230)
- Several foreign key constraint checks that were storage engine agnostic were being done inside `InnoDB`. These are now done at the SQL layer. (Bug #25252847)
- Some GIS out-of-bounds checking was simplified. Thanks to Daniel Black for the contribution. (Bug #25202470, Bug #84062)
- The set of compiler flags used for Developer Studio 12.5 was improved. (Bug #25148549, Bug #83955)
- `CMake` now uses `CMAKE_POLICY` of `CMP0022 NEW` rather than `CMP0022 OLD`. (Bug #25090147, Bug #83804)

- If MySQL was configured to build with the `-Wno-error` option, `mysql_config` produced incorrect output for its `--cflags` option.

  The set of compiler options that `mysql_config` and `pkg-config` produce now is determined by whitelisting rather than blacklisting. (Bug #25040566, Bug #22898475, Bug #80662)

- The performance of UTF-8 binary collations was improved. (Bug #24788778, Bug #83247, Bug #25076862)

- The systemd service file for `mysqld` now includes a `Documentation` value in the `[Unit]` section to provide a link to the systemd documentation in the MySQL Reference Manual. (Bug #24735762)

- Unit testing now uses Google Mock 1.8. (Bug #24572381, Bug #82823)

- The unimplemented and nonstandard `WITH CUBE` clause for `GROUP BY` is no longer supported.

  **Note**

  `CUBE` now is a reserved word and cannot be used as an identifier without identifier quoting.

  (Bug #24572048, Bug #82821)

- These outdated build scripts were removed from MySQL source distributions: `BUILD/SETUP.sh`, `BUILD/autorun.sh`, `BUILD/cmake_configure.sh`, `BUILD/compile-dist`, and `cmake/configure.pl`. (Bug #24512933)

- The removal of the `mysql_shutdown()` C API function in MySQL 8.0.0 has been reverted. It is still to be removed in a future MySQL version, but that removal has been deferred. (Bug #24496976, Bug #82681, WL #9014)

- Consistency and maintainability of Debian/Ubuntu packaging maintainer scripts was improved. (Bug #23588977)

- In-place `ALTER TABLE` operations are now possible for column modifications of tables containing generated columns that depend on columns with a `DEFAULT` value if the modified columns are not involved in the generated column expressions. For example, changing the `NULL` property of a separate column can be done in place without a table rebuild. (Bug #22987899, Bug #80832)

- For dumps of the `mysql` system database, `mysqldump` no longer generates DDL statements for the `innodb_index_stats` and `innodb_table_stats` tables. Such statements fail at dump reload time because those tables are not user accessible. (Bug #22655287)

- If `mysqld` is invoked with `--daemonize`, `stdout` and `stderr` are redirected to `/dev/null` if connected to a terminal type device, so that `mysqld` can behave as a true daemon. (Bug #21627629)

- `mysql_secure_installation` is more strict about what it considers valid yes and no responses. (Bug #13344753, Bug #62925)

- MySQL client and utility programs reported their own program-specific version number when invoked with the `--version` option, and the version strings did not follow a consistent format. Now these programs print the MySQL version number (same as `mysqld`) in a consistent format. (Bug #11763201, Bug #55885)

- A new `mysqldump` option, `--network-timeout`, enables large tables to be dumped by setting `max-allowed-packet` to its maximum value and network read and write timeouts to a large value. This option is enabled by default. To disable it, use `--skip-network-timeout`. (Bug #11754493, Bug #46103)

- Added two JSON aggregation functions `JSON_ARRAYAGG()` and `JSON_OBJECTAGG()`. The `JSON_ARRAYAGG()` function takes a column or column expression as an argument, and aggregates the result set as a single `JSON` array, as shown here:

```
mysql> SELECT col FROM t1;
+-------------------------------------+
| col                                 |
+-------------------------------------+
| {"key1": "value1", "key2": "value2"} |
| {"keyA": "valueA", "keyB": "valueB"} |
+-------------------------------------+
2 rows in set (0.00 sec)

mysql> SELECT JSON_ARRAYAGG(col) FROM t1;
+---------------------------------------------------------------------------------+
| JSON_ARRAYAGG(col)                                                              |
+---------------------------------------------------------------------------------+
| [{"key1": "value1", "key2": "value2"}, {"keyA": "valueA", "keyB": "valueB"}] |
+---------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

  The order of the array elements is unspecified.

  `JSON_OBJECTAGG()` takes two columns or expressions which it interprets as a key and a value, respectively; it returns the result as a single `JSON` object, as shown here:

```
mysql> SELECT id, col FROM t1;
+------+-------------------------------------+
| id   | col                                 |
+------+-------------------------------------+
|    1 | {"key1": "value1", "key2": "value2"} |
|    2 | {"keyA": "valueA", "keyB": "valueB"} |
+------+-------------------------------------+
2 rows in set (0.00 sec)

mysql> SELECT JSON_OBJECTAGG(id, col) FROM t1;
+-----------------------------------------------------------------------------------------------+
| JSON_OBJECTAGG(id, col)                                                                       |
+-----------------------------------------------------------------------------------------------+
| {"1": {"key1": "value1", "key2": "value2"}, "2": {"keyA": "valueA", "keyB": "valueB"}} |
+-----------------------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

  A `NULL` key causes an error; duplicate keys are ignored.

  For more information, see Aggregate Functions. (Bug #78117, Bug #21647417, WL #7987)

- MySQL now supports an `IMPORT TABLE` statement that imports `MyISAM` tables based on information contained in `.SDI` (serialized dictionary information) metadata files. This statement is useful for exporting “raw” table files from one server and importing them into another, and provides a faster alternative to dumping tables as a file of SQL statements using `mysqldump` and processing the dump file using `mysql`. For more information, see IMPORT TABLE Statement. (WL #7524)

## Bugs Fixed

- **Incompatible Change:** The stored definition of a view for which an explicit column name list was provided could be invalid. For example, for this statement:

```
CREATE VIEW v1 (name2) AS SELECT 1 AS name1 UNION SELECT 2 ORDER BY name1;
```

  The stored definition looked like this, which is invalid:

```
CREATE VIEW v1 AS SELECT 1 AS name2 UNION SELECT 2 AS 2 ORDER BY name1;
```

Now column names within the stored definition are not replaced by the column name list:

```
CREATE VIEW v1 (name2) AS SELECT 1 AS name1 UNION SELECT 2 AS 2 ORDER BY name1;
```

An incompatibility resulting from this change is that the `CREATE VIEW` statement can no longer be recreated solely from the `INFORMATION_SCHEMA VIEWS` table, because the `VIEW_DEFINITION` value does not show the column name list. Instead, you must also consult the `COLUMNS` table to get the column name list. Alternatively (and more simply), use `SHOW CREATE VIEW`. (Bug #23265335, Bug #81377)

- **InnoDB; Microsoft Windows:** On a MySQL 64-bit build on Windows, a file I/O retry result was misinterpreted due to a missing cast necessary for the correct operation of the retry path, resulting in a failing assertion and operating system error. (Bug #24711351)
- **InnoDB:** In debug builds, a call that initialized latch debugging raised an assertion due to a race condition. (Bug #25700405)
- **InnoDB:** The restriction that required the first undo tablespace to use space ID 1 was removed. The first undo tablespace may now be assigned a space ID other than 1. Space ID values for undo tablespaces are still assigned in a consecutive sequence. (Bug #25551311)
- **InnoDB:** A `DROP TABLE` operation raised an assertion on a server with an `innodb_force_recovery` setting of 5 or 6. `DROP TABLE` is no longer permitted with these `innodb_force_recovery` settings. (Bug #25385590)

  References: This issue is a regression of: Bug #19779113.
- **InnoDB:** Compiling the server without the Performance Schema caused a build failure. (Bug #25348787)
- **InnoDB:** During read-ahead, the wrong page size was used to calculate the tablespace size. (Bug #25330449)
- **InnoDB:** Compiling on Fedora 25 using `DWITH_LZ4=system` resulted in a build failure due to a deprecated `LZ4_COMPRESS_LIMITEDOUTPUT` function. (Bug #25297593)
- **InnoDB:** Disabling macros such as `UNIV_PFS_MUTEX`, `UNIV_PFS_RWLOCK`, and `UNIV_PFS_THREAD` caused compilation errors. (Bug #25251082)
- **InnoDB:** A NULL virtual column field name in a virtual index caused a server exit during a field name comparison that occurs while populating virtual columns affected by a foreign key constraint. (Bug #25222337)
- **InnoDB:** The file handle type name for `InnoDB` file I/O Performance Schema instrumentation was changed from `os_pfs_file_t` to `pfs_os_file_t`. (Bug #25220118)
- **InnoDB:** Transactions with subqueries on `INFORMATION_SCHEMA` tables could block concurrent DDL operations. (Bug #25200952)
- **InnoDB:** During a range comparison, a secondary index field number was passed instead of clustered index field number, eventually causing the retrieval of an incorrect field. (Bug #25175249)
- **InnoDB:** A server exit on restart was caused by missing `my_thread_init()` and `my_thread_exit()` functions for background threads that initialize the `st_my_thread_var` structure. (Bug #25167032)

- **InnoDB:** A `memcached` read operation with a non-default read batch size configuration resulted in a server exit. (Bug #25147515)
- **InnoDB:** The `INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS` table reported `NULL` for a foreign key constraint name (`UNIQUE_CONSTRAINT_NAME`) after restarting the server. (Bug #25126722)
- **InnoDB:** A gap lock was taken unnecessarily during foreign key validation while using the `READ COMMITTED` isolation level. (Bug #25082593)
- **InnoDB:** Loading `InnoDB` tables required more memory in MySQL 5.7 due primarily to the addition of in-memory structure members introduced with temporary table optimizations. The in-memory structure members, only used for temporary tables, are now only allocated as needed. (Bug #25080442)
- **InnoDB:** After a `TRUNCATE TABLE` operation on a table with a `FULLTEXT` index, space size was incorrectly calculated resulting in an invalid read. (Bug #25053705)
- **InnoDB:** A prepared XA transaction was rolled back by a high priority transaction. The high priority transaction should wait if the blocking transaction is in a prepared state. (Bug #25032066)
- **InnoDB:** `InnoDB` passed an invalid argument to `syscall(SYS_futex)`. (Bug #24923840, Bug #83375)
- **InnoDB:** After redo log recovery, the node size of an undo tablespace object could sporadically be zero, eventually causing an error. Additionally, the undo tablespace object was incorrectly placed on the LRU list, and the header page of the undo tablespace could be present the buffer cache when the tablespace object is freed, causing a file-open failure on the first page read. (Bug #24916359)
- **InnoDB:** During a checkpoint, all `MLOG_FILE_NAME` redo log records were written in a single mini-transaction (mtr), causing a log parsing buffer overflow. (Bug #24793413, Bug #83245)
- **InnoDB:** The GCC `mach_parse_compressed` function should load one to five bytes depending on the value of the first byte. Due to a GCC bug, GCC 5 and 6 emit code to load four bytes before the first byte value is checked (GCC Bug #77673). A workaround prevents this behavior. Thanks to Laurynas Biveinis for the patch. (Bug #24707869, Bug #83073)
- **InnoDB:** Code related to tablespace type functions was improved to address issues with naming consistency, function distribution, and function usage. (Bug #24706739, Bug #83092)
- **InnoDB:** A mechanism was added to debug builds to ensure that keys for `InnoDB` Performance Schema instrumentation are registered with Performance Schema. The mechanism causes startup to fail on debug builds if the number of Performance Schema keys does not match the number of registered Performance Schema keys. (Bug #24686908)
- **InnoDB:** A race condition while updating table statistics could result in an estimated row count of 1 and an incorrect query execution plan. (Bug #24666839, Bug #82968)
- **InnoDB:** Due to a `glibc` bug, short-lived detached threads could exit before the caller had returned from `pthread_create()`, causing a server exit. Thanks to Laurynas Biveinis for the patch. (Bug #24605956, Bug #82886)
- **InnoDB:** An error in code related to table statistics raised an assertion in the `dict0stats.cc` source file. (Bug #24585978)
- **InnoDB:** The list of module base names representing modules in the `InnoDB` code base that allocate memory using `ut_malloc` or `ut_new` was incomplete. The list is used by the Performance Schema for reporting of `InnoDB` memory allocation event data. (Bug #24571816)
- **InnoDB:** Some `InnoDB` rw-lock and mutex keys were not registered for use with the Performance Schema. (Bug #24571597)

- **InnoDB:** After increasing the value of `innodb_undo_logs` and restarting the server, the number of active undo tablespaces was not increased when assigning undo tablespaces to newly allocated rollback segments. (Bug #24488141)

- **InnoDB:** The unused `MLOG_UNDO_HDR_DISCARD` redo log record type and related functions were removed. (Bug #24482001)

- **InnoDB:** An assertion was raised when the purge thread started due to the server not recognizing a tablespace ID as an undo tablespace ID. (Bug #24479773)

  References: This issue is a regression of: Bug #23517560.

- **InnoDB:** `InnoDB` incorrectly reported an error about missing encryption when restoring pages from the doublewrite buffer during recovery. (Bug #24471076)

- **InnoDB:** `InnoDB` reported an incorrect estimate for the number of pages in the buffer pool for a table. The number of pages was not decremented correctly when pages were removed. (Bug #24464147)

  References: This issue is a regression of: Bug #21747906.

- **InnoDB:** The following code changes related to undo tablespace management were implemented:

  - An in-memory array for temporary tablespace rollback segments was added, freeing 32 slots in the `TRX_SYS` page for use by redo-enabled rollback segments.

    As a result of this change, you no longer need to consider temporary tablespace rollback segments when setting the `innodb_rollback_segments` configuration option during configuration of separate undo tablespaces.

  - Code comments related to undo logs and rollback segments were revised.

  - Fixed arrays that list rollback segments and undo tablespaces were replaced by dynamic size arrays.

  - The function that initializes undo tablespaces was refactored.

  - A class was added for managing just-in-time creation of undo and undo file names.

  (Bug #24462978)

- **InnoDB:** A cached undo segment was not removed from the rollback segment history during a slow shutdown. (Bug #24450908)

- **InnoDB:** An error during a table-rebuilding operation on a table with only a generated clustered index (`GEN_CLUST_INDEX`) raised and assertion due to an error called with an invalid key name. (Bug #24444831)

- **InnoDB:** MySQL did not build with GCC 6.1.1. (Bug #24438752)

- **InnoDB:** Rotating the tablespace encryption master key while the server is in read-only mode raised an assertion instead of displaying an error message. (Bug #24404091)

- **InnoDB:** On a table without an explicitly defined primary key, `InnoDB` did not replace the implicit clustered index (`GEN_CLUST_INDEX`) when a unique key was defined on a `NOT NULL` column. (Bug #24397406)

- **InnoDB:** `handler::keys_to_use_for_scanning()` was removed from the handler API. The function was no longer used. The `HA_READ_ORDER` index flag provides the same information for each index. (Bug #24364448)

- **InnoDB:** Page cleaner threads asserted due to a regression related to the adaptive hash index feature. (Bug #24346574)

  References: This issue is a regression of: Bug #21407023.

- **InnoDB:** `InnoDB` failed to free memory used by the full-text optimizer thread. (Bug #24331265)

- **InnoDB:** During recovery, `InnoDB` attempted to fetch LOB pages using a null reference. (Bug #23615208)

- **InnoDB:** When adding a new index, the server dropped an internally defined foreign key index and attempted to use a secondary index defined on a virtual generated column as the foreign key index, causing a server exit. `InnoDB` now permits a foreign key constraint to reference a secondary index defined on a virtual generated column. (Bug #23533396)

- **InnoDB:** An `INFORMATION_SCHEMA.FILES` query resulted in a server exit due to a race condition with a concurrent tablespace creation operation. (Bug #23477214, Bug #81614)

- **InnoDB:** A concurrent DML operation during an in-place `ALTER TABLE` operation that rebuilt the table did not update a virtual index, resulting in a mismatch between the virtual index and clustered index. (Bug #23219499)

- **InnoDB:** A `TRUNCATE TABLE` operation held the `dict_sys` mutex while scanning for and removing pages from the buffer pool, causing concurrent DDL operations to stall. The mutex is now released during the scan and acquired again when the scan is completed. (Bug #23070734, Bug #80060)

- **InnoDB:** Inserting GIS data into an R-tree raised an assertion due to a missing page number field that was encountered when storing the B-tree cursor. (Bug #23044098, Bug #80939)

- **InnoDB:** Writing to an `InnoDB` internal temporary table did not increment the `Handler_write` counter. (Bug #23024178)

- **InnoDB:** Changes to the `InnoDB` recovery process in MySQL 5.7 could require up to three scans of the redo log during recovery. To reduce the number of scans, the first and second scans were merged. With this change, there is only one scan unless the redo log record hash table that is populated by the scan reaches its memory threshold. In this case, a second scan is initiated that performs a simultaneous scan and apply. (Bug #22963951, Bug #80788)

- **InnoDB:** A table-copying online `ALTER TABLE` operation on a `ROW_FORMAT=REDUNDANT` table with indexed virtual columns raised an assertion. (Bug #22018745)

- **InnoDB:** After a server restart, concurrent `INSERT` operations a table with an auto-increment primary key resulted in a duplicate entry error. The current auto-increment value was not changed after `auto_increment_increment` and `auto_increment_offset` settings were modified. (Bug #20989615, Bug #76872)

- **InnoDB:** Performance Schema instrumentation for `InnoDB` file I/O was disabled on Windows. (Bug #14025581)

- **InnoDB:** An automatically generated foreign key constraint identifier that exceeded the 64 character limit appeared truncated in the `INFORMATION_SCHEMA.TABLE_CONSTRAINTS CONSTRAINT_NAME` column. Automatically generated foreign key constraint identifiers can no longer exceed the 64 character limit. (Bug #11745347, Bug #13942)

- **InnoDB:** The `row_search_mvcc()` function unnecessarily traversed the entire table for a range query, which occurred when the record was not in the transaction read view. (Bug #84202, Bug #23481444, Bug #25251375)

- **Packaging:** The **my_create_minidump** function in the mysys library called the Windows API function **MiniDumpWriteDump** with an incorrect value (NULL) for the ExceptionParam parameter. This resulted in exception information being omitted from the minidump file. (Bug #24505650, Bug #82695)
- **Partitioning:** For a partitioned table, dropping a partition dropped triggers for the table as well. (Bug #24449174)
- **Partitioning:** Updating a row of a table that had partitioning on a generated column could raise an assertion failure for debug builds, and return incorrect results in nondebug builds. (Bug #22574695, Bug #80080)
- **Replication:** MySQL 8.0.1 adds the original commit timestamp related statements, but `mysqlbinlog` was wrongly reporting that these statements have been present since MySQL 8.0.0 because they were written to the log with the incorrect version number `80000`. The fix changes the version to `80001` in `mysqlbinlog` dumps. (Bug #25710507)
- **Replication:** Lock contention impeded binary relay performance during processing of the relay log file on the slave. (Bug #25321231, Bug #77778)
- **Replication:** A failed node that was removed from the cluster could be expelled upon rejoining the cluster. (Bug #25311008)
- **Replication:** When using a multithreaded slave, applier errors displayed worker ID data that was inconsistent with data externalized in Performance Schema replication tables. (Bug #25231367)
- **Replication:** Compiling MySQL 5.7.17 failed with a variable length array error. (Bug #25163241, Bug #83994)
- **Replication:** In row-based replication, a message that incorrectly displayed field lengths was returned when replicating from a table with a `utf8mb3` column to a table of the same definition where the column was defined with a `utf8mb4` character set. (Bug #25135304, Bug #83918)
- **Replication:** If the binary log on a master server was rotated and a full disk condition occurred on the partition where the binary log file was being stored, the server could stop unexpectedly. The fix adds a check for the existence of the binary log when the dump thread switches to next binary log file. If the binary log is disabled, all binary logs up to the current active log are transmitted to slave and an error is returned to the receiver thread. (Bug #25076007)
- **Replication:** The GTID transaction skipping mechanism that silently skips a GTID transaction that was previously executed did not work properly for XA transactions. (Bug #25041920)
- **Replication:** If a relay log index file named relay log files that did not exist, `RESET SLAVE ALL` sometimes did not fully clean up properly. (Bug #24901077)
- **Replication:** When the MTS slave applier stopped because of an (injected) error, it reported no useful information for troubleshooting. (Bug #24822686)
- **Replication:** `FLUSH BINARY LOGS` could become slow with data replicated from many servers. (Bug #24806259, Bug #83270)
- **Replication:** When using XA transactions, if a lock wait timeout or deadlock occurred for the applier (SQL) thread on a replication slave, the automatic retry did not work. The cause was that while the SQL thread would do a rollback, it would not roll the XA transaction back. This meant that when the transaction was retried, the first event was `XA START` which was invalid as the XA transaction was already in progress, leading to an `XAER_RMFAIL` error. (Bug #24764800)

  References: See also: Bug #83588, Bug #24923091, Bug #24966941.

- **Replication:** A partially failed `CREATE USER`, `RENAME USER`, or `ALTER USER` statement was not correctly consuming an auto-generated or specified GTID when binary logging was disabled. (Bug #24693798)
- **Replication:** `Binlog_sender`, which writes events from the binary log to a packet buffer and then sends the packet to the slave, did not reduce the size of the send buffer as expected. (Bug #24643036)
- **Replication:** The group commit update of GTIDs has been refactored to improve performance on workloads with many small transactions. (Bug #24398760)
- **Replication:** If the `relay_log` option was not specified in a configuration file, the `relay_log_basename` variable was being internally constructed on the fly using `hostname` but the `relay_log_basename` variable was not set. When a slave tried to access this uninitialized variable it resulted in an unexpected halt of the server. (Bug #24352667)
- **Replication:** When using a multithreaded slave (`slave_parallel_workers` greater than 0) the value of `Seconds_Behind_Master` was incorrect when rotating a relay log. (Bug #23532304)
- **Replication:** An `XA PREPARE` statement that failed during the intermediate steps could lead to an inconsistent `XA` transaction state, where `ID` = -1 but the `binlogged` flag was set to `true`. This caused asserts while executing `XA COMMIT` and `XA ROLLBACK` queries. (Bug #22915670)
- **Replication:** The server prevented several replication-related administrative statements from working if the `read_only` system variable was enabled. (Bug #22857926, Bug #25363745, Bug #25326058, Bug #84350, Bug #84437)
- **Replication:** `CHANGE MASTER TO` for a channel that did not exist could raise an assertion. (Bug #22255698)
- **Replication:** The delay specified by the `binlog_group_commit_sync_delay` system variable was applied to too many binary log commit groups. (Bug #21420180)
- **Replication:** The number of generated unwanted fseeks into the binary log file being replicated to a slave has been reduced. (Bug #83226, Bug #24763579)
- **Replication:** The fix for Bug #81657 was not correctly merged into MySQL 8.0. Thanks to Laurynas Biveinis for alerting us. (Bug #83124, Bug #24715790)
- **Replication:** The `rpl.rpl_binlog_errors` test was failing sporadically on Windows. (Bug #82302, Bug #24330138)
- **Replication:** When `binlog_group_commit_sync_delay` was set to a value between 1 and 9, if `binlog_group_commit_sync_no_delay_count` was set to a value greater than 1, and the number of transaction commits was less than `binlog_group_commit_sync_no_delay_count`, these commits hung forever if no more commits were received; and if `binlog_group_commit_sync_no_delay_count` was set to 0, all transaction commits hung forever. (Bug #80652, Bug #22891628)
- **Replication:** Concurrent `CREATE TRIGGER` and `DROP TRIGGER` statements were not being binary logged in the correct order, causing slaves to fail. (Bug #77095, Bug #21114768)
- **Group Replication; Microsoft Windows:** Removed unnecessary warnings raised when the Group Replication plugin was compiled on Windows platforms. (Bug #25119288)
- **Group Replication; Microsoft Windows:** Building Group Replication support on Windows now requires CMake version 2.8.12 or later. (Bug #24964522)

- **Group Replication:** The `MEMBER_STATE` of a group replication member did not change from `ERROR` to `OFFLINE` when the `STOP GROUP_REPLICATION` command was executed when the error was `ER_GROUP_REPLICATION_CONFIGURATION`. (Bug #25674926)
- **Group Replication:** With flow control enabled, reaching a minimum flow control quota of 1 caused Group Replication not to stop throttling when the reason for throttling was no longer in effect. (Bug #25461354)
- **Group Replication:** Using an unresolvable host name in `group_replication_group_seeds` caused `START GROUP_REPLICATION` to fail. This fix ensures that host names in `group_replication_group_seeds` are validated when starting Group Replication and that the list contains at least one valid address. Invalid addresses are ignored. (Bug #25460324, Bug #84674)
- **Group Replication:** The `_gr_user` account created by Group Replication plugin installation was not reliably removed when the plugin was uninstalled. (Bug #25298987)
- **Group Replication:** When starting Group Replication on an offline node, it was possible for the node to be configured for replication, but but be unable to recover. (Bug #25256910)
- **Group Replication:** Group Replication GCS did not discard messages from an inactive member of the group. (Bug #25134074)
- **Group Replication:** An attempt by the Group Replication plugin to contact the server that was no longer possible led to an assertion. (Bug #25071492)
- **Group Replication:** After executing restarts on the group replication applier SQL thread, the plugin could no longer detect failure of the thread. (Bug #24969065)
- **Group Replication:** Enabling the group replication plugin caused the `performance_schema_max_mutex_classes` default value of 200 to be exceeded. As a result, some group replication mutex instruments did not appear in the `performance_schema.setup_instruments` table. (Bug #24746530)
- **Group Replication:** For servers built with yaSSL, using group replication with encrypted connections could result in timeout failures waiting for view delivery. (Bug #23592214)
- **Microsoft Windows:** On Windows, `SHOW TABLES FROM` *`db_name`* hung if *`db_name`* was given in uppercase. (Bug #24800048, Bug #83262)
- **Microsoft Windows:** 32-bit builds are no longer supported on Windows, but `CMake` failed to detect when a 32-built build would be attempted, resulting in compilation errors later. Now `CMake` detects 32-bit build attempts and produces an appropriate error message. (Bug #24487483, Bug #82645)
- **Solaris:** When the `WITH_INNODB_MEMCACHED` `CMake` option is enabled, `memcached` now can be built on Solaris 11 and 12 using the Developer Studio 12.5 compiler. (Bug #24504155, Bug #82692)
- **Solaris:** Library search path handling on Solaris was incorrect. (Bug #24487934, Bug #82646)
- **Solaris:** The minimum required version of Solaris is now Solaris 11 update 3, due to a dependency on system runtime libraries.
- **Solaris:** On Solaris, MySQL is now built with Developer Studio 12.5 instead of `gcc`. The binaries require the Developer Studio C/C++ runtime libraries to be installed. See here for how to install only the libraries:

  ```
  https://docs.oracle.com/cd/E60778_01/html/E60743/gozsu.html
  ```

- **JSON:** When a `JSON` value consisted of a large sub-document wrapped in many levels of JSON arrays, objects, or both, serialization of the `JSON` value sometimes required an excessive amount time to complete. (Bug #23031146)

- **JSON:** When a `NULL` value existed in a `JSON` column, the result from a query using `GROUP_CONCAT()` together with the `ORDER BY` clause was not always correct. (Bug #22992666)

- **JSON:** The internal `rapid_json_handler` used its own data structures to represent a partially-built DOM; these had to be converted into a `Json_dom` graph before returning the result. Now this handler builds the graph directly, which reduces the amount of work required to build it, and thus to parse a JSON document. (Bug #22900110)

- **JSON:** The internal function `Item_func_case::val_json()` did not always set the null value flag as expected when a `CASE` expression evaluated to `NULL`, leading to an assertion in debug builds of the server. (Bug #22887227)

- **JSON:** The `SUM()` function truncated decimal values extracted from JSON documents, producing an integer result. (Bug #84935, Bug #25530204)

- **JSON:** A JSON document that contained a double value slightly greater than the maximum value that can be represented by a double silently replaced it with zero instead of rejecting the value and raising an error. Such values are now handled correctly in MySQL JSON documents.

  The underlying issue was traced to a problem with RapidJSON, which has been reported to that library's developers as Issue #849. (Bug #84891, Bug #25518504)

- **JSON:** The `JSON_SEARCH()` and `JSON_CONTAINS_PATH()` functions did not work when the *`one_or_all`* argument was specified using UTF-16 encoding. For both of these functions, this argument is now converted to `utfmb4` if need be before its value is checked. (Bug #84880, Bug #22516960)

- **JSON:** The `JSON_UNQUOTE()` function did not work with strings that used UTF-16 encoding. Now these strings are converted to `utfmb4` internally before being processed. (Bug #84878, Bug #25516881)

- **JSON:** Updating the same `JSON` column in a single statement could cause incorrect values to be written into the table. This occurred when the second update overwrote the column value with a subset of itself. An example of such a statement is shown here:

  ```
  UPDATE t SET col = JSON_ARRAY(value), col = col->'$[0]';
  ```

  (Bug #84694, Bug #25461627)

- **JSON:** The functions `JSON_QUOTE()` and `JSON_UNQUOTE()` did not work correctly with multibyte character sets such as `utf8mb4`. (Bug #84680, Bug #25455065)

  References: See also: Bug #77234, Bug #21193273.

- **JSON:** The error message for Error 3152 `ER_JSON_USED_AS_KEY` has been changed from `JSON column '%s' cannot be used in key specification` to the less confusing and more accurate `JSON column '%s' supports indexing only via generated columns on a specified JSON path`. (Bug #81364, Bug #23274244)

- **JSON:** Internal tests for MySQL JSON functionality ran out of stack space on some platforms when run against a debug-enabled server. Because timely checks were not made for stack usage, the server did not detect this situation, leading to a server exit.

  The fix for this issue is twofold:

  - Stack overrun checks are now made before attempting to serialize a nested array or object, so that the operation fails gracefully when processing deeply nested JSON documents, rather than causing an exit.

- Serialization of JSON documents has been reorganized so that it requires less use of the stack when compiled without optimization.

(Bug #81083, Bug #23106330, Bug #26399306)

- `BIN()` could produce incorrect truncation with multibyte character sets when used within `UNION`. (Bug #27511490, Bug #89581)
- MySQL did not compile with GCC 7. (Bug #25643811, Bug #26825211)
- The (undocumented) `WINDOWS_RUNTIME_MD` `CMake` option has been removed. (Bug #25611359)
- If `--skip-innodb` or one of its variants was used, a spurious warning about `avoid_temporal_upgrade` was generated. (Bug #25573578)
- `mysqld_safe` failed to restart the server if a *`PID_FILE`*`.shutdown` file was present. (Bug #25572504)

  References: This issue is a regression of: Bug #11751149.
- For Debian/Ubuntu packages, user-defined collation files could be overwritten during MySQL upgrades. Charset files are now marked as conffiles so that user customizations generate a prompt during upgrades whether to overwrite them. (Bug #25525628, Bug #84761)
- For `CREATE TABLE` statements that specified the table name with a database qualifier and included a `DATA DIRECTORY` or `INDEX DIRECTORY` option, an error occurred if there was no default database. (Bug #25514146, Bug #84861)
- `referenced_table_schema` and `referenced_table_name` field values in the `mysql.foreign_keys` data dictionary table were not stored in lowercase when `lower_case_table_names` was enabled. (Bug #25495714)
- Starting the server with `performance_schema_digests_size=1` caused an abnormal exit. (Bug #25492129, Bug #84786)
- For clients linked against yaSSL, connections became invalid when a read timeout occurred, rather than retrying the read. (Bug #25444075)
- MySQL compilation in different directories produced different builds to leakage of absolute paths into debug information and `__FILE__`. (Bug #25436469, Bug #84608, Bug #25859274, Bug #85855)
- Calculations for UCA 9.0.0 collations were inefficient for tailoring rules containing contraction characters. (Bug #25426632, Bug #84577, Bug #25426632, Bug #84577)
- A negative internal connection timeout value caused the connection to go idle and abort. A negative value now causes the connection to block indefinitely in the absence of I/O. (Bug #25408557)
- A mutex Performance Schema name was too long and produced a warning at server startup. (Bug #25406915)
- MySQL failed to compile on some platforms with `-DWITH_LIBWRAP=ON`. `CMake` support now checks whether `tcpd.h` has proper function prototypes. (Bug #25395543, Bug #84495)
- An index defined on a data dictionary table column exceeded the maximum index key length when the instance was initiated with `innodb_page_size=4k`. (Bug #25384527)
- `mysqld_safe` did not check whether the directory named by the `--basedir` option existed. (Bug #25365194)

- Configuring `CMake` with `-G ninja` resulted in build output that was inappropriate for build platforms other than Xcode or Visual Studio. (Bug #25358460)
- `mysqld_safe` failed if the error log file named by the `--log-error` option was a FIFO. (Bug #25356221, Bug #84427)
- For prepared statements, an alias within a subquery or derived table might cause incorrect behavior during statement execution if another alias depended on it. (Bug #25343335, Bug #84398, Bug #25171608)
- `mysqld_safe` could fail if the `--datadir` option value ended with a `/` character. (Bug #25319457)
- A recent change to `mysqld_safe` caused the `mysql.server` script to be unable to start it if the base directory was specified as an absolute path that differed from the compiled-in default absolute path. (Bug #25319392, Bug #84263)
- The `CONNECTION_CONTROL` plugin failed to compile if the Performance Schema was disabled. (Bug #25308357, Bug #84304)
- Passwords did not expire correctly for accounts created using MySQL Workbench. (Bug #25299309)
- For System V init scripts for RPMs, the `[mysqld]` option-file section was being ignored for some options, such as `pid-file`. (Bug #25287707, Bug #84172)
- Init scripts failed to launch `mysqld_safe` if a non-default base directory was used. (Bug #25261472, Bug #84219)
- `CMake` now detects whether a GCC 5.3.0 loop optimization bug occurs and attempts a workaround if so. (Bug #25253540)
- `mysqld_safe --no-defaults` did not work (inadvertent consequence of an earlier bug fix). (Bug #25244898, Bug #84173)
- Semicolon (`;`) characters within or between statements could cause distinct digests to be generated from identical statements. (Bug #25244533, Bug #83253)
- Components could not register services without referring to their private implementation. A `SERVICE_IMPLEMENTATION(`*`component`*`,`*`service`*`)` macro now enables this to be avoided. (Bug #25238906)
- Certain stored functions, if used in a query `WHERE` clause, could be handled using Index Condition Pushdown (which should not happen), resulting in a server exit. (Bug #25196653, Bug #25174454)
- For `ai_ci` collations based on Unicode Collation Algorithm 9.0.0, accented characters that compare equal were treated as different by `LIKE` comparisons. (Bug #25167284, Bug #83999)
- For a client linked against `libmysqlclient`, invalid memory access could occur during use of prepared statements. (Bug #25164932)
- Some Linux startup scripts did not process the `datadir` setting correctly. (Bug #25159791)
- If a character set is specified for a column of a partitioned table, a segmentation fault could occur while upgrading. (Bug #25153261)
- `LOAD DATA` failed to accept multibyte characters that followed an escape sequence. (Bug #25147988, Bug #83950, Bug #25865525)
- The fix for Bug #25088048 caused the command used by `mysqld_safe` to start the MySQL server to no longer include the `mysqld` path. (Bug #25144379)

References: This issue is a regression of: Bug #25088048.

- For UCA collations, `LIKE` comparisons against a pattern that ended with the escape character returned incorrect results. (Bug #25140629, Bug #83930)
- The default character set and collation were used instead of the character set and collation defined in `db.opt` file when upgrading the schema from MySQL 5.7 to MySQL 8.0. (Bug #25139901)
- Instead of updating the data dictionary cache at the end of DDL statements using separate function calls, the data dictionary cache is now updated as part of transaction commit. (Bug #25095798, Bug #83818)
- `CREATE TABLE` now requires the `FILE` privilege if `DATA DIRECTORY` or `INDEX DIRECTORY` is specified explicitly as a table or partition option. `ALTER TABLE` requires the `FILE` privilege if either option is specified explicitly as a partition option (it ignores them if specified as table options). (Bug #25092566)
- There were some differences between ICU order and MySQL Unicode collations for Hungarian contractions and ligatures. MySQL now follows ICU order. (Bug #25090543)
- The data dictionary failed to initialize and start when using a binary collation. The query string generated to create the data dictionary schema did not add quotes to the binary collation name. (Bug #25054104, Bug #83706)
- Executing a stored procedure containing a query that accessed a view could allocate memory that was not freed until the session ended. (Bug #25053286)
- Compilation on FreeBSD 11 failed attempting to check `MAP_NORESERVE`, which is no longer defined. (Bug #25048128, Bug #83689)
- `mysql-test-run.pl` now checks whether the `TSAN_OPTIONS` environment variable is set. If so, the value is taken as the path name of a file containing ThreadSanitizer suppressions (errors to be ignored during test runs). Additionally, the `--sanitize` option now causes `mysql-test-run.pl` to scan the server error logs for ThreadSanizer messages. (Bug #24970905, Bug #83601)
- After starting the server with a nonzero `--lower-case-table-names` setting, an assertion was raised when `USE INFORMATION_SCHEMA;` was the first instruction from the first client. (Bug #24963580)
- For case-insensitive Unicode collations, the various space characters did not hash to the same value, resulting in incorrect comparisons between them. (Bug #24956750, Bug #83549)
- Dictionary clients now track uncommitted dictionary objects that are being modified by DDL statements, making uncommitted changes by the same session visible to a dictionary client without affecting the dictionary object cache prior to commit time. (Bug #24956365, Bug #83548)
- Simultaneous dictionary object cache misses were not handled correctly. An object retrieved by one thread could be evicted from the cache before a waiting thread attempted to access it. (Bug #24949179)
- `CMake` now avoids configuring the `-fexpensive-optimizations` option for GCC versions for which the option triggers faulty shift-or optimizations. (Bug #24947597, Bug #83517)
- `CMake` support was added for compiling using the `-std=c++03` option under Developer Studio 12.5. This is now used rather than `stlport` by default. (Bug #24947136, Bug #83512, Bug #25229424)
- `NCHAR` and `NATIONAL CHAR` are synonyms, but `CAST(`*`expr`*` AS NCHAR)` succeeded, whereas `CAST(`*`expr`*` AS NATIONAL CHAR)` did not. Now both work. (Bug #24934161)
- OEL RPM packages now better detect which platforms have multilib support (for which 32-bit and 64-bit libraries can be installed). Thanks to Alexey Kopytov for the patch. (Bug #24925181, Bug #83457)

- OEL RPM packages now better detect which platforms do not have multilib support (for which 32-bit and 64-bit libraries can be installed). Thanks to Alexey Kopytov for the patch. (Bug #24916428, Bug #83428)
- The `LOCATE()` function returned `NULL` if the *`substr`* or *`str`* argument was `NULL`, but not if the *`pos`* argument was `NULL`. Now it returns `NULL` if any argument is `NULL`. (Bug #24911350, Bug #83427)
- Bit operations could cause a server exit to occur if argument nullability was mishandled. (Bug #24910958, Bug #24930038, Bug #24930829)
- The Block Nested Loop algorithm could allocate too much memory during query execution. (Bug #24909223)
- Information about building MySQL 5.6 compatibility libraries in the MySQL 5.7 and higher `.spec` file is needed only for building `libmysqlclient` and `libmysqld`. Information about building the `InnoDB` memcached plugin was removed. (Bug #24908345, Bug #83409)
- Incorrect updating of view metadata could raise an assertion. (Bug #24834622)
- Initialization of the `keyring_okv` plugin failed if the `STANDBY_SERVER` setting was missing from the `okvclient.ora` configuration file, effectively making this a mandatory setting. `STANDBY_SERVER` is now optional. (Bug #24816271)
- Data dictionary objects acquired by `Dictionary_client::acquire_uncached()` are now owned by the current auto-releaser instead of the caller. Also, acquisition of `TABLE_SHARE` view objects is performed by `acquire() and` clone() instead of `acquire_uncached()`. (Bug #24813358, Bug #83296)
- Privilege checking could be incorrect for a derived table used within a multiple-table `UPDATE` invoked within a stored procedure or view object, for the second or subsequent execution of the object, if the derived table was merged into the outer query. (Bug #24810564)
- The `Created_tmp_tables` status variable was incremented in some cases when no temporary table was created. (Bug #24808970, Bug #83287)
- An in-place upgrade from MySQL 5.7 to MySQL 8.0 failed if parsing of a stored routine body failed while migrating the routine. Now a warning is reported and the routine is created without parsing its body. Also, warnings are now reported when dependency resolution fails for a view during view migration. (Bug #24805140, Bug #83275)
- For compilation, `mysql_upgrade` is dependent on the dynamically generated `sql_commands_system_tables_data_fix.h` file, but a missing dependency could cause that file not to be generated. (Bug #24802377, Bug #83272)
- A race condition between transactions accessing the access-control list (ACL) cache and use of the ACL cache to populate `INFORMATION_SCHEMA` tables could cause a server exit. (Bug #24786029)
- For debug builds, `EXPLAIN` or `DESCRIBE` for a table with a database or table name longer than the maximum permitted length raised an assertion rather than displaying an appropriate error. (Bug #24751177, Bug #83114)
- The implementation of several `INFORMATION_SCHEMA` tables as views on data dictionary tables introduced a number of native SQL functions intended only for internal use by the server, but they could be invoked by users. Those functions now produce an error if invoked by users. (Bug #24749248, Bug #83189)
- `CASE`, `COALESCE()`, `IF()` and `IFNULL()` could merge a mix of signed and unsigned arguments incorrectly and produce an incorrect result type. (Bug #24733658, Bug #83148)

- Connections from a client to a server with SSL enabled succeeded even if `--ssl-mode` had a value of `VERIFY_CA` or `VERIFY_IDENTITY` and the client did not provide a CA certificate. (Bug #24732452, Bug #23189252, Bug #25397416, Bug #84508)
- Manual creation of a directory in the data directory resulted in `USE dir_name` succeeding even though the directory was not registered as a database in the data dictionary. (Bug #24732194, Bug #83140)
- If `InnoDB` statistics were incorrect, `FOUND_ROWS()` could return 1 even when the previous `SELECT` returned no rows. (Bug #24714857, Bug #83110)
- `ALTER TABLE` on a temporary table could raise an assertion if a nontemporary table with the same name existed. (Bug #24713918, Bug #83117)
- `CMake` now sets `-DWITH_NUMA=ON` for Debian platforms where possible. (Bug #24689101)
- To better provide atomic file creation, Debian packaging scripts now use the coreutils `install` command rather than `touch`, `chmod`, and `chown`. (Bug #24688682)
- Enabling the `DISABLE_SHARED` `CMake` option caused compilation failure. (Bug #24687701, Bug #83039)

  References: This issue is a regression of: Bug #24481181.
- The `BIN()`, `OCT()`, and `HEX()` functions could mishandle values of the `BIT` data type. (Bug #24686658, Bug #83031)
- For debug builds, an invalid `utf8` character in the comment of a `CREATE EVENT` or `ALTER EVENT` statement raised an assertion. This now produces an `ER_INVALID_CHARACTER_STRING` error. (Bug #24679962)
- The encoding for the `utf8` character set permitted characters between U+D800 and U+DFFF as valid, though they are reserved for surrogate pairs and do not directly represent characters. They are now considered invalid. (Bug #24672415)
- A query could produce incorrect results if the `WHERE` clause contained a dependent subquery, the table had a secondary index on the columns in the select list followed by the columns in the subquery, and `GROUP BY` or `DISTINCT` permitted the query to use a Loose Index Scan. (Bug #24671968, Bug #83005)
- After a `RENAME TABLE` operation on a table with a foreign key, the generated constraint name was not updated in `TABLE_CONSTRAINTS`. (Bug #24666169)
- On macOS, `CMake` AddressSanitizer support did not work. (Bug #24661626, Bug #82976)

  References: This issue is a regression of: Bug #23759968.
- Compilation failed on macOS 10.11 with Xcode 8.0. (Bug #24661523, Bug #82975)
- Index hints applied to invisible indexes produced no error. (Bug #24660093, Bug #82960)
- In some cases, `INFORMATION_SCHEMA.KEY_COLUMN_USAGE` did not report data for all foreign key constraints. (Bug #24655803, Bug #82961)
- `REPLACE()` on large strings could be slow and unkillable. (Bug #24652792)
- The `DebugPrintTest` and `DebugPrintDeathTest` unit tests did not handle divide-by-zero testing properly on the Aarch64 platform. Thanks to Alexey Kopytov for the patch. (Bug #24624555, Bug #82889)
- Compilation on FreeBSD using GCC 6 did not work. (Bug #24619561, Bug #82922)

- Changes made to `mysqld_safe` in recent MySQL releases require the `--ledir`, `--mysqld`, `--mysqld-version` options to be specified on the command line; they can no longer be specified in option files. This could cause failure of init scripts that invoke `mysqld_safe`. Such scripts now pass the value of the `MYSQLD_OPTS` environment variable as the first command-line argument to `mysqld_safe`, with the value set to such command line-only `mysqld_safe` option values as may be required. On platforms that use systemd, the `MYSQLD_OPTS` value can be set in `/etc/sysconfig/mysqld` with a line such as this:

  ```
  MYSQLD_OPTS=" --ledir=/mysqld_ledir --mysqld=my_wrapper "
  ```

  The value of `MYSQLD_OPTS` can also include `mysqld` options for `mysqld_safe` to pass to `mysqld`. (Bug #24619033, Bug #82920)

  References: This issue is a regression of: Bug #24464380, Bug #24483092, Bug #25088048, Bug #25378439, Bug #25378565.

- `SET PERSIST innodb_buffer_pool_size = value` stored the original value of `innodb_buffer_pool_size` to `mysqld-auto.cnf`, not the new value. (Bug #24613005, Bug #82905)

- For SLES packages, a typo in the installation script postamble prevented some cleanup from occurring. (Bug #24605300, Bug #82389)

- Some messages written by the server to the error log while upgrading the data directory were missing the standard timestamp and process ID information. (Bug #24600054, Bug #82874)

- It was possible for the session values of the `Last_query_cost` and `Last_query_partial_plans` status variables to be accessed before they had been initialized. (Bug #24596263)

- Warnings occurring during `CREATE TABLE ... SELECT` could cause a server exit. (Bug #24595992)

- For `LOAD DATA` used to insert data into an updateable view, the check to verify whether a column is actually updatable was missing. (Bug #24595937)

- The server could dereference a null pointer when a deterministic function returning `LONGTEXT` was used in a subquery. (Bug #24595581)

- A view altered with `ALTER VIEW` might be dropped if the statement failed with an error. (Bug #24594140)

- Conversion of `JSON` documents to string could be slow if the document was large and contained many signed integers. (Bug #24586888)

- The data dictionary can contain entries for temporary tables (names beginning with `#sql`). These tables were exposed to `INFORMATION_SCHEMA` queries and through `SHOW` statements. This could cause `mysqldump` and `mysqlpump` to fail when they attempted to dump such tables. Temporary tables are now hidden to `INFORMATION_SCHEMA` queries and `SHOW` statements. (Bug #24580599, Bug #24571427)

- When a view was defined, the character set of the definition was not considered. If this differed from the default character set, table names used in the view might not be recognized, causing the view to be marked invalid.

  For debug builds, an assertion could be raised if an error occurred while parsing an `ALTER VIEW` statement and the diagnostics area was empty. (Bug #24580586)

- For `LOAD DATA` statements, input data with too many column values produced only a warning, rather than an error as in MySQL 5.6. An error now occurs. (Bug #24577194, Bug #82830)

- Using `SET PERSIST` with the `global_log` system variable was ineffective. (Bug #24569624, Bug #82807)
- In the Performance Schema `variables_info` table, the `VARIABLE_SOURCE` column was not set properly for some boolean options specified in option files without a variable. (Bug #24567960)
- The `.mylogin.cnf` option file is intended for use by client programs, but the server was reading it as well. The server no longer reads it. (Bug #24557925)
- Use of `boost::chrono` was replaced with `std::chrono` because the former was causing link errors. (Bug #24556808, Bug #82781)
- The X Plugin was built with compilation options different from other plugins. (Bug #24555770, Bug #82777)
- When populating the `variables_by_thread` table, the Performance Schema could attempt to access session variables of other threads that were being deinitialized. (Bug #24555658)
- Users without proper privileges could load and unload server components. (Bug #24528148)
- Concurrent execution of `INSTALL COMPONENT` and `UNINSTALL COMPONENT` statements could cause a server exit. (Bug #24527148)
- On Debian/Ubuntu platforms, the systemd startup script for MySQL ignored `datadir` settings in `/etc/mysql/my.cnf`. (Bug #24517024, Bug #82709)
- With a `LOCK TABLES` statement in effect, DML statements on a table that had triggers could cause a server exit. (Bug #24506766)
- Parallel slave threads running account-management statements could fail due to a race condition in handling privilege cache locks. (Bug #24503606)
- Executing `ALTER TABLE` on a table that has triggers concurrently with other DDL operations could result in a corrupted data dictionary. (Bug #24497803)
- If `mysqladmin shutdown` encountered an error determining the server process ID file, it displayed an error message that did not clearly indicate the error was nonfatal. It now indicates that execution continues. (Bug #24496214)
- For the `null_audit` plugin, setting the `null_audit_event_record` system variable improperly could cause a server exit. This variable should be set only from within the `null_audit` plugin, so it is now read only. (Bug #24493829, Bug #82670)
- In certain contexts, an expression such as `IF(col_name > 5000, (1 / col_name), 5000)` could get a type of `DECIMAL(6,4)`, which would truncate results. (Bug #24492965, Bug #82668)
- The data structure used for `ZEROFILL` columns could experience memory corruption, leading eventually to a server exit. (Bug #24489302)
- Operation of the `mysql-multi.server.sh` script was based on `my.cnf` in the data directory. That option file is no longer used, so `mysql-multi.server.sh` has been removed. (Bug #24487870)
- A query could produce incorrect results if `MIN()` or `MAX()` in a subquery referred to an indexed column. (Bug #24484060, Bug #82638, Bug #24657798, Bug #82965)
- `SHOW TRIGGERS` output order could differ on Linux and Windows. (Bug #24482919, Bug #82637)
- `mysqld_safe` attempted to read `my.cnf` in the data directory, although that is no longer a standard option file location. (Bug #24482156)

- After running `mysql_upgrade`, executing an `INSTALL COMPONENT` statement could cause a server exit. (Bug #24453571)
- A regular expression pattern match into a large string could result in a server exit due to memory allocation failure or integer overflow. (Bug #24449076, Bug #24449090)
- Starting the server with a MySQL 5.7 data directory resulted in failure due to absence of the `default_roles` and `role_edges` system tables. (Bug #24447771)
- An incorrect error was reported for `CREATE TABLE` statements with a large value for the `CONNECTION` table option. The value is now limited to 1024 bytes. (Bug #24437124)
- An assertion could be raised if an `ER_LOCK_WAIT_TIMEOUT` error occurred during execution of `SHOW CREATE TRIGGER`. (Bug #24420809, Bug #82483)
- Constant folding could produce incorrect results for large unsigned integers. (Bug #24401273, Bug #82425)
- Use of very long subpartition names could result in a server exit. Now partition or subpartition names larger than 64 characters produce an `ER_TOO_LONG_IDENT` error. (Bug #24400628, Bug #82429)
- The `Gis_wkb_vector<Gis_point>` copy constructor was not explicitly instantiated, causing build problems for the Intel compiler. (Bug #24397833, Bug #82358)
- Privilege escalation was possible by exploiting the way `REPAIR TABLE` used temporary files. (Bug #24388746)
- A race condition between `UNINSTALL PLUGIN` and `SHOW PLUGINS` could result in a server exit. (Bug #24344026)
- With `ROW` mode binary logging, component installation raised an assertion. (Bug #24343582)
- A potential memory leak related to roles was fixed. (Bug #24337928)
- Subqueries that were converted to semijoins and programmatically generated an `INFORMATION_SCHEMA` table could incorrectly treat the `INFORMATION_SCHEMA` table as empty.

  A workaround for this problem prior to the bug fix: `SET optimizer_switch='semijoin=off';` (Bug #24287772, Bug #82214)
- `mysqldumpslow` failed to parse timestamps in the slow query log; it had not been updated to track a change in log timestamp format. (Bug #24007040)
- The AppArmor profile installed by Ubuntu packages was missing an entry permitting `libnuma` to read a `/sys` hierarchy path, resulting in server startup failure. (Bug #23854929)
- The optimizer could incorrectly treat `RAND()` as a constant for queries subjected to semijoin transformation. (Bug #23854015)
- For debug builds, unequal-length binary operands for bit operators could raise an assertion. (Bug #23853628)
- If a transaction rollback request rolled back only a statement and not the whole transaction, any attempt at attaching another operation to the transaction raised an assertion. (Bug #23753319, Bug #82143)
- `STRCMP()` for arguments with the `utf8mb4_unicode_ci` collation could return results different from `ORDER BY`. `STRCMP()` has been corrected. (Bug #23752284, Bug #82132)
- Complete logical backups made with `mysqlpump` could not be restored if GTIDs were enabled.

To enable control over GTID information written to the dump file, `mysqlpump` now has a `--set-gtid-purged` option that indicates whether to add a `SET @@GLOBAL.gtid_purged` statement to the output. (Bug #23748432)

- Infinite recursion could occur if the `audit_log` plugin signalled an error while handling an error. (Bug #23717558, Bug #82052)
- Messages written by the `audit_log` plugin to the error log regarding `MYSQL_AUDIT_CONNECT` event failures now print the underlying error cause as well to aid debugging. (Bug #23710632)
- MySQL now uses `readdir()` rather than `readdir_r()`. The latter has been deprecated since `glibc` 2.24 and caused debug builds of MySQL and builds using GCC 6.1 to fail.

  Additionally, several problems resulting in GCC 6.1 compiler warnings were corrected. (Bug #23708395, Bug #24437737, Bug #82515, Bug #24459890, Bug #82583, Bug #25103242)
- `FORCE INDEX` was ineffective for `SELECT COUNT(*)` queries. (Bug #23596760, Bug #81854)
- During startup, the server creates a lock file for the Unix socket file (for example, `mysql.sock.lock` as a lock file for `mysql.sock`). If the server failed to write the process ID to the lock file, it failed to remove that file, which could cause subsequent server startups to fail until the file was removed manually. (Bug #23582603, Bug #81838)
- For debug builds, queries executed using `Item_func_spatial_collection::val_str()` could raise an assertion. (Bug #23573720)
- For audit log events in the connection class, the `connection_type` value was available only for connect events. The value is now available in connect, disconnect, and change-user events. (Bug #23541550)
- The `audit_log` plugin `audit_log_filter_remove_filter()` function caused a server exit if given a `NULL` argument. (Bug #23522793)
- Attempts to configure MySQL without the `FEDERATED` storage engine failed (the `CMake -DWITH_FEDERATED_STORAGE_ENGINE=0` option did not work). (Bug #23508203, Bug #81665)
- On Solaris, `gettimeofday()` could return an invalid value and cause a server shutdown. (Bug #23499695)
- The `keyring_file` plugin could attempt to write keys to its storage file when the file did not exist. To ensure that keys are flushed only when the correct storage file exists, `keyring_file` now stores a SHA-256 checksum of the keyring in the file. Before updating the file, the plugin verifies that it contains the expected checksum. (Bug #23498254)
- A union query resulting in tuples larger than `max_join_size` could result in a server exit. (Bug #23303485)
- Grant tables with incorrect structure may cause problems in user management operations. As a consequence of the fix for this, for any operation that modifies a grant table, the server now checks whether the table has the expected structure and produces an error if not. `mysql_upgrade` must be run to update the tables to the expected structure. (Bug #23295423, Bug #25095876, Bug #25448037)
- `ST_ExteriorRing()` could cause a server exit due to being passed an invalid WKB string believed to be valid. (Bug #23280574)
- For debug builds, failure to prepare a branch of an XA transaction could lead to a server exit. (Bug #23264552, Bug #81375)

- The optimizer could choose `ref` access on a secondary index rather than `range` access on the primary key, even when the cost was higher. (Bug #23259872, Bug #81341)
- An instance of the disk-full error message contained the wrong error code. (Bug #23247332, Bug #81346)
- For a query with `ORDER BY` and `LIMIT`, an optimizer trace did not record the optimizer's switch to a different index. (Bug #23227428, Bug #81250)
- Improper handling of a lock used by the `version_tokens` plugin and functions could result in a server exit if one of the functions was called while `version_tokens` was being uninstalled. (Bug #23210850)
- Certain errors in DML statements executed within stored programs could be mishandled and result in a server exit. (Bug #23209989)
- The `QUOTE()` function could allocate excessive memory. A limit of `max_allowed_packet` bytes is now imposed and returns `NULL` with a warning for attempts to allocate more. (Bug #23195404)
- For some deeply nested expressions, the optimizer failed to detect stack overflow, resulting in a server exit. (Bug #23135667)
- For sessions created through the X Plugin, incorrect thread attachment/detachment could cause a server exit. (Bug #23057045)
- When attempting to locate the data directory, `mysqld_safe` incorrectly considered `$MY_BASEDIR_VERSION/var` as one of the possible locations. (Bug #23013510, Bug #80866)
- For some generated columns, character set conversion of the column definition for table rebuilds could change column values. (Bug #22991924)
- The OS X DMG installer did not properly set up keyring plugin installation. (Bug #22991650)
- If a query performed a `GROUP BY` on a column of a derived table and the select list contained an expression mixing an aggregate function and the group column, an error was raised if the `ONLY_FULL_GROUP_BY` SQL mode was enabled. (Bug #22924183, Bug #80726)
- A binary (in-place) upgrade from MySQL 5.6 to 5.7 followed by a data export performed using `mysqlpump` resulted in an `Invalid default value for` *`date_column`* error for attempts to reload the dump file. (Bug #22919028, Bug #80706)
- A failed cast of a long float to integer could cause a server exit. (Bug #22907691)
- On Unix and Unix-like systems, the error log file resulting from specifying `--log-error` without an option value was incorrectly created in the directory of the PID file if the `--pid-file` option was also given. (Bug #22900354)
- The `main.log_tables-big` test case could be unstable on highly loaded hosts. Thanks to Laurynas Biveinis for the patch. (Bug #22874167, Bug #80607)
- SQL statements executed through the X Plugin were not instrumented in the Performance Schema. (Bug #22859462)
- The `rpl.rpl_key_rotation` test case did not synchronize properly with the master server. Thanks to Laurynas Biveinis for the patch. (Bug #22838596, Bug #80531)
- `DROP INDEX` operations could fail due to inconsistent handling of index prefix lengths for `TEXT`-type columns (`TINYTEXT` and so forth).

  A consequence of this fix is more restrictive behavior for `CREATE TABLE` and `CREATE INDEX` statements for which a specified index prefix exceeds the maximum column data type size:

  - For a nonunique index, either an error occurs (if strict SQL mode is enabled), or the index length is reduced to lie within the maximum column data type size and a warning is produced (if strict mode is not enabled).

  - For a unique index, an error occurs regardless of SQL mode because reducing the index length might enable insertion of nonunique entries that do not meet the specified uniqueness requirement.

  (Bug #22740093, Bug #80392)

- Metadata locking on stored routine names was performed on a case-sensitive basis, but routine names are not case sensitive. (Bug #22700385)

- Queries that used an aggregate function with `DISTINCT` could produce incorrect results. (Bug #22686994, Bug #80310)

- The `innodb_numa_interleave` system variable was erroneously available on some systems that were not NUMA-enabled. Thanks to Tomislav Plavcic for the patch.

  `CMake` now sets the default `WITH_NUMA` value based on whether the current platform has `NUMA` support. For platforms without NUMA support, `CMake` behaves as follows:

  - With no NUMA option (the normal case), `CMake` continues normally, producing only this warning: NUMA library missing or required version not available

  - With `-DWITH_NUMA=ON`, `CMake` aborts with this error: NUMA library missing or required version not available

  (Bug #22678436, Bug #80288)

- Certain SQL queries involving complex `WHERE` conditions could cause warnings, memory corruption, or a server exit. (Bug #22671573)

- When taking the server offline, a race condition within the Performance Schema could lead to a server exit. (Bug #22551677)

- On macOS, if a table with an associated trigger was renamed to a new name containing both lowercase and uppercase characters, `DROP TRIGGER` for the trigger resulted in an `ER_NO_SUCH_TABLE` error for the table. (Bug #22512899, Bug #79873)

- In the `MYSQL_FIELD` C API structure, the `org_table` value for derived tables was `*`, which could cause failure for queries that depend on this value. The `org_table` value for views and derived tables now is set as follows: If the column is selected from a view, `org_table` names the view. If the column is selected from a derived table, `org_table` names the base table. If a derived table wraps a view, `org_table` still names the base table. If the column is an expression, `org_table` is the empty string. (Bug #22364401, Bug #79641)

- For RPM packages, the default `error-log` location in the deployed `/etc/my.cnf` file differed from the location in the installed `logrotate` script, causing `logrotate` to fail. (Bug #22322685)

- The Performance Schema `events_statements_summary_by_digest` table could contain multiple rows for the same statement digest and schema combination, rather than the expected single (unique) row. (Bug #22320066, Bug #79533)

- For Performance Schema system and status variable tables, variable values expressed in a character set different from `utf8` could be truncated or incorrect. (Bug #22313205)

- For debug builds, altering a table partitioning expression using an expression attribute with an invalid UTF-8 name caused an assertion to be raised. (Bug #22152229)
- After performing inserts in a table containing an `AUTO_INCREMENT` column and then performing a `SELECT` operation, the `LAST_INSERT_ID()` returns the correct value, but the value of the `mysql_insert_id()` C API function was being reset to 0. (Bug #22028117, Bug #78778)
- On Ubuntu, error messages were displayed during upgrades from Community to Commercial packages that made it appear as though `mysqld` and `my_print_defaults` had not been installed. Those messages were spurious and have been silenced. (Bug #21807248)
- With the `use_index_extensions` flag of the `optimizer_switch` system variable disabled, some `SELECT DISTINCT` queries could return incorrect results. (Bug #21749123, Bug #78244)
- An invalid string value in the `WHERE` clause of an `UPDATE` statement, caused an index scan rather than a range scan to be used. For values not present in the index, this could be much slower. Now the optimizer determines this to be an “impossible `WHERE`” condition. (Bug #21032418, Bug #76933)
- Debian packages were missing an AppArmor-related include file and incorrectly were marked dependent on AppArmor (making it impossible to disable AppArmor by uninstalling it). (Bug #20768958)
- When the `automatic_sp_privileges` system variable was enabled, it did not have the expected effect for anonymous users. (Bug #20266641)
- In a replication environment, `SET PASSWORD` or `ALTER USER` could fail to execute on the slave due to failure to parse the hash string correctly. (Bug #20228478)
- The optimizer `resolve_const_item()` function called `Item_decimal()` with the last two arguments in the wrong order. (Bug #19062566)
- An in-place `ALTER TABLE` operation failed to report an error when adding a `DATE` or `DATETIME` column under these conditions: a) the column was `NOT NULL` and no default value was supplied; b) strict and `NO_ZERO_DATE` SQL modes were enabled; c) the table was not empty.

  An `ALTER TABLE` operation failed with an error rather than a warning when adding a `DATE` or `DATETIME` column under these conditions: a) the column was `NOT NULL` and no default value was supplied; b) strict SQL mode was enabled and `NO_ZERO_DATE` SQL mode was not enabled; c) the table was not empty. (Bug #16888677)
- Inserting a `TIME`, `DATE`, or `TIMESTAMP` value with a fractional seconds part into a column having the same type but fewer fractional digits resulted in rounding. This differs from MySQL 5.5, which used truncation rather than rounding. To enable control over this behavior, a new `TIME_TRUNCATE_FRACTIONAL` SQL mode is available. The default is to use rounding. If this mode is enabled, truncation occurs instead. (Bug #16583910, Bug #68760)
- `NOT IN` subqueries could produce incorrect results when an index prefix of an inner table was used. (Bug #13915291)
- On non-Linux Unix systems, the `mysql.server` startup script used the Linux command `pidof` rather than `pgrep`. (Bug #13788154, Bug #64342)
- Starting multiple instances of `mysqld_safe` after an abnormal server exit could result in one `mysqld_safe` instance killing another. As a consequence of the bug fix, the `mysqld_safe.pid` file is no longer used. (Bug #11751149, Bug #41908)
- The `--help` message for `mysqld_safe` was corrected to mention that the `--no-defaults`, `--defaults-file`, and `--defaults-extra-file` options, if given, must be the first argument. (Bug #11745176, Bug #11192)

- The bounds check for the XML parser position stack for each level (which has a fixed depth) used the size of the array as the upper limit, and so was off by one. This is fixed by decreasing the allowable depth by one, which actually matches the maximum number of elements in the position stack. (Bug #83871, Bug #25111907)

  References: See also: Bug #14040071, Bug #15948580.

# Changes in MySQL 8.0.0 (2016-09-12, Development Milestone)

For general information about upgrades, downgrades, platform support, etc., please visit https://docs.oracle.com/cd/E17952_01/mysql-8.0-relnotes-en/.

**Note**

This is a milestone release, for use at your own risk. Upgrades between milestone releases (or from a milestone release to a GA release) are not supported. Significant development changes take place in milestone releases and you may encounter compatibility issues, such as data format changes that require attention in addition to the usual procedure of running `mysql_upgrade`. For example, you may find it necessary to dump your data with `mysqldump` before the upgrade and reload it afterward. (Making a backup before the upgrade is a prudent precaution in any case.)


- Account Management Notes
- C API Notes
- Character Set Support
- Compilation Notes
- Component Notes
- Configuration Notes
- Data Dictionary Notes
- Data Type Notes
- Deprecation and Removal Notes
- Doxygen Notes
- Optimizer Notes
- Packaging Notes
- Parser Notes
- Performance Schema Notes
- Security Notes
- Spatial Data Support
- Test Suite Notes

- X Plugin Notes
- Functionality Added or Changed
- Bugs Fixed

## Account Management Notes

- **Incompatible Change:** The grant tables in the `mysql` system database are now `InnoDB` (transactional) tables. Previously, these were `MyISAM` (nontransactional) tables. This change applies to these tables: `user`, `db`, `tables_priv`, `columns_priv`, `procs_priv`, `proxies_priv`.

  The change of grant table storage engine underlies an accompanying change to the behavior of account-management statements. Previously, an account-management statement that named multiple users could succeed for some users and fail for others. Now, each statement is transactional and either succeeds for all named users or rolls back and has no effect if any error occurs. The statement is written to the binary log if it succeeds, but not if it fails; in that case, rollback occurs and no changes are made. The preceding behavior applies to these statements: `ALTER USER`, `CREATE ROLE`, `CREATE USER`, `DROP ROLE`, `DROP USER`, `GRANT`, `RENAME USER`, `REVOKE`. (`SET PASSWORD` is not listed because it applies to at most one user and is effectively transactional already.) A side effect of this change in behavior is that partially completed account management statements on a MySQL 5.7 master fail when replicated on a MySQL 8.0 slave. For more information, see Atomic Data Definition Statement Support.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `mysql` system database.

  **Note**

  If MySQL is upgraded from an older version but the grant tables have not been upgraded from `MyISAM` to `InnoDB`, the server considers them read only and account-management statements produce an error.

  Due to the change of storage engine from `MyISAM` to `InnoDB`, `SELECT` without `ORDER BY` on grant tables can produce different row orders than previously. If a query result must have specific row ordering characteristics, include an `ORDER BY` clause. (WL #7158, WL #9045)

- MySQL now supports roles, which are named collections of privileges. Roles enable assignment of sets of privileges to accounts and provide a convenient alternative to granting individual privileges, both for conceptualizing desired privilege assignments and implementing them:

  - Roles can be created and dropped.
  - Roles can have privileges granted to and revoked from them.
  - Roles can be granted to and revoked from user accounts.
  - The active roles for an account can be selected from among those granted to the account, and can be changed during sessions for that account.

  For more information, see Using Roles.

  **Note**

  `ROLE` now is a reserved word and cannot be used as an identifier without identifier quoting.

  (WL #988)

## C API Notes

- The `libmysqlclient` shared library major version number is increased from 20 (used in MySQL 5.7) to 21 for MySQL 8.0. (Bug #77600, Bug #21363863)

## Character Set Support

- The `utf8mb4` Unicode character set has a new general collation named `utf8mb4_0900_ai_ci`. `utf8mb4` also has several new language-specific collations with characteristics similar to `utf8mb4_0900_ai_ci` except that language-specific rules take precedence where applicable. The language-specific collations are indicated by ISO 639-1 language codes in the collation name, as shown in the following table. In two cases the language code has an additional item that denotes a variant (German phone book order, Traditional Spanish).

**Table 5 utf8mb4 UCA 9.0.0 language-specific collations**

| Language | Collation |
|---|---|
| Croatian | `utf8mb4_hr_0900_ai_ci` |
| Czech | `utf8mb4_cs_0900_ai_ci` |
| Danish | `utf8mb4_da_0900_ai_ci` |
| Esperanto | `utf8mb4_eo_0900_ai_ci` |
| Estonian | `utf8mb4_et_0900_ai_ci` |
| German phone book order | `utf8mb4_de_pb_0900_ai_ci` |
| Hungarian | `utf8mb4_hu_0900_ai_ci` |
| Icelandic | `utf8mb4_is_0900_ai_ci` |
| Latvian | `utf8mb4_lv_0900_ai_ci` |
| Lithuanian | `utf8mb4_lt_0900_ai_ci` |
| Polish | `utf8mb4_pl_0900_ai_ci` |
| Classical Latin | `utf8mb4_la_0900_ai_ci` |
| Romanian | `utf8mb4_ro_0900_ai_ci` |
| Slovak | `utf8mb4_sk_0900_ai_ci` |
| Slovenian | `utf8mb4_sl_0900_ai_ci` |
| Modern Spanish | `utf8mb4_es_0900_ai_ci` |
| Traditional Spanish | `utf8mb4_es_trad_0900_ai_ci` |
| Swedish | `utf8mb4_sv_0900_ai_ci` |
| Turkish | `utf8mb4_tr_0900_ai_ci` |
| Vietnamese | `utf8mb4_vi_0900_ai_ci` |

`utf8mb4_0900_ai_ci` also works as an accent- insensitive, case-insensitive collation for the languages in the following table.

**Table 6 Languages for which utf8mb4_0900_ai_ci is suitable**

| Language Name | Language Code |
|---|---|
| German (dictionary order) | de |
| English | en |

| Language Name | Language Code |
|---|---|
| Canadian French (locale fr_CA) | fr |
| Irish Gaelic | ga |
| Indonesian | id |
| Italian | it |
| Luxembourgian | lb |
| Malay | ms |
| Dutch | nl |
| Portuguese | pt |
| Swahili | sw |
| Zulu | zu |

`utf8mb4_da_0900_ai_ci` also works as an accent-insensitive, case-insensitive collation for the languages in the following table.

**Table 7 Languages for Which utf8mb4_da_0900_ai_ci is Suitable**

| Language Name | Language Code |
|---|---|
| Norwegian | no |
| Norwegian Bokmål | nb |
| Norwegian Nynorsk | nn |

The nonlanguage-specific `utf8mb4_0900_ai_ci` and language-specific `utf8mb4_LANG_0900_ai_ci` Unicode collations each have these characteristics:

- The collation is based on Unicode Collation Algorithm (UCA) 9.0.0 and Common Locale Data Repository (CLDR) v30, is accent insensitive, and case insensitive. These characteristics are indicated by `_0900`, `_ai`, and `_ci` in the collation name. Exception: `utf8mb4_la_0900_ai_ci` is not based on CLDR because Classical Latin is not defined in CLDR.
- The collation works for all characters in the range [U+0, U+10FFFF].
- If the collation is not language specific, it sorts all characters, including supplemental characters, in default order (described following). If the collation is language specific, it sorts characters of the language correctly according to language-specific rules, and characters not in the language in default order.
- By default, the collation sorts characters having a code point listed in the DUCET table (Default Unicode Collation Element Table) according to the weight value assigned in the table. The collation sorts characters not having a code point listed in the DUCET table using their implicit weight value, which is constructed according to the UCA.
- For non-language-specific collations, characters in contraction sequences are treated as separate characters. For language-specific collations, contractions might change character sorting order.

For more information, see Unicode Character Sets. (WL #9125, WL #9108, WL #9479)

## Compilation Notes

- **Microsoft Windows:** For building MySQL on Windows, the toolchain now prefers 64-bit tools when possible (previously 32-bit). This speeds up linking and avoids issues related to limited address space with the 32-bit linker. (Bug #80675, Bug #22900585)
- `CMake` now causes the build process to link with the GNU `gold` linker if it is available and not explicitly disabled. To disable use of this linker, specify the `-DUSE_LD_GOLD=OFF` `CMake` option. (Bug #23759968, Bug #82163)
- The `WITH_EXTRA_CHARSETS` `CMake` option has been removed. MySQL builds are configured with all character sets by default now. Users who want fewer character sets can edit `cmake/character_sets.cmake` directly and recompile the server. (Bug #80005, Bug #22552125)
- The minimum version of the Boost library for server builds is now 1.60.0. (Bug #79380, Bug #22253921)
- Work was done to clean up the source code base, including: Removing unneeded `CMake` checks; removing unused macros from source files; reorganizing header files to reduce the number of dependencies and make them more modular, removing function declarations without definitions, replacing locally written functions with equivalent functions from industry-standard libraries.
- MySQL source code now permits and uses C++11 features. To enable a good level of C++11 support across all supported platforms, the following minimum compiler versions now apply:
  - GCC: 4.8 or higher
  - Clang: 3.4 or higher (Xcode 7 on OS X)
  - Solaris Studio: 12.4 or higher (Solaris client build only)
  - Visual Studio: 2015
  - CMake: On Windows, the required Visual Studio version results in a required CMake version of 3.2.3 or higher

  On Solaris, the `stlport` library is no longer used. This makes the `SUNPRO_CXX_LIBRARY` `CMake` option obsolete, so it has been removed. (WL #8896)

## Component Notes

- MySQL Server now includes a component-based infrastructure for improving server extensibility:
  - A component provides services that are available to the server and other components. (With respect to service use, the server is a component, equal to other components.) Components interact with each other only through the services they provide.
  - The `INSTALL COMPONENT` and `UNINSTALL COMPONENT` statements provide an SQL interface for component manipulation at runtime.
  - A loader service registers installed components in the `mysql.component` system table, and installs registered components during the startup sequence for subsequent server restarts.

  For general information about the component infrastructure and its SQL-level interface, see MySQL Components. For information about the internal implementation of components, see the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html. (WL #4102)

## Configuration Notes

- **Incompatible Change; InnoDB:** Previously, enabling the `innodb_read_only` system variable prevented creating and dropping tables only for the `InnoDB` storage. As of MySQL 8.0. enabling `innodb_read_only` prevents these operations for all storage engines. Table creation and drop operations modify data dictionary tables in the `mysql` system database, but those tables use the `InnoDB` storage engine and cannot be modified when `innodb_read_only` is enabled. The same principle applies to other table operations that require modifying data dictionary tables, and to operations that modify other tables in the `mysql` database that use the `InnoDB` storage engine, such as the grant tables and the `func` and `plugin` tables. (Bug #21611899)

- The hardcoded memory page size of 8KB for the memory-mapped transaction coordinator was too small for platforms such as ARM64 and PowerPC where the page size is much larger. The server now invokes a system call to get the page size of the current platform rather than using a hardcoded value. A consequence for the `--log-tc-size` option is that the minimum and default values are now 6 times the page size. Also, the value must be a multiple of the page size. Thanks to Alexey Kopytov for the patch. (Bug #23014086, Bug #80818, Bug #26931470, Bug #87995)

- MySQL now supports a `SET PERSIST` variant of `SET` statement syntax, for making configuration changes at runtime that also persist across server restarts. Like `SET GLOBAL`, `SET PERSIST` is permitted for any global system variable that is dynamic (settable at runtime). The statement changes the runtime variable value, but also writes the variable setting to an option file named `mysqld-auto.cnf` in the data directory. At startup, the server processes this file after all other option files. For more information, see Persisted System Variables.

  To provide information showing how each system variable was most recently set, the Performance Schema now has a `variables_info` table that lists each system variable and the source from which it got its value. See Performance Schema variables_info Table.

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate this change into the Performance Schema. (WL #8688)

## Data Dictionary Notes

- **Incompatible Change:** MySQL Server now incorporates a global data dictionary containing information about database objects in transactional tables. In previous MySQL releases, dictionary data was stored in metadata files and nontransactional system tables.

  **Important**

  A data dictionary-enabled server entails some general operational differences compared to a server that does not have a data dictionary; see Data Dictionary Usage Differences. Also, for upgrades to MySQL 8.0, the upgrade procedure differs somewhat from previous MySQL releases and requires that you verify the upgrade readiness of your installation by checking specific prerequisites. For more information, see Upgrading MySQL, particularly Preparing Your Installation for Upgrade.

  `InnoDB` continues to use its own data dictionary in the MySQL 8.0.0 release.

  The following list briefly describes the main implications of this change:

  - The `.frm` metadata files previously associated with base tables and views no longer exist. Metadata previously stored in `.frm` files is now stored in data dictionary tables.

Similarly, trigger metadata previously stored in `.TRG` and `.TRN` files is stored in a data dictionary table and those files no longer exist.

- With the removal of `.frm` files, the 64KB table definition size limit imposed by the `.frm` file structure is removed.
- With the removal of `.frm` files, the `INFORMATION_SCHEMA.TABLES VERSION` field now reports a hardcoded value of `10`, which is the last `.frm` file version used in MySQL 5.7.
- With the removal of `.frm` files, the `sync_frm` system variable is removed.
- A new dictionary object cache that serves the MySQL data dictionary stores previously accessed data dictionary objects in memory to enable object reuse and minimize disk I/O. An LRU-based eviction strategy is used to evict least recently used objects from memory. The cache comprises several partitions that store different object types. For more information, see Dictionary Object Cache.
- New internal data dictionary APIs enable the server, internal storage engines, and plugins to access and store data in the MySQL data dictionary. Internal data dictionary APIs are introduced for handling of schemas, tablespaces, tablespace files, tables, partitioned tables, table partition data, triggers, stored routines, events, table objects, views, character sets, and collations.

  With this change, data dictionary updates and binary log writes for `CREATE TRIGGER` and `DROP TRIGGER` operations are combined into a single, atomic transaction.
- Data dictionary tables are invisible, but in most cases there are corresponding `INFORMATION_SCHEMA` tables that can be queried instead. This enables the underlying data dictionary tables to be changed as server development proceeds, while maintaining a stable `INFORMATION_SCHEMA` interface for application use.

  Some `INFORMATION_SCHEMA` tables have been reimplemented entirely as views on data dictionary tables:

```
CHARACTER_SETS
COLLATIONS
COLLATION_CHARACTER_SET_APPLICABILITY
COLUMNS
KEY_COLUMN_USAGE
SCHEMATA
STATISTICS
TABLES
TABLE_CONSTRAINTS
VIEWS
```

  Queries on those tables are now more efficient because they obtain information from data dictionary tables rather than by other, slower means. In particular, for each `INFORMATION_SCHEMA` table that is a view on data dictionary tables:

  - The server no longer must create a temporary table for each query of the `INFORMATION_SCHEMA` table.
  - When the underlying data dictionary tables store values previously obtained by directory scans (for example, to enumerate database names or table names within databases) or file-opening operations (for example, to read information from `.frm` files), `INFORMATION_SCHEMA` queries for those values now use table lookups instead. (Additionally, even for a non-view `INFORMATION_SCHEMA` table, values such as database and table names are retrieved by lookups from the data dictionary and do not require directory or file scans.)

- Indexes on the underlying data dictionary tables permit the optimizer to construct efficient query execution plans, something not true for the previous implementation that processed the `INFORMATION_SCHEMA` table using a temporary table per query.

The preceding improvements also apply to `SHOW` statements that display information corresponding to the `INFORMATION_SCHEMA` tables that are views on data dictionary tables. For example, `SHOW DATABASES` displays the same information as the `SCHEMATA` table.

For `INFORMATION_SCHEMA` queries that retrieve table statistics, the server now can use statistics cached in `INFORMATION_SCHEMA` tables, or obtain the latest statistics directly from storage engines. The `information_schema_stats` system variable controls which statistics source the server uses.

- When `information_schema_stats` is `CACHED` (the default), the server uses cached statistics stored in the `STATISTICS` and `TABLES` tables.
- When `information_schema_stats` is `LATEST`, the server obtains statistics directly from storage engines. In this case, the server treats queries on `STATISTICS` and `TABLES` as queries for the latest statistics stored in the `STATISTICS_DYNAMIC` and `TABLES_DYNAMIC` tables.

  Affected `INFORMATION_SCHEMA` table statistic columns include:

```
STATISTICS.CARDINALITY
TABLES.AUTO_INCREMENT
TABLES.AVG_ROW_LENGTH
TABLES.CHECKSUM
TABLES.CHECK_TIME
TABLES.CREATE_TIME
TABLES.DATA_FREE
TABLES.DATA_LENGTH
TABLES.INDEX_LENGTH
TABLES.MAX_DATA_LENGTH
TABLES.TABLE_ROWS
TABLES.UPDATE_TIME
```

For more information, see Optimizing INFORMATION_SCHEMA Queries.

- The `foreign_keys` and `foreign_key_column_usage` tables now store foreign key information. The standard SQL way to obtain foreign key information is by using the `INFORMATION_SCHEMA REFERENTIAL_CONSTRAINTS` and `KEY_COLUMN_USAGE` tables; these tables are now implemented as views on the `foreign_keys`, `foreign_key_column_usage`, and other data dictionary tables.

  For some foreign key errors, the server now produces more appropriate and more informative error messages.

  **Note**

  **Incompatibility:** Previously, MySQL supported foreign key names longer than 64 characters. Foreign key names as stored in the `foreign_keys` and `foreign_key_column_usage` tables are a maximum of 64 characters, per the SQL standard, so longer foreign key names are no longer permitted.

- Because the data dictionary provides information about database objects, the server no longer checks directory names in the data directory to find databases. Consequently, the `--ignore-db-dir` option and `ignore_db_dirs` system variable are extraneous and have been removed. Update system configurations and application programs accordingly.

- Previously, this was possible to use `CREATE TEMPORARY TABLE` to create a table in a nonexistent database by qualifying the table name with the name of a nonexistent database. This is no longer permitted.

- System table changes:

  - Many system tables have been converted from `MyISAM` (nontransactional) tables to `InnoDB` (transactional) tables. For example, as discussed elsewhere in these release notes, the grant tables are now `InnoDB` tables. Other examples follow.

  - The `func` table that stores loadable function information in the `mysql` system database now is an `InnoDB` (transactional) table. Previously, it was a `MyISAM` (nontransactional) table.

    In consequence of this change, `CREATE FUNCTION` and `DROP FUNCTION` statements cause an implicit commit, even when used for loadable functions (see Statements That Cause an Implicit Commit). Previously, they caused an implicit commit when used for stored functions, but not for loadable functions.

  - Previously, information about stored routines and events was stored in the `proc` and `event` tables of the `mysql` system database. Those tables are no longer used. Instead, information about stored routines and events is stored in the `routines`, `events`, and `parameters` data dictionary tables in the `mysql` system database. The old tables used the `MyISAM` (nontransactional) storage engine. The new tables use the `InnoDB` (transactional) engine.

    Previously, creating a stored routine that contained illegal characters produced a warning. This is now an error.

  - To permit access to system tables (for example, time zone or log tables) to be distinguished from access to nonsystem tables, the server uses the `Locking system tables` and `Opening system tables` thread states rather than the `System lock` and `Opening tables` thread states. See General Thread States.

- `InnoDB` changes:

  - Persistent `InnoDB` tablespaces now include transactional storage for *Serialized Dictionary Information (SDI)*, which is dictionary object data in serialized form. Along with the disappearance of `.frm` and trigger metadata files, mentioned previously, you might notice the appearance of `.SDI` files. These are serialized dictionary information files. SDI transactional storage is reserved for an in-progress feature not yet fully implemented.

  - A new command-line utility, `ibd2sdi`, is used to extract serialized dictionary information (SDI) from persistent `InnoDB` tablespaces. SDI data is not present in persistent `InnoDB` tablespaces in this release. The `ibd2sdi` utility is reserved for future use.

  - `InnoDB` startup code was refactored to support MySQL initialization changes related to the MySQL data dictionary feature.

- Upgrade and downgrade implications:
  - To upgrade to MySQL 8.0 from MySQL 5.7, you must perform the upgrade procedure described at Upgrading MySQL.
  - Downgrading from MySQL 8.0 to MySQL 5.7 is only supported using the logical downgrade method (a `mysqldump` downgrade). In-place downgrades are not supported.

(Bug #80481, Bug #22811659, WL #6378, WL #6383, WL #7896, WL #6384, WL #7897, WL #6388, WL #7898, WL #7284, WL #7630, WL #7836, WL #6390, WL #6382, WL #6389, WL #6387, WL #6385, WL #7053, WL #7158, WL #8980, WL #8150, WL #7488, WL #7066, WL #6929, WL #6391, WL #6392, WL #6599)

## Data Type Notes

- Bit functions and operators comprise `BIT_COUNT()`, `BIT_AND()`, `BIT_OR()`, `BIT_XOR()`, `&`, `|`, `^`, `~`, `<<`, and `>>`. Prior to MySQL 8.0, bit functions and operators required `BIGINT` (64-bit integer) arguments and returned `BIGINT` values, so they had a maximum range of 64 bits. Non-`BIGINT` arguments were converted to `BIGINT` prior to performing the operation and truncation could occur. Now bit functions and operators permit binary string type arguments (`BINARY`, `VARBINARY`, and the `BLOB` types) and return a value of like type, which enables them to take arguments and produce return values larger than 64 bits. Nonbinary string arguments are converted to `BIGINT` and processed as such, as before.

  Permitting binary string arguments for bit functions and operators makes it easier not only to manipulate larger values, but to perform bit operations not easily done previously on certain types of data, such as UUID and IPv6 values. For examples, see Bit Functions and Operators.

  An implication of this change in behavior is that bit operations on binary string arguments might produce a different result in MySQL 8.0 than in 5.7. For information about how to prepare in MySQL 5.7 for potential incompatibilities between MySQL 5.7 and 8.0, see Bit Functions and Operators, in MySQL 5.7 Reference Manual. (WL #8699)

## Deprecation and Removal Notes

- **Important Change; InnoDB:** The following `InnoDB` file format configuration options were deprecated in MySQL 5.7.7 and are now removed:
  - `innodb_file_format`
  - `innodb_file_format_check`
  - `innodb_file_format_max`
  - `innodb_large_prefix`

  File format configuration options were necessary for creating tables compatible with earlier versions of `InnoDB` in MySQL 5.1. Now that MySQL 5.1 has reached the end of its product lifecycle, these options are no longer required.

  The `FILE_FORMAT` column was removed from the `INNODB_SYS_TABLES` and `INNODB_SYS_TABLESPACES` Information Schema tables. (WL #7704)

- **InnoDB:** The `innodb_stats_sample_pages` system variable was removed. `innodb_stats_sample_pages` was deprecated in MySQL 5.6.3 and replaced by `innodb_stats_transient_sample_pages`. (WL #8903)

- **InnoDB:** The `innodb_locks_unsafe_for_binlog` system variable was removed. `innodb_locks_unsafe_for_binlog` was deprecated in MySQL 5.6.3. The `READ COMMITTED` isolation level provides similar functionality. (WL #8894)

- **InnoDB:** The `innodb_support_xa` system variable, which enables support for two-phase commit in XA transactions, was removed. As of MySQL 5.7.10, `InnoDB` support for two-phase commit in XA transactions is always enabled. (WL #8843)

- The deprecated `mysql_install_db` program has been removed from MySQL distributions. Data directory initialization should be performed by invoking `mysqld` with the `--initialize` or `--initialize-insecure` option instead. In addition, the deprecated `--bootstrap` option for `mysqld` that was used by `mysql_install_db` has been removed, and the `INSTALL_SCRIPTDIR` `CMake` option that controlled the installation location for `mysql_install_db` has been removed.

  Version 1 test suite code previously was located in the `mysql-test/lib/v1` directory of MySQL source distributions. This code used `mysql_install_db` and has been removed. The `MYSQL_INSTALL_DB` environment variable and a value of 1 for the `MTR_VERSION` environment variable are no longer supported. (WL #9071)

- The `mysql_plugin` utility has been removed. Alternatives include loading plugins at server startup using the `--plugin-load` or `--plugin-load-add` option, or at runtime using the `INSTALL PLUGIN` statement. (WL #8927)

- The deprecated `mysql_shutdown()` C API function and corresponding `COM_SHUTDOWN` client/server protocol command have been removed. Instead, use `mysql_query()` to execute a `SHUTDOWN` statement. (WL #9014)

## Doxygen Notes

- The MySQL source code has been updated to use Doxygen for the internal documentation. This is a work in progress. As new MySQL versions are distributed, the Doxygen documentation will be updated, with the latest version always available at https://dev.mysql.com/doc/index-other.html.

  It is also possible to generate the Doxygen content locally from a MySQL source distribution using the instructions at Generating MySQL Doxygen Documentation Content. (WL #8493)

## Optimizer Notes

- **InnoDB:** The storage engine interface now enables the optimizer to provide information about the size of the record buffer to be used for scans that the optimizer estimates will read multiple rows. The buffer size can vary based on the size of the estimate. `InnoDB` uses this variable-size buffering capability to take advantage of row prefetching, and to reduce the overhead of latching and B-tree navigation. Previously, `InnoDB` used a small, fixed-size buffer. (WL #7093)

- The optimizer now supports table-level `MERGE` and `NO_MERGE` hints for specifying whether derived tables or views should be merged into the outer query block or materialized using an internal temporary table. Examples:

  ```
  SELECT /*+ MERGE(dt) */ * FROM (SELECT * FROM t1) AS dt;
  SELECT /*+ NO_MERGE(dt) */ * FROM (SELECT * FROM t1) AS dt;
  ```

  For more information, see Optimizer Hints. (Bug #79554, Bug #22328100, WL #9307)

- MySQL now supports invisible indexes. An invisible index is not used by the optimizer at all, but is otherwise maintained normally. Indexes are visible by default. Invisible indexes make it possible to test the effect of removing an index on query performance, without making a destructive change that must be

undone should the index turn out to be required. This feature applies to `InnoDB` tables, for indexes other than primary keys.

To control whether an index is invisible explicitly for a new index, use a `VISIBLE` or `INVISIBLE` keyword as part of the index definition for `CREATE TABLE`, `CREATE INDEX`, or `ALTER TABLE`. To alter the invisibility of an existing index, use a `VISIBLE` or `INVISIBLE` keyword with the `ALTER TABLE ... ALTER INDEX` operation. For more information, see Invisible Indexes. (WL #8697)

- The `mysql` system database now contains a `column_stats` table designed to store statistics about column values. For more information, see Optimizer Statistics. (WL #8706)

## Packaging Notes

- Development milestone releases in previous MySQL series were numbered using a suffix of `-mN`, to indicate development milestone `N`. In MySQL 8.0, development releases use the suffix `-dmr`. For example, this release of MySQL is numbered `8.0.0-dmr`. (Bug #80408, Bug #22748154)
- As a consequence of the use of C++11 features described elsewhere in these release notes, the following packaging changes have been made:
  - Support for Red Hat Enterprise Linux 5 and Oracle Linux 5 RPMs has been dropped
  - Generic binary tarball builds have been moved to Red Hat Enterprise Linux 6

  (WL #8896)

## Parser Notes

- **Incompatible Change:** The parser rules for `SELECT` and `UNION` were refactored to be more consistent (the same `SELECT` syntax applies uniformly in each such context) and reduce duplication. Several user-visible effects resulted from this work, which may require rewriting of certain statements:
  - `NATURAL JOIN` permits an optional `INNER` keyword (`NATURAL INNER JOIN`), in compliance with standard SQL.
  - Right-deep joins without parentheses are permitted (for example, `... JOIN ... JOIN ... ON ... ON`), in compliance with standard SQL.
  - `STRAIGHT_JOIN` now permits a `USING` clause, similar to other inner joins.
  - The parser accepts parentheses around query expressions. For example, `(SELECT ... UNION SELECT ...)` is permitted.
  - The parser better conforms to the documented permitted placement of the `SQL_CACHE` and `SQL_NO_CACHE` query modifiers.
  - Left-hand nesting of unions, previously permitted only in subqueries, is now permitted in top-level statements. For example, this statement is now accepted as valid:

    ```
    (SELECT 1 UNION SELECT 1) UNION SELECT 1;
    ```

  - Locking clauses (`FOR UPDATE`, `LOCK IN SHARE MODE`) are allowed only in non-`UNION` queries. This means that parentheses must be used for `SELECT` statements containing locking clauses. This statement is no longer accepted as valid:

    ```
    SELECT 1 FOR UPDATE UNION SELECT 1 FOR UPDATE;
    ```

Instead, write the statement like this:

```
(SELECT 1 FOR UPDATE) UNION (SELECT 1 FOR UPDATE);
```

(Bug #11746363, Bug #25734, WL #8083, WL #8907)

- The parser rules for `CREATE TABLE` were refactored to be context independent and improve maintainability and extensibility. Several user-visible effects resulted from this work:

  - For generated columns, including `NOT NULL NULL` resulted in a column that included the `NOT NULL` attribute, which differed from nongenerated columns. Such definitions now use the final attribute `NULL`, resulting in a nullable column (consistent with nongenerated columns).

  - `CREATE TEMPORARY TABLE` no longer permits multiple instances of `TEMPORARY`.

  - Previously, `PARSE_GCOL_EXPR` was a keyword and could not be used as a label in stored programs. It is no longer a keyword and can be used as a label.

  - Messages for some syntax errors are more precise with respect to the location of the error within the statement.

  (WL #7840, WL #8067, WL #8433, WL #8434, WL #8345)

## Performance Schema Notes

- **Incompatible Change:** The Performance Schema now instruments server errors (and warnings), and exposes statistical information about them through a set of summary tables:

  - The `error` instrument controls whether error information is collected (enabled by default).

  - Several tables contain error information, summarized in various ways: `events_errors_summary_global_by_error`, `events_errors_summary_by_account_by_error`, `events_errors_summary_by_host_by_error`, `events_errors_summary_by_thread_by_error`, `events_errors_summary_by_user_by_error`.

  - The `performance_schema_error_size` system variable controls the number of instrumented errors.

  For more information, see Error Summary Tables

  If you upgrade to this MySQL release from an earlier version, you must run `mysql_upgrade` (and restart the server) to incorporate these changes into the `performance_schema` database.

  In consequence of the preceding changes, two server error symbols were renamed: `ER_CANT_SET_ENFORCE_GTID_CONSISTENCY_ON_WITH_ONGOING_GTID_VIOLATING_TRANSACTIONS` is now `ER_CANT_ENFORCE_GTID_CONSISTENCY_WITH_ONGOING_GTID_VIOLATING_TX` and `ER_SET_ENFORCE_GTID_CONSISTENCY_WARN_WITH_ONGOING_GTID_VIOLATING_TRANSACTIONS` is now `ER_ENFORCE_GTID_CONSISTENCY_WARN_WITH_ONGOING_GTID_VIOLATING_TX`.

  Also, several server error codes were found to be no longer used in the server and have been removed as obsolete. Applications that test specifically for any of these errors should be updated:

```
ER_BINLOG_READ_EVENT_CHECKSUM_FAILURE
ER_BINLOG_ROW_RBR_TO_SBR
ER_BINLOG_ROW_WRONG_TABLE_DEF
```

```
ER_CANT_ACTIVATE_LOG
ER_CANT_CHANGE_GTID_NEXT_IN_TRANSACTION
ER_CANT_CREATE_FEDERATED_TABLE
ER_CANT_CREATE_SROUTINE
ER_CANT_DELETE_FILE
ER_CANT_GET_WD
ER_CANT_SET_GTID_PURGED_WHEN_GTID_MODE_IS_OFF
ER_CANT_SET_WD
ER_CANT_WRITE_LOCK_LOG_TABLE
ER_CREATE_DB_WITH_READ_LOCK
ER_CYCLIC_REFERENCE
ER_DB_DROP_DELETE
ER_DELAYED_NOT_SUPPORTED
ER_DIFF_GROUPS_PROC
ER_DISK_FULL
ER_DROP_DB_WITH_READ_LOCK
ER_DROP_USER
ER_DUMP_NOT_IMPLEMENTED
ER_ERROR_DURING_CHECKPOINT
ER_ERROR_ON_CLOSE
ER_EVENTS_DB_ERROR
ER_EVENT_CANNOT_DELETE
ER_EVENT_CANT_ALTER
ER_EVENT_COMPILE_ERROR
ER_EVENT_DATA_TOO_LONG
ER_EVENT_DROP_FAILED
ER_EVENT_MODIFY_QUEUE_ERROR
ER_EVENT_NEITHER_M_EXPR_NOR_M_AT
ER_EVENT_OPEN_TABLE_FAILED
ER_EVENT_STORE_FAILED
ER_EXEC_STMT_WITH_OPEN_CURSOR
ER_FAILED_ROUTINE_BREAK_BINLOG
ER_FLUSH_MASTER_BINLOG_CLOSED
ER_FORM_NOT_FOUND
ER_FOUND_GTID_EVENT_WHEN_GTID_MODE_IS_OFF__UNUSED
ER_FRM_UNKNOWN_TYPE
ER_GOT_SIGNAL
ER_GRANT_PLUGIN_USER_EXISTS
ER_GTID_MODE_REQUIRES_BINLOG
ER_GTID_NEXT_IS_NOT_IN_GTID_NEXT_LIST
ER_HASHCHK
ER_INDEX_REBUILD
ER_INNODB_NO_FT_USES_PARSER
ER_LIST_OF_FIELDS_ONLY_IN_HASH_ERROR
ER_LOAD_DATA_INVALID_COLUMN_UNUSED
ER_LOGGING_PROHIBIT_CHANGING_OF
ER_MALFORMED_DEFINER
ER_MASTER_KEY_ROTATION_ERROR_BY_SE
ER_NDB_CANT_SWITCH_BINLOG_FORMAT
ER_NEVER_USED
ER_NISAMCHK
ER_NO_CONST_EXPR_IN_RANGE_OR_LIST_ERROR
ER_NO_FILE_MAPPING
ER_NO_GROUP_FOR_PROC
ER_NO_RAID_COMPILED
ER_NO_SUCH_KEY_VALUE
ER_NO_SUCH_PARTITION__UNUSED
ER_OBSOLETE_CANNOT_LOAD_FROM_TABLE
ER_OBSOLETE_COL_COUNT_DOESNT_MATCH_CORRUPTED
ER_ORDER_WITH_PROC
ER_PARTITION_SUBPARTITION_ERROR
ER_PARTITION_SUBPART_MIX_ERROR
ER_PART_STATE_ERROR
ER_PASSWD_LENGTH
ER_QUERY_ON_MASTER
ER_RBR_NOT_AVAILABLE
ER_SKIPPING_LOGGED_TRANSACTION
```

```
ER_SLAVE_CHANNEL_DELETE
ER_SLAVE_MULTIPLE_CHANNELS_HOST_PORT
ER_SLAVE_MUST_STOP
ER_SLAVE_WAS_NOT_RUNNING
ER_SLAVE_WAS_RUNNING
ER_SP_GOTO_IN_HNDLR
ER_SP_PROC_TABLE_CORRUPT
ER_SQL_MODE_NO_EFFECT
ER_SR_INVALID_CREATION_CTX
ER_TABLE_NEEDS_UPG_PART
ER_TOO_MUCH_AUTO_TIMESTAMP_COLS
ER_UNEXPECTED_EOF
ER_UNION_TABLES_IN_DIFFERENT_DIR
ER_UNSUPPORTED_BY_REPLICATION_THREAD
ER_UNUSED1
ER_UNUSED2
ER_UNUSED3
ER_UNUSED4
ER_UNUSED5
ER_UNUSED6
ER_VIEW_SELECT_DERIVED_UNUSED
ER_WRONG_MAGIC
ER_WSAS_FAILED
```

(WL #8058)

- Previously, the `DIGEST` and `DIGEST_TEXT` columns in the Performance Schema `events_statements_current` table were populated only after statement execution ended. Now, the columns are populated just after parsing and before statement execution begins. This enables monitoring applications to access statement digest information during statement execution. (Bug #23336542)

- Previously, Performance Schema optimizations focused on reducing the overhead involved in collecting monitoring data. Complementing that earlier work, overhead now is also reduced for Performance Schema queries that retrieve that data. This is achieved by the addition of indexes to most Performance Schema tables, which gives the optimizer access to execution plans other than full table scans. These indexes also improve performance for related objects, such as `sys` schema views that use those tables. For more information, see Optimizing Performance Schema Queries. (WL #6616)

- The size of the `ROLE` column of the `setup_actors` Performance Schema table was increased from 16 to 32 characters. (WL #9262)

## Security Notes

- The `validate_password_check_user_name` system variable is now enabled by default rather than disabled. This means that when the `validate_password` plugin is enabled, by default it now rejects passwords that match the current session user name. (WL #9480)

- The client-side `--ssl` and `--ssl-verify-server-cert` options have been removed. Use `--ssl-mode=REQUIRED` instead of `--ssl=1` or `--enable-ssl`. Use `--ssl-mode=DISABLED` instead of `--ssl=0`, `--skip-ssl`, or `--disable-ssl`. Use `--ssl-mode=VERIFY_IDENTITY` instead of `--ssl-verify-server-cert` options. (The server-side `--ssl` option remains unchanged.)

  For the C API, `MYSQL_OPT_SSL_ENFORCE` and `MYSQL_OPT_SSL_VERIFY_SERVER_CERT` options for `mysql_options()` correspond to the client-side `--ssl` and `--ssl-verify-server-cert` options and have been removed. Use `MYSQL_OPT_SSL_MODE` with an option value of `SSL_MODE_REQUIRED` or `SSL_MODE_VERIFY_IDENTITY` instead. (WL #9091)

## Spatial Data Support

- Spatial functions for import and export of Well-Known Text (WKT) values used MySQL `'GEOMETRYCOLLECTION()'` nonstandard syntax rather than OpenGIS `'GEOMETRYCOLLECTION EMPTY'` standard syntax. Now both syntaxes are understood for import and the standard syntax is used for export. See Functions That Create Geometry Values from WKT Values. (Bug #23632147, Bug #81964)

- The `ST_X()` and `ST_Y()` spatial functions now permit an optional second argument that specifies an X or Y coordinate value, respectively. With two arguments, the function result is the point value from the first argument with the appropriate coordinate modified. In addition, `ST_X()` and `ST_Y()` with a single argument now are stricter and produce an `ER_UNEXPECTED_GEOMETRY_TYPE` error rather than returning `NULL` if the argument is a valid geometry but not a point. For more information, see Point Property Functions. (WL #8606)

- The `ST_SRID()` spatial function now permits an optional second argument that specifies a SRID value. With two arguments, the function result is the geometry value from the first argument with its SRID modified according to the second argument. For more information, see General Geometry Property Functions. (WL #8543)

- MySQL now stores information about spatial reference systems other than SRID 0, for use with spatial data. This information is stored in the `st_spatial_reference_systems` data dictionary table and is based on EPSG Dataset 8.7. For information about spatial reference systems, see Spatial Reference System Support.

  Previously, the `ST_IsValid()`, `ST_MakeEnvelope()`, and `ST_Validate()` functions required geometry arguments with SRID 0. They now accept geometry arguments with an SRID for a projected spatial reference system. (WL #8579)

- In MySQL 5.7, several spatial functions available under multiple names were deprecated to move in the direction of making the spatial function namespace more consistent, the goal being that each spatial function name begin with `ST_` if it performs an exact operation, or with `MBR` if it performs an operation based on minimum bounding rectangles. The deprecated functions have now been removed to leave only the corresponding `ST_` and `MBR` functions:

  - These functions are removed in favor of the `MBR` names: `Contains()`, `Disjoint()`, `Equals()`, `Intersects()`, `Overlaps()`, `Within()`.

  - These functions are removed in favor of the `ST_` names: `Area()`, `AsBinary()`, `AsText()`, `AsWKB()`, `AsWKT()`, `Buffer()`, `Centroid()`, `ConvexHull()`, `Crosses()`, `Dimension()`, `Distance()`, `EndPoint()`, `Envelope()`, `ExteriorRing()`, `GeomCollFromText()`, `GeomCollFromWKB()`, `GeomFromText()`, `GeomFromWKB()`, `GeometryCollectionFromText()`, `GeometryCollectionFromWKB()`, `GeometryFromText()`, `GeometryFromWKB()`, `GeometryN()`, `GeometryType()`, `InteriorRingN()`, `IsClosed()`, `IsEmpty()`, `IsSimple()`, `LineFromText()`, `LineFromWKB()`, `LineStringFromText()`, `LineStringFromWKB()`, `MLineFromText()`, `MLineFromWKB()`, `MPointFromText()`, `MPointFromWKB()`, `MPolyFromText()`, `MPolyFromWKB()`, `MultiLineStringFromText()`, `MultiLineStringFromWKB()`, `MultiPointFromText()`, `MultiPointFromWKB()`, `MultiPolygonFromText()`, `MultiPolygonFromWKB()`, `NumGeometries()`, `NumInteriorRings()`, `NumPoints()`, `PointFromText()`, `PointFromWKB()`, `PointN()`, `PolyFromText()`, `PolyFromWKB()`, `PolygonFromText()`, `PolygonFromWKB()`, `SRID()`, `StartPoint()`, `Touches()`, `X()`, `Y()`.

  - `GLength()` is removed in favor of `ST_Length()`.

  (WL #8157)

## Test Suite Notes

- `mysql-test-run.pl` now supports a `--do-suite` option, which is similar to `--do-test` but permits specifying entire suites of tests to run. (Bug #24350345)
- The `mysqltest rmdir` command fails if the directory to be removed contains any files or directories. To enable recursive removal of a directory as well as its contents, if any, `mysqltest` now supports a `force-rmdir` command. (Bug #24316799)
- Two new test suite options make it easier to debug test cases:
  - `mysql-test-run.pl` supports a `--mysqltest=`*`options`* option that enables options to be passed to `mysqltest`.
  - `mysqltest` supports a `--trace-exec` option that causes it to immediately print output from executed programs to `stdout`.

  `mysql-test-run.pl` now recognizes the `MTR_CTEST_TIMEOUT` environment variable. If set, the value is a timeout in seconds to pass to `ctest` unit test commands. (Bug #21821049, Bug #21278845)
- For test cases in the MySQL test suite, it was previously possible to use symbolic error names for the `--error` command only for server errors. This is now also possible for client errors. For example:

  ```
  --error CR_SERVER_GONE_ERROR
  ```

  (Bug #21048973, Bug #76972)
- The `mysqltest` program now has a `copy_files_wildcard` command that copies all files that match a pattern from a source directory to a destination directory. See the MySQL Server Doxygen documentation, available at https://dev.mysql.com/doc/index-other.html. (Bug #82111, Bug #23743035)

## X Plugin Notes

- The `Protobuf` decoder class limited the number of nested objects to 50 (the default value). (Bug #23707238, Bug #82025)
- The statement `list_objects` incorrectly reported a table as a collection. (Bug #23631240)
- The `create_collection` statement created a collection table with a unique key index on the `'_id'` column instead of on the primary key. (Bug #23284569)

## Functionality Added or Changed

- **Incompatible Change; Partitioning:** The generic partitioning handler has been removed from the MySQL server. As part of this change, `mysqld` no longer supports the `--partition` and `--skip-partition` options, and the server can no longer be built using `-DWITH_PARTITION_STORAGE_ENGINE`. `partition` is also no longer displayed in the output of `SHOW PLUGINS`, or shown in the `INFORMATION_SCHEMA.PLUGINS` table.

  In order to support partitioning of a given table, the storage engine used for the table must now provide its own (“native”) partitioning handler. `InnoDB` is the only storage engine supported in MySQL 8.0 which includes a native partitioning handler. An attempt to create partitioned tables in MySQL 8.0 using any other storage engine fails. (The `NDB` storage engine used by MySQL NDB Cluster also provides its own partitioning handler, but is currently not supported by MySQL 8.0.)

**Effects on upgrades.** The direct upgrade of a partitioned table using a storage engine other than `InnoDB` (such as `MyISAM`) from MySQL 5.7 (or earlier) to MySQL 8.0 is not supported. There are two options for upgrading such a table to be compatible with MySQL 8.0, listed here:

- Remove the table's partitioning; you can do this without any data loss by executing an `ALTER TABLE ... REMOVE PARTITIONING` statement.
- Change the storage engine used for the table to `InnoDB`, using `ALTER TABLE ... ENGINE=INNODB`; this leaves the table's partitioning in place. At least one of these operations must be performed for any partitioned non-`InnoDB` table, prior to upgrading the server to MySQL 8.0. Otherwise, such a table cannot be used following the upgrade.

  For information about converting `MyISAM` tables to `InnoDB`, see Converting Tables from MyISAM to InnoDB.

An analogous situation is met when importing databases from a dump file that was created in MySQL 5.7 or earlier using `mysqldump` into a MySQL 8.0 server, due to the fact that table creation statements that would result in a partitioned table using a storage engine without such support fail with an error in MySQL 8.0. For this reason you must ensure that any statements in the dump file creating partitioned tables do not also specify an unsupported storage engine. You can do this either by removing any references to partitioning from `CREATE TABLE` statements that use a value for the `STORAGE ENGINE` option other than `InnoDB`, or by specifying the storage engine as `InnoDB` (or allowing `InnoDB` to be used by default).

For more information, see Partitioning Limitations Relating to Storage Engines. (WL #8971, WL #9457)

- **InnoDB:** The `innodb_buffer_pool_debug` option permits multiple buffer pool instances when the buffer pool is less than 1GB in size, ignoring the 1GB minimum buffer pool size constraint imposed on `innodb_buffer_pool_instances`. (Bug #24287290)
- **InnoDB:** A new dynamic configuration option, `innodb_deadlock_detect`, may be used to disable deadlock detection. On high concurrency systems, deadlock detection can cause a slowdown when numerous threads wait for the same lock. At times, it may be more efficient to disable deadlock detection and rely on the `innodb_lock_wait_timeout` setting for transaction rollback when a deadlock occurs. (Bug #23477773, WL #9383)
- **InnoDB:** The `libinnodb_zipdecompress.a` library allows external tools to use the `page_zip_decompress_low()` function to decompress `InnoDB` pages. (Bug #21405300, Bug #77664)
- **InnoDB:** To address contention that could occur under some workloads, the buffer pool mutex was removed and replaced by several list and hash protecting mutexes. Also, several buffer pool related variables no longer require buffer pool mutex protection. Thanks to Yasufumi Kinoshita and Laurynas Biveinis for the patch. (Bug #20381905, Bug #75534, WL #8423)
- **InnoDB:** `InnoDB` now avoids intermediate commits that would occur every 10000 rows during `ALTER TABLE ALGORITHM=COPY` operations. The purpose of intermediate commits was to speed up recovery in the case of an aborted `ALTER TABLE ALGORITHM=COPY` operation. If an `ALTER TABLE ALGORITHM=COPY` operation is aborted, the new, uncommitted table is now dropped during DDL log recovery before the undo log is rolled back, thereby avoiding time-consuming data rollback for the

uncommitted table. Undo logging is now suppressed for `ALTER TABLE ALGORITHM=COPY` operations unless there is an `IGNORE` clause or something else that requires rollback capability.

If there is full-text index on the table being altered, full-text data is inserted into full-text auxiliary tables as the `ALTER TABLE ALGORITHM=COPY` operation inserts rows into the new, uncommitted table. Previously, full-text data was only processed on transaction commit. (Bug #17479594)

- **InnoDB:** To reduce read-write lock contention that can result from multiple purge threads purging rows from the same table, undo records are now grouped and assigned to different purge threads by table ID. (WL #9387)
- **InnoDB:** `InnoDB` code now uses the C++ `std::thread` library for thread management. (WL #9359)
- **InnoDB:** `BLOB` code was refactored to provide an internal C++ interface for operations on compressed and uncompressed `BLOB` data. (WL #8985, WL #9141)
- **InnoDB:** The `InnoDB memcached` plugin now supports multiple `get` operations (fetching multiple key/value pairs in a single `memcached` query) and range queries. See InnoDB memcached Multiple get and Range Query Support. (WL #6650)
- **InnoDB:** When encountering index tree corruption, `InnoDB` writes a corruption flag to the redo log, which makes the corruption flag crash safe. `InnoDB` also writes in-memory corruption flag data to an engine-private system table on each checkpoint. During recovery, `InnoDB` reads corruption flags from both locations and merges results before marking in-memory table and index objects as corrupt. (WL #7816)
- **InnoDB:** `InnoDB` no longer creates `.isl` files (`InnoDB` Symbolic Link files) when creating tablespace data files outside of the MySQL data directory.

  With this change, moving a remote tablespace while the server is offline by manually modifying an `.isl` file is not supported. (WL #6416)
- **InnoDB:** `InnoDB` no longer supports compressed temporary tables. When `innodb_strict_mode` is enabled (the default), `CREATE TEMPORARY TABLE` returns an error if `ROW_FORMAT=COMPRESSED` or `KEY_BLOCK_SIZE` is specified. If `innodb_strict_mode` is disabled, warnings are issued and the temporary table is created using a non-compressed row format.

  With this change, all temporary tables are created in the shared temporary tablespace, `ibtmp1`.

  The `PER_TABLE_TABLESPACE` and `IS_COMPRESSED` columns were removed from the `INFORMATION_SCHEMA.INNODB_TEMP_TABLE_INFO`. (WL #7899)
- **InnoDB:** The new `INFORMATION_SCHEMA.INNODB_CACHED_INDEXES` table reports the number of index pages cached in the `InnoDB` buffer pool for each index. (WL #7170)
- **InnoDB:** The `innodb_checksums` system variable was removed. `innodb_checksums` was replaced by `innodb_checksum_algorithm` in MySQL 5.6.3. (WL #8893)
- **InnoDB:** `InnoDB` startup code was refactored. (WL #7488)
- **InnoDB:** The `innodb_flush_method` default value is no longer `NULL`. On Unix-like systems, the default value is `fsync`. On Windows systems, the default value is `unbuffered`.

  On Windows, the `innodb_flush_method` setting no longer affects the `innodb_use_native_aio` setting. There are now two possible settings for `innodb_flush_method` on Windows, `unbuffered` (unbuffered I/O) and `normal` (buffered I/O). With this change, you can enable asynchronous I/O with buffered I/O, which is a new combination (`innodb_use_native_aio=ON` and `innodb_flush_method=normal`). The `async_unbuffered` setting was removed.

You can now set `innodb_flush_method` and `innodb_change_buffering` configuration options using numeric values. (WL #7488)

- **InnoDB:** `InnoDB` no longer creates an `.isl` file (InnoDB Symbolic Link files) when creating a file-per-table tablespace data file outside of the MySQL data directory. `InnoDB` now uses the redo log to locate remote tablespace data files.

  Offline relocation of a file-per-table tablespace data file created outside of the MySQL data directory by modifying the `.isl` file is no longer supported. (WL #6416)

- **InnoDB:** The current maximum auto-increment counter value is now written to the redo log each time the value changes, and saved to an engine-private system table on each checkpoint. These changes make the current maximum auto-increment counter value persistent across server restarts. Additionally:

  - A server restart no longer cancels the effect of the `AUTO_INCREMENT = N` table option. If you initialize the auto-increment counter to a specific value, or if you alter the auto-increment counter value to a larger value, the new value is persisted across server restarts.
  - A server restart immediately following a `ROLLBACK` operation no longer results in the reuse of auto-increment values that were allocated to the rolled-back transaction.
  - If you modify an `AUTO_INCREMENT` column value to a value larger than the current maximum auto-increment value (in an `UPDATE` operation, for example), the new value is persisted, and subsequent `INSERT` operations allocate auto-increment values starting from the new, larger value.

  For more information, see AUTO_INCREMENT Handling in InnoDB, and InnoDB AUTO_INCREMENT Counter Initialization. (Bug #199, Bug #13726455, WL #6204)

- **Replication:** There are two improvements to how a CHANGE MASTER TO statement is written into the error log (`mysqld.log`):

  - Before, no commas were put between the option specifications (for example `MASTER_USER =` and `MASTER_PASSWORD =` ), so users who wanted to use the statement by copy and paste had to insert the commas manually. Commas are now inserted when the statement is written to the error log.
  - When the literal “<secret>” is inserted as a placeholder for the `MASTER_PASSWORD` value, no quotes are used now, so users who forget to replace the literal with the real password before a copy and paste gets a syntax error immediately, instead of running into other issues.

  (Bug #18194384)

- **Replication:** It is now possible to restore a backup of a GTID-based replication server because you can add GTIDs to `gtid_purged`, regardless of whether `gtid_executed` is empty or not. This enables you to restore backups from GTID-based replication servers without losing existing GTID information and binary logs. The GTIDs to add are those which existed in `gtid_executed` at the time of taking the backup. The syntax for `SET GTID_PURGED` has been extended so that `SET GTID_PURGED ="+gtid_set"` adds *`gtid_set`* to the existing `gtid_purged` GTID set. (WL #6591)

- **Replication:** New Performance Schema stages have been added to show the progress of row-based replication. You can use these stages to check the progress of slow operations in row-based replication. Additionally you can find out which database the changes are being applied to. This assists in troubleshooting row-based replication issues and provides more information for performance tuning. For more information see Monitoring Row-based Replication (WL #7364)

- **JSON:** This release adds an unquoting extraction operator `->>`, sometimes also referred to as an inline path operator, for use with `JSON` documents stored in MySQL. The new operator is similar to the `->`

operator, but performs JSON unquoting of the value as well. For a JSON column `mycol` and JSON path expression `mypath`, the following three expressions are equivalent:

- `JSON_UNQUOTE( JSON_EXTRACT(mycol, "$.mypath") )`
- `JSON_UNQUOTE(mycol->"$.mypath")`
- `mycol->>"$.mypath"`

The `->>` operator can be used in SQL statements wherever `JSON_UNQUOTE(JSON_EXTRACT())` would be allowed. This includes (but is not limited to) `SELECT` lists, `WHERE` and `HAVING` clauses, and `ORDER BY` and `GROUP BY` clauses.

For more information, see Functions That Search JSON Values, and JSON Path Syntax. (Bug #78736, Bug #21980346, WL #9124)

- To produce more accurate estimates, the `MEMORY` storage engine now calculates index statistics (records per key estimates) using floating-point rather than integer arithmetic. (Bug #23024059)
- A new `CMake` option, `INSTALL_STATIC_LIBRARIES`, enables control over whether to install static libraries. The default is `ON`. If set to `OFF`, these libraries are not installed: `libmysqlclient.a`, `libmysqld.a`, `libmysqlservices.a`. (Bug #22891432)
- The internal `mysql_prepare_create_table()` server function has been refactored for improved code maintainability and clarity. This code revision results in the following minor changes of behavior for `CREATE TABLE` and `ALTER TABLE`:
  - Attempts to create a second primary key based on a `NULL` column now produce an `ER_MULTIPLE_PRI_KEY` error rather than `ER_PRIMARY_CANT_HAVE_NULL`.
  - Attempts to create a second primary key based on a generated column now produce an `ER_MULTIPLE_PRI_KEY` error rather than `ER_UNSUPPORTED_ACTION_ON_GENERATED_COLUMN`.
  - Attempts to create a full-text key on a `JSON` column now produce an `ER_JSON_USED_AS_KEY` error rather than `ER_BAD_FT_COLUMN`.
  - Attempts to create a key in a storage engine that does not support keys (for example, `EXAMPLE`) now produce an `ER_TOO_MANY_KEYS` error rather than `ER_TOO_MANY_KEY_PARTS`.

  (Bug #22884886)
- Previously, the Performance Schema was not built for `libmysqld`, the embedded server. This prevented use of the `SHOW STATUS` and `SHOW VARIABLES` statements with `show_compatibility_56=OFF` because, with that setting, those statements take their results from Performance Schema tables. Now for `libmysqld`, the required Performance Schema tables are built (with no instrumentation collected), so that those `SHOW` statements can be supported with `show_compatibility_56=OFF`. (Bug #22809694)
- Several internal functions used by `JSON_CONTAINS()`, `JSON_SEARCH()`, and other MySQL JSON functions created excessive numbers of local copies of keys, values, or both, when performing inspections of JSON objects. Such copying has been eliminated or reduced in many cases. In addition, the lifetimes of temporary objects used by some of these functions have been reduced. These changes should make these and related JSON functions perform more efficiently than previously, and with fewer resources required. (Bug #22602142)
- If the system `lz4` and `openssl zlib` commands are available, the `lz4_decompress` and `zlib_decompress` utilities are unneeded. Two changes enable those utilities not to be built: If the new

`WITH_LZ4` `CMake` option is set to `system`, `lz4_decompress` is not built or installed. If the `WITH_ZLIB` `CMake` option is set to `system`, `zlib_decompress` is not built or installed. (Bug #22329851)

- Source files for the MySQL strings library have been converted from C (`.c` suffix) to C++ (`.cc` suffix). This enables stricter compilation checks and use of C++ features in the library code. (Bug #22124719)
- Source code for the `mysys` library now uses C++ rather than C to take advantage of stricter compilation checks and permit use of C++ features. (Bug #21881278)
- For improved performance and better compatibility with other platforms, `my_sync()` on OS X now uses `fsync()` rather than `fcntl()` with the `F_FULLSYNC` flag. (Bug #20742269)
- A new `CMake` option, `WITH_TSAN`, permits enabling ThreadSanitizer for compilers that support it. (Bug #80409, Bug #23171902)
- The global list of connections, previously protected by a single mutex, has been partitioned into eight parts, each protected by its own instance of the mutex. The result is a reduction of overhead and improved performance for connection processing. An implication of this change for monitoring purposes is that the Performance Schema now exposes eight different instances each of the `LOCK_thd_list` mutex, `LOCK_thd_remove` mutex, and `COND_thd_list` condition variable. (WL #9250)
- MySQL now provides functions to manipulate UUID values and make them easier to work with:
  - `UUID_TO_BIN()` and `BIN_TO_UUID()` convert between UUID values in string and binary formats (represented as hexadecimal characters and `VARBINARY(16)`, respectively). This permits conversion of string UUID values to binary values that take less storage space. UUID values converted to binary can be represented in a way that permits improved indexing efficiency.
  - `IS_UUID()` returns 1 or 0 to indicate whether its argument is a valid string-format UUID value.

  For more information about these functions, see Miscellaneous Functions (WL #8920)
- The server now relies on storage engines to clean up temporary tables left from previous server runs. `InnoDB` does this by discarding the temporary tablespace on restart. `MyISAM` and other similar storage engines still rely on scanning the temporary directory to detect leftover tables, by looking for files belonging to these engines with a certain name pattern. (WL #7784)
- The server no longer performs conversion of pre-MySQL 5.1 database names containing special characters to 5.1 format with the addition of a `#mysql50#` prefix. Because these conversions are no longer performed, the `--fix-db-names` and `--fix-table-names` options for `mysqlcheck`, the `UPGRADE DATA DIRECTORY NAME` clause for the `ALTER DATABASE` statement, and the `Com_alter_db_upgrade` status variable have been removed.

  Upgrades are supported only from one major version to another (for example, 5.0 to 5.1, or 5.1 to 5.5), so there should be little remaining need for conversion of older 5.0 database names to current versions of MySQL. As a workaround, upgrade a MySQL 5.0 installation to MySQL 5.1 before upgrading to a more recent release. (WL #8186)

## Bugs Fixed

- **Incompatible Change:** Concatenation of spatial values makes little sense, so the `CONCAT()` and `CONCAT_WS()` functions now produce an error for spatial arguments. (Bug #22893669)
- **Important Change; JSON:** The empty string value is now accepted as a key when used with `JSON` functions such as `JSON_EXTRACT()`. In such cases, it must be quoted. (Bug #79643, Bug #22366102)
- **NDB Cluster:** Previously, the `mysql.ndb_binlog_index` table was created even if the server was built without `NDB`. Now the table is created only if the server is built with `NDB`. (Bug #22874872)

- **InnoDB; Microsoft Windows:** An unspecified block size resulted in an empty `INFORMATION_SCHEMA.TABLESPACES` table on Windows NTFS with a cluster page size greater than or equal to 8K. (Bug #23598872)
- **InnoDB; Microsoft Windows:** Compilation of `InnoDB` with Visual Studio 2015 Update 2 returned warnings. (Bug #23056963)
- **InnoDB:** An `ALTER TABLE ... ALGORITHM=COPY` operation that added a foreign key constraint failed after due to an intermediate commit that occurred after 10000 rows were copied. The intermediate commit reset the foreign key checks flag, causing the operation to fail. The intermediate commit no longer occurs. (Bug #28662255, Bug #92471)

  References: See also: Bug #17479594.
- **InnoDB:** Unnecessary checks were removed from the `ut_cpuid()` function which is used to fetch information about the CPU. (Bug #24405292)
- **InnoDB:** An asynchronous read operation on a deleted tablespace raised an error. (Bug #24388498)
- **InnoDB:** `dict_col_t` accessors were added to the `InnoDB` code. (Bug #24363566)
- **InnoDB:** `dict_col` functions in the `InnoDB` code were replaced by accessors. (Bug #24361098)
- **InnoDB:** `dict_index_t` functions in the `InnoDB` code were replaced by accessors. (Bug #24361023)
- **InnoDB:** Unnecessary code that checked for and released reserved adaptive hash index search latches was removed. (Bug #24300175)
- **InnoDB:** A system tablespace data file size greater than 4G on a 32-bit operating system could result in an overflow condition. (Bug #23753625)
- **InnoDB:** Internal methods for accessing table object data were added to `dict_table_t`. (Bug #23748128)
- **InnoDB:** The restriction that required the first undo tablespace to use `space_id` 1 was removed to avoid `space_id` conflicts with existing tablespaces during upgrade. The first undo tablespace can now use a `space_id` other than 1. `space_id` values for undo tablespaces are still assigned in a consecutive sequence. (Bug #23517560)
- **InnoDB:** Internal accessor functions for iterating the indexes of a table were replaced with accessor methods. Dead code was removed. (Bug #23336108)
- **InnoDB:** The `mysql.innodb_index_stats` and `mysql.innodb_table_stats` table definitions, which were previously created by an SQL script, are now hard-coded. As a result, the `dict_table_schema_check` function is longer required and was removed. (Bug #23336079)
- **InnoDB:** The `ut_snprint` function was replaced by the C++11 `snprintf` function. (Bug #23329353)
- **InnoDB:** For consistency, instances of `ulint` in `InnoDB` code were replaced with `space_id_t` and `page_no_t` data types. (Bug #23297169)
- **InnoDB:** Use of boost::atomic in `InnoDB` code was replaced with std::atomic. (Bug #23280649)
- **InnoDB:** MySQL binaries were not built with the NUMA feature. (Bug #23259754)
- **InnoDB:** References to `UNIV_NONINL` and `UNIV_MUST_NOT_INLINE` were removed. The `fut0fut.cc` and `ut0byte.cc` files, which were only necessary when `UNIV_NONINL` was defined, were also removed. (Bug #23150562)

- **InnoDB:** The `mutex_own()` mapping caused warnings when compiling with Clang or newer GCC compilers. (Bug #23090278)
- **InnoDB:** Querying the Performance Schema for `InnoDB` memory allocation event data incorrectly reported values of 0. (Bug #23020280)
- **InnoDB:** `DBUG_OFF` compile-time flags were replaced by `UNIV_DEBUG` flags. To improve error log output, `ut_dbg_assertion_failed()` now uses `sql_print_error()` to display the file name, line number, and message in a single line. The thread ID is displayed in a subsequent line. (Bug #22996442, Bug #23028144)
- **InnoDB:** `SHOW ENGINE INNODB STATUS` output displayed negative spin rounds per wait values. Thanks to Laurynas Biveinis for the patch. (Bug #22844987, Bug #79703)
- **InnoDB:** The `innodb_disable_resize_buffer_pool_debug` option was removed. The patch for this change also removed a code variable and simplified the `buf_pool_resize()` function. (Bug #22755053)
- **InnoDB:** A global counter (`ut_rnd_ulint_counter`) was changed to a thread-local counter to make it scalable on multi-core systems. (Bug #22733635, Bug #80354)
- **InnoDB:** After a successful `ALTER TABLE ... ALGORITHM=COPY` operation, an assertion was raised while building a previous version of a clustered index record. (Bug #22707367)
- **InnoDB:** A DML operation that updated a counter in a table with a virtual index raised on assertion in `row_parse_int()`. (Bug #22650195)
- **InnoDB:** The `InnoDB memcached` plugin would not load when compiled with libevent 2.0. (Bug #22646919)
- **InnoDB:** Unused calculations for integer-based `rec_per_key` values were removed from `InnoDB`. Integer-based `rec_per_key` information was replaced by floating point index statistics in an earlier release. (Bug #22625348)
- **InnoDB:** On slow shutdown, purge thread shutdown was initiated before the background rollback thread exited, resulting in an assertion failure. (Bug #22561332)
- **InnoDB:** Blocks were lost in `row_vers_old_has_index_entry()` due to unfreed heaps. (Bug #22543834, Bug #79973)
- **InnoDB:** A transportable tablespace debug test raised an assertion that was due to a race condition. (Bug #22453668)
- **InnoDB:** The server failed to start due to missing undo tablespaces. (Bug #22452992)
- **InnoDB:** `InnoDB` recovery asserted while attempting to close an undo tablespace due to buffered undo tablespace changes introduced by the recovery process. (Bug #22361764)
- **InnoDB:** Reallocation of memcached-referenced memory raised an assertion. (Bug #22304250, Bug #79500)
- **InnoDB:** Building `InnoDB` with C++11 returned “register” deprecation warnings. Handling of “register” deprecation warnings remained in the code after the deprecated “register” keyword was removed. Also, an unused declaration of `yyset_extra()` was removed. (Bug #22292704)
- **InnoDB:** `SHOW CREATE TABLE` output for partitioned tables did not accurately display tablespace assignment information for table partitions. (Bug #22245554)

- **InnoDB:** An `ALTER TABLE...TRUNCATE PARTITION` operation ignored the table's `KEY_BLOCK_SIZE` attribute and used the default value instead, which is half of the `innodb_page_size` value. (Bug #22186558, Bug #79223)
- **InnoDB:** Memory leaks in `innochecksum` were corrected. (Bug #22179518)
- **InnoDB:** A `SPACE_ID` column was added to the `INNODB_CACHED_INDEXES` table. The `INDEX_ID` value is no longer a global unique identifier. (Bug #22172026)
- **InnoDB:** A purge thread open table callback for virtual columns raised an assertion due to an unexpected data dictionary table latch. As a temporary workaround, purge is temporarily disabled for virtual generated columns. This temporary workaround may cause b-tree expansion due to unpurged delete-marked records for indexes on virtual columns. (Bug #22153217)
- **InnoDB:** Creating a table with a full-text index and a foreign key constraint failed when `foreign_key_checks` was disabled. (Bug #22094601, Bug #78955)

  References: This issue is a regression of: Bug #16845421.
- **InnoDB:** The `ha_innobase::m_primary_key` field was removed. It was redundant. A boolean predicate, `TABLE_SHARE::is_missing_primary_key()`, was added. (Bug #21928734, Bug #78662)
- **InnoDB:** A buffer pool load operation that attempted to load an uninitialized page caused a Valgrind failure. (Bug #21747906)
- **InnoDB:** Unused functions introduced by the `InnoDB memcached` plugin were removed. (Bug #21625760)
- **InnoDB:** An internal global variable used by the `innodb_buffer_pool_size` configuration option was removed. (Bug #21512749)
- **InnoDB:** An `InnoDB` page cleaner thread asserted during a buffer pool resize operation. (Bug #21473497)
- **InnoDB:** Log buffer contention was reduced with the addition of a second buffer, allowing for concurrent log buffer writing and flushing. A new mutex was added to protect log buffer flushing. Thanks to Zhai Weixiang for the patch. (Bug #21352937, Bug #77094)
- **InnoDB:** Unused `InnoDB` and `libsql` functions and variables were removed, and global symbols were converted to static keywords, where possible. (Bug #21153166, Bug #21141390, Bug #77146, Bug #21178589)
- **InnoDB:** A number of unused predefined functions were removed from an internal SQL parser that is used for implementing full-text indexes, updating the data dictionary, and updating persistent statistics. (Bug #21126390, Bug #77111)
- **InnoDB:** The `SysTablespace::parse_units()` function now returns the number of pages in a file instead of the number of megabytes. The `SysTablespace::normalize_size()` function was removed. Error messages in `SysTablespace::parse_params()` were revised. (Bug #21040199, Bug #76949)
- **InnoDB:** For persistent tables, the internal unique identifier for `InnoDB` indexes (`index_id`) now includes a tablespace identifier (`space_id`,`index_id`). This change makes index identifiers unique at the tablespace level as well as the `InnoDB` instance level, and supports future work related to index identifier allocation. (Bug #20737524, Bug #76392)
- **InnoDB:** In `mtr0mtr.cc`, a redundant function was removed, and the `ReleaseBlocks` function was renamed to `AddDirtyBlocksToFlushList`. (Bug #20735882, Bug #76343)

- **InnoDB:** Code related to `innochecksum` was cleaned up and reorganized. Checksum functionality is now located in `buf0checksum.cc`. (Bug #20518099)
- **InnoDB:** `__attribute__((nonnull))` was removed from `InnoDB` code. The attribute is no longer permitted by `InnoDB` coding guidelines. (Bug #20468234)
- **InnoDB:** A new struct was added to provide a logical interface for handling and manipulating external `BLOB` field references. (Bug #18195972)
- **InnoDB:** `TRUNCATE TABLE` is now mapped to `DROP TABLE` and `CREATE TABLE`. This change has the following implications:
  - On systems with a large buffer pool and `innodb_adaptive_hash_index` enabled, `TRUNCATE TABLE` operations previously caused a temporary drop in system performance due to an LRU scan that occurred when removing the table's adaptive hash index entries. The remapping of `TRUNCATE TABLE` to `DROP TABLE` and `CREATE TABLE` avoids the problematic LRU scan.
  - `TRUNCATE TABLE` is temporarily non-atomic. A server exit during a `TRUNCATE TABLE` operation can result in a dropped table and orphaned foreign key constraints in the `InnoDB` `SYS_FOREIGN` and `SYS_FOREIGN_COLS` system tables.
  - The `InnoDB` `memcached` plugin `flush_all` command invokes `DELETE` instead of `TRUNCATE TABLE`. `DELETE` has a higher overhead cost than `FLUSH TABLES` since it involves undo-logging, delete-marking, and eventually purging each deleted row.
  - A log checkpoint that occurred for internal truncate table operations on file-per-table tablespaces was replaced by a log flush.

  (Bug #16834993, Bug #68184, Bug #16207919, WL #6795)
- **InnoDB:** `InnoDB` wasted most pages in extents used for fragment pages. (Bug #16204823, Bug #67963)
- **Partitioning:** In some cases, an issue with partition pruning being attempted a second time during optimization after all partitions had already been pruned at parsing time led to an assert. (Bug #23194259)
- **Partitioning:** A partitioned table whose table name and any partition name had a combined length in excess of 61 characters could not be imported from a backup created using `mysqldump`. When the table also employed subpartitioning, then the combined length of the table name, any partition name, and the name of any subpartition of this partition could not exceed 57 characters without triggering the same issue.

  This was due to the fact that the internal `mysql.innodb_table_stats` table allowed a maximum of 64 characters for the column used to store the table name, even though InnoDB stores, for a partitioned or subpartitioned table, a row in `innodb_table_stats` for each partition or subpartition wherein the value actually used to represent the table name follows the pattern *`table_name`*`#P#`*`partition_name`* or *`table_name`*`#P#`*`partition_name`*`#SP#`*`subpartition_name`*, respectively. This issue is fixed by changing the definition of the `innodb_table_stats` to accommodate the maximum combined length of these attributes plus `#P#` and `#SP#` (199 characters). (Bug #72061, Bug #18416479)
- **Replication:** In `Slave_worker::write_info()`, `DBUG_ENTER()` had "`Master_info::write_info`" as its argument instead of "`Slave_worker::write_info`". This fix corrects the argument. Thanks to Stewart Smith for the patch. (Bug #21658067, Bug #78133)
- **Replication:** When using `START SLAVE UNTIL` *`position`* statements with a multithreaded slave the only `UNTIL` clause available was `SQL_BEFORE_GTIDS`. Now multithreaded slaves are compatible

with the `START SLAVE UNTIL MASTER_LOG_FILE = 'log_name', MASTER_LOG_POS = log_position` and `START SLAVE UNTIL RELAY_LOG_FILE = 'log_name', RELAY_LOG_POS = log_position` statements. (Bug #75843, Bug #20513547)

- **Microsoft Windows:** On Windows, setting the global `log_syslog` system variable in multiple threads could cause a server exit. (Bug #22180046)
- **Solaris:** The client library failed to build on Solaris using the `Cstd` library. (Bug #24353920, Bug #82347)
- **JSON:** `CHECKSUM TABLE` calculated the checksums for `JSON` values using the memory addresses of the values rather than the values themselves, which made the checksum vary. Now in such cases the calculation is based on the actual `JSON` value, and not on that value's address. (Bug #23535703)
- **JSON:** Passing `NULL` to a stored procedure expecting a `JSON` parameter led to an assertion failure in debug builds. (Bug #23209914)
- **JSON:** Parsing of `JSON` path arguments failed to distinguish between a NULL path and one that was syntactically invalid.

  This has been changed so that parsing of these paths now clearly distinguishes between valid non-NULL paths, NULL paths, and invalid paths. (Bug #22816576)
- **JSON:** For debug builds, an assertion could be raised when the server created a temporary table to hold `JSON` objects. (Bug #22782948)
- **JSON:** Queries that executed a `JSON` function that raised an error could cause a server exit. (Bug #22253965)
- Renaming a table to be part of a nonexistent database failed (correctly), but with an `Unknown error` message. A proper error message is now produced; this was corrected as part of the data dictionary implementation. (Bug #25167507, Bug #84000)
- For segmentation faults on FreeBSD, the server did not generate a stack trace. (Bug #24566529, Bug #23575445, Bug #81827)
- On macOS, stack trace demangling now occurs for builds compiled using Clang, just as for GCC. (Bug #23606094, Bug #81908)
- `libevent` was built on macOS even when not needed. (Bug #23228287, Bug #81311)
- A function that returns a `JSON` value could cause a server exit if called as part of a `CASE` statement in a stored procedure. (Bug #23212765)
- Previously, different values were reported by `SHOW ENGINE PERFORMANCE_SCHEMA STATUS` and `SELECT * FROM performance_schema.memory_summary_global_by_event_name` for total memory used in the Performance Schema. The memory for scalable buffer pages, instrumented as `memory/performance_schema/scalable_buffer`, was missing from the `SHOW ENGINE STATUS` output. That statement now includes the missing memory, displayed as `(pfs_buffer_scalable_container).memory`. (Bug #23104498)
- The `-fexpensive-optimizations` option to GCC caused ARM64 and PowerPC builds to compute floating-point operations slightly differently from other platforms. This option was enabled by `-O2` and higher optimization levels. The option now is disabled on platforms negatively affected by it. (Bug #23046775)
- After a failed administrative operation such as `ALTER TABLE ... OPTIMIZE PARTITION`, selecting from the Performance Schema in lock-tables mode could hang. (Bug #23044286)

- In builds with AddressSanitizer enabled, `CAST(... AS BINARY)` could cause a server exit. (Bug #22900560)
- Some spatial functions were reported using a different spatial function name in error messages. (Bug #22883056, Bug #80627)
- Fixed Valgrind warnings with Clang in optimized mode for the `my_strtod_int()` function. (Bug #22839888)
- Precision math operations on values with 64 decimals could produce a 0 result. (Bug #22828692)
- `ST_GeomFromGeoJSON()` could return an error with valid arguments. (Bug #22804853)
- For debug builds with `STRICT_TRANS_TABLES` SQL mode enabled, an assertion could be raised by `INSERT` or `REPLACE` statements that had made changes that could not be rolled back and that subsequently generated an `ER_NO_DEFAULT_FOR_FIELD` error. (Bug #22635253)
- The optimizer contained a `memcpy()` call that did not check for overlapping source and destination. (Bug #22537196)
- For debug builds, a missing error check on the result of a subquery that accessed a `JSON` value could raise an assertion. (Bug #22522073)
- Preparing a `CREATE TABLE ... SELECT` statement, then flushing tables (thus closing the table) and executing the prepared statement could cause a server exit. (Bug #22393309)
- A prepared statement that used a parameter in the select list of a derived table that was part of a join could cause a server exit. (Bug #22392374, Bug #24380263)
- Some grant tables did not account for the increase in maximum user name length from 16 to 32 characters in MySQL 5.7.8. (Bug #22379607, Bug #79680)
- Re-evaluation of a generated column expression could cause access to previously freed memory and a server exit. (Bug #22346120)
- `HANDLER` read statements that searched an index when the target index value was not stored into the row buffer successfully could cause a server exit. (Bug #22321965)
- Improper handling of numeric-to-`ZEROFILL` conversion for `NULL` values could lead to a server exit. (Bug #22281205)
- Using a subquery containing a row constructor to set a variable in a `SET` statement could cause a server exit. (Bug #22276843)
- If the SQL mode did not include `ALLOW_INVALID_DATES`, a query that contained *`invalid_date`* `IN (`*`subquery`*`)` and was handled by subquery materialization could cause a server exit. (Bug #22262843)
- For the embedded server, the code following the check for invalid arguments was invoked with missing or incorrect arguments, which could lead to an improper exit. (Bug #22262706)
- On OS X, `vio_io_wait()` used `select()`, limiting the number of file descriptors to 1024. Now kqueue event notification is used instead to avoid this limit. FreeBSD was changed to use kqueue as well. (Bug #22244911)
- Memory leaks could result if stored routine loading involved temporarily changing the default database and errors occurred restoring the original default database. (Bug #22179795)
- `CMake` configuration was adjusted to check for `-W`*`xxx`* compiler options instead of `-Wno-`*`xxx`* because the latter produce false positives for GCC. (Bug #21881753)

- There could be discrepancies between the values of `INFORMATION_SCHEMA.EVENTS.LAST_EXECUTED` and `mysql.event.last_executed`. This no longer occurs. Event information is stored in the `mysql.events` data dictionary table, which is invisible, so that `INFORMATION_SCHEMA.EVENTS` is the sole interface to event metadata. (Bug #21374010)
- Views could evaluate user-defined or SQL functions before evaluating restrictions from the view definition. (Bug #20933307)
- With `-DENABLE_DTRACE=ON`, `CMake` did not check whether a working DTrace installation was present. Now it checks and aborts if DTrace cannot be found. (Bug #20671056)
- `SHOW CREATE USER` returned unexpected results. (Bug #20625566)
- If given a relative path name for the `--log-error` option, `mysqld` could sent `stdout` and `stderr` to the wrong location. (Bug #20609063)
- Evaluation of `LEAST()` and `GREATEST()` could use too small a sort buffer for datetime and string literals, causing an assertion to be raised. (Bug #20565160)
- The range of error numbers for errors that are new in MySQL 8.0 has been designated to begin with 3500. (Bug #20538173)
- Debian packaging was updated not to set the `sql_mode` system variable in `my.cnf`. (Bug #20535729)
- Event loading from the `mysql.event` system table could fail if the `PAD_CHAR_TO_FULL_LENGTH` SQL mode was enabled. (Bug #20073523, Bug #74947)
- Statements such as `INSERT` and `LOAD DATA` that use the `REPLACE` or `IGNORE` keyword to handle duplicate records could affect subsequent operations. (Bug #20017428)
- `CREATE TABLE ... SELECT` where non-`BIT` data was selected from the source table into a `BIT` column in the destination table could cause a server exit. (Bug #19930894)
- Compilation failed on OS X when MySQL was configured with `-DMYSQL_MAINTAINER_MODE=1` and compiled with clang/Xcode 6.0. (Bug #19694515, Bug #74100)
- For `CHANGE MASTER TO` statements rewritten to filter the password before being written to the general query log, any `MASTER_AUTO_POSITION` clause was lost. (Bug #19622609)
- Use of the `VALUES()` function in a `SELECT` clause could result in a server exit. (Bug #19601973)
- A potential null-pointer dereference and memory leak in table-rename code were corrected. (Bug #18194270)
- Using `MATCH ... AGAINST` to compare a character column and an aggregate function could cause a server exit. (Bug #17865492)
- Slightly different values for the number of connections could be reported in various information sources, such as the `Connections` status variable, Performance Schema `threads` and `global_status` tables, and `SHOW PROCESSLIST` statement. (Bug #17666696)
- A query with a subquery containing a set operation with an outer reference might cause a server exit. (Bug #17270896)
- Using `GRANT` to change a password for an invalid user produced an error, but also updated the `mysql.user` system table. (Bug #17180985)
- The parser for spatial WKT data accepted numbers such as `0.23` but not `.23`, the equivalent value without the leading zero. Now both formats are accepted. (Bug #17167633)

- Previously, if a client attempted to send connection attribute key/value pairs that in aggregate had a size larger than the value of the `performance_schema_session_connect_attrs_size` system variable, the Performance Schema truncated the attribute data. In addition, the Performance Schema wrote this message to the error log if the `log_warnings` system variable was greater than zero:

  ```
  [Warning] Connection attributes of length N were truncated
  ```

  This message was not helpful to a DBA attempting to determine the problematic client, so several changes have been made to connection attribute handling:

  - Truncation of connection attributes still occurs for excessive data, but the log message is more informative. It includes the number of bytes lost, the connection identifier, and information about the client user. The additional information should enable DBAs to more easily identify clients for which attribute truncation occurred.
  - When truncation occurs, a `_truncated` attribute is added to the session attributes with a value indicating how many bytes were lost, if the attribute buffer has sufficient space. This enables the Performance Schema to expose per-connection truncation information in the connection attribute tables.
  - A new status variable, `Performance_schema_session_connect_attrs_longest_seen`, indicates the longest connection attribute buffer smaller than 64KB seen by the server. If this value is larger than `performance_schema_session_connect_attrs_size`, attribute truncation has occurred, and DBAs may wish to increase the latter value, or, alternatively, investigate which clients are sending large amounts of attribute data.

  For more information, see Performance Schema Connection Attribute Tables. (Bug #16576959)

- Introduction of the data dictionary enables several `INFORMATION_SCHEMA` problems to be addressed:

  - Queries on `INFORMATION_SCHEMA.STATISTICS` could return different results depending on the order of columns in the select list.
  - Some `INFORMATION_SCHEMA` tables had suboptimal column types and sizes. Such tables that are now views on data dictionary tables in the `mysql` system database have more appropriate column definitions.
  - Queries on `INFORMATION_SCHEMA` tables that resulted in directory scans to determine database or file names no longer do so, but instead read database and table names from the data dictionary.
  - Queries on `INFORMATION_SCHEMA` tables that opened `.frm` files to obtain table metadata no longer do so, but instead read this information from the data dictionary.
  - For comparisons of database or table names in `INFORMATION_SCHEMA` queries, using `COLLATE` to force a given collation worked only if applied to the `INFORMATION_SCHEMA` table column, but not if applied to the comparison value.

    For additional information about performing such comparisons, see Using Collation in INFORMATION_SCHEMA Searches.

  (Bug #14017351, Bug #65121, Bug #17559183, Bug #70462, Bug #23259470, Bug #81347, Bug #20372562, Bug #75532, Bug #13878164, Bug #11756519, Bug #48445)

- For abnormal server exit on Windows, the server previously created a minidump file named *`module_name`*`.dmp`, where *`module_name`* is the name of the server executable file. To prevent earlier minidump files from being overwritten, minidump file names now include the process ID and have the form *`module_name`*`..`*`pid`*`dmp`; for example, `mysqld.exe.7296.dmp`. (Bug #12779463)

- For queries on `INFORMATION_SCHEMA` tables, comparisons of schema and table names could be case sensitive or insensitive, depending on the characteristics of the underlying file system and the `lower_case_table_names` system variable value. Furthermore, it was ineffective to provide a `COLLATE` clause to change the comparison properties because that clause was ignored. This has been changed so that `COLLATE` is no longer ignored and can be used to obtain the desired comparison properties. (Bug #11748044, Bug #34921)
- `FLOOR(CEIL())` truncated large `BIGINT UNSIGNED` arguments. (Bug #80873, Bug #23013359)
- Manipulation of a value returned by the `JSON_MERGE()` function using `JSON_SET()` sometimes produced an invalid result. (Bug #80787, Bug #22961128)
- `ST_AsGeoJSON()` failed when geometry arguments were supplied using user-defined variables. (Bug #80697, Bug #22912800)
- `CAST(`*`expr`*` AS BINARY(`*`N`*`) )` unexpectedly returned `NULL` for some valid values of *`N`*. (Bug #80630, Bug #22885819)
- Geometry import functions that took an SRID parameter cast it to an unsigned 32-bit integer without warning or error, so negative values or values larger than unsigned 32-bit integer range were silently converted to a number within the range. Now, all geometry functions that take the SRID as a parameter check that it is within unsigned 32-bit integer range and produce an `ER_DATA_OUT_OF_RANGE` error if not. This also applies to GeoJSON and GeoHash functions that previously checked that the parameter was within range but returned a different error code. (Bug #80499, Bug #22819614)
- If rounding occurred while storing a predicate value, the range optimizer might not return correct results for the `<` and `<=` operators. (Bug #80244, Bug #22661012)
- For the `mf_iocache` unit test, add a missing `va_end()`, fix a memory leak by calling `my_end()`, and add a target for the test. Thanks to Daniel Black for the patch on which these changes were based. (Bug #80085, Bug #22578670)
- `SELECT DISTINCT SUBSTR()` could incorrectly discard values as duplicates for large position or length arguments. The same issue also affected `LEFT()` and `RIGHT()`. (Bug #80047, Bug #22565155)
- `SUBSTRING_INDEX(`*`str, delim, count`*`)` did not properly handle *`count`* values larger than 32 bits. (Bug #79978, Bug #22545429)
- For calls to `CONVERT()`, literal string arguments could be modified during execution, producing incorrect results. (Bug #79924, Bug #22531111)
- `REPLACE('a', BINARY 'b', NULL)` returned `'a'` rather than `NULL`. (Bug #79912, Bug #22523836)
- Transaction state tracking now avoids a function that is not 8-bit safe, for enhanced compatibility with nonstandard character sets. (Bug #79905, Bug #22523383)
- `GREATEST()` and `LEAST()` treated all integer input as signed. (Bug #79902, Bug #22523685)
- Lines in the general query log were missing a tab between the timestamp and the thread ID. Thanks to Tsubasa Tanaka for the patch. (Bug #79868, Bug #22508563)
- Some comparisons between unsigned values and negative upper limits could return incorrect results (for example, `CAST(100 AS UNSIGNED) BETWEEN 1 AND -1`). (Bug #79857, Bug #22501606)
- The `REPEAT()` function did not properly handle output from the `SUBSTR()` function. (Bug #79695, Bug #22391186)
- The `JSON_TYPE()` function now shows the type of `BIT` literals cast to JSON as `BLOB`, rather than `BIT`. (Bug #79308, Bug #22297987)

- Configuring MySQL with the `-DWITH_UBSAN=ON CMake` option produced a server that was not fully functional. (Bug #79238, Bug #22194071)

- `sql_common.h`, a header file included in MySQL distributions, included and was therefore dependent on `hash.h`, a header file not included in MySQL distributions. This resulted in compilation failures. To eliminate this dependency, `sql_common.h` was modified to no longer include `hash.h`. (Bug #79237, Bug #22187997, Bug #70672, Bug #17633467)

- Timers used for checking maximum statement execution time were initialized even when the server was started with the `--help` option. If `--help` is given, this is no longer done. (Bug #79182, Bug #22172389)

- The optimizer failed when trying to optimize away expressions of the form `IF(true, '2015-01-01', '2015-01-01') IS NOT NULL`. (Bug #79114, Bug #22148586)

- Subtraction of an unsigned decimal could return a negative value, but with metadata type information of `UNSIGNED BINARY`. Subtraction for unsigned decimal subtraction now is handled the same way as for unsigned integer: Produce an `ER_DATA_OUT_OF_RANGE` error if the result is negative, unless the `NO_UNSIGNED_SUBTRACTION` SQL mode is enabled. (Bug #78914, Bug #22083757)

- Handling by the `HEX()` function of numbers larger than $2^{64}$ was improved. (Bug #78828, Bug #22297983)

  References: This issue is a regression of: Bug #9854.

- The client-side plugin deinitialization function signature was changed from `int (*deinit)()` to `int (*deinit)(void)` to avoid warnings when compiling with `-Wstrict-prototypes`. (Bug #78177, Bug #21680094, Bug #81419, Bug #23282498)

- `CREATE TABLE` reported an incorrect error if a very long or incorrect path name was specified for the `DATA DIRECTORY` or `INDEX DIRECTORY` table option. Now `ER_PATH_LENGTH` or `ER_WRONG_VALUE` are reported for those cases. (Bug #76635, Bug #20857556)

- The server now tries to provide more informative messages for these error codes: `ER_CANT_CREATE_DB`, `ER_CANT_CREATE_TABLE`, `ER_DB_DROP_RMDIR`, `ER_ERROR_DURING_COMMIT`, `ER_ERROR_DURING_ROLLBACK`, `ER_GET_ERRNO`. (Bug #76298, Bug #20694494)

- For some instances of failure to prepare an XA transaction, incomplete transaction cleanup could raise an assertion. (Bug #75809, Bug #20488921)

- `mysqld` could attempt to close an invalid socket file descriptor. Thanks to Zhai Weixiang for the patch. (Bug #75778, Bug #20504513)

- A statement of the following form converted the table data to `latin1`, but also changed the table default character set to `latin1` and ignored the `utf8` clause:

  ```
  ALTER TABLE tbl_name CHARACTER SET utf8, CONVERT TO CHARACTER SET latin1;
  ```

  Thanks to Daniel Black for the patch. (Bug #75320, Bug #20279241)

- In `mysqld.cc`, the `abort_loop` variable was quantified with `volatile`, which on some platforms could result in changes not being seen immediately in threads running on different cores. Thanks to Stewart Smith for the patch. (Bug #74846, Bug #20134637)

- Calling a procedure which created a view from a trigger, or creating a function that called a procedure that executed `RENAME TABLE` could, under certain circumstances, raise an assertion. (Bug #74740, Bug #19988193, Bug #21198646)

- Timestamps for server-side prepared statements could be written to the binary log up to a second behind timestamps for the corresponding nonprepared statements, leading to time value differences between master and slave servers. (Bug #74550, Bug #19894382, Bug #25187670)
- For dynamic storage engine plugins, `DROP TABLE`, `TRUNCATE TABLE`, and `RENAME TABLE` did not work due to incorrectly determining the engine from the `.frm` file. (Bug #74277, Bug #19902868)
- Executed prepared statements are logged with `?` parameter markers replaced by data values. Construction of the logged string was inefficient and has been improved. (Bug #73056, Bug #20955496)
- Assignment by a plugin to its thread variables of string type could leak memory. (Bug #71759, Bug #19917521)
- Grouping with a view could produce an `ER_INVALID_GROUP_FUNC_USE` error ("Invalid use of group function") when selecting from the base table did not. (Bug #70220, Bug #17406425)
- Queries with multiple `LIMIT` clauses were not always interpreted in accordance with the SQL standard.

  For example, each of the following queries now returns two rows, as expected:

  ```
  SELECT * FROM t LIMIT 2;

  (SELECT * FROM t LIMIT 2) LIMIT 4;

  ((SELECT * FROM t LIMIT 2) LIMIT 4) LIMIT 3;
  ```

  This also fixes an issue with `UNION` subselects using `FROM DUAL`; for example, the following query raised an error:

  ```
  SELECT 'A' AS `x` FROM DUAL
    UNION
  SELECT 'B' FROM DUAL
    UNION
  SELECT 'C' FROM DUAL
    ORDER BY `x` ASC;
  ```

  (Bug #50510, Bug #74251, Bug #11758322, Bug #19774083)
- Test cases that were intended to be storage engine-agnostic but were actually using a specific engine were corrected. (WL #7264)

# Index

## Symbols

!, 334
!include, 448, 507
!includedir, 448, 507
#define, 114
$, 71
%, 45
&>, 334
* (asterisk), 247
--abort-slave-event-count, 114
--admin-ssl option, 171
--basedir, 529
--binary-as-hex, 290, 408, 529
--binary-as-hex option, 211

--character-set-client-handshake, 45
--column-statistics, 529
--comments, 529
--compress, 71, 334
--compression-algorithms, 71
--daemonize, 529
--default-time-zone, 150
--des-key-file, 507
--disabled-storage-engines, 247
--disconnect-slave-event-count, 114
--early-plugin-load, 334
--gdb, 432, 448
--get-server-public-key, 472
--help mysqld option, 171
--hex-blob, 432
--initialize, 529
--keyring-migration-to-component option, 195
--log-raw, 290
--log-warnings, 507
--loose, 36
--malloc-lib, 247
--mtr-port-exclude, 247
--network-namespace, 228
--network-timeout, 554
--new, 45
--no-beep, 408
--no-dd-upgrade, 360
--no-monitor, 432
--old, 45
--secure-auth, 507
--server-public-key-path, 472
--set-gtid-purged, 71
--single-transaction, 71
--skip-comments, 529
--skip-grant-tables, 408, 432, 507
--skip-grant-tables mysqld option, 247
--skip-host-cache, 98
--skip-log-bin, 472
--skip-symbolic-links, 472
--ssl, 597
--ssl option, 171
--ssl-cipher, 472
--ssl-fips-mode, 360, 408
--ssl-mode, 597
--ssl-mode=VERIFY_IDENTITY, 432
--ssl-verify-server-cert, 432, 597
--symbolic-links, 472, 529
--temp-pool, 554
--transaction-isolation, 507
--transaction-read_only, 507
--upgrade, 360
--verbose mysqld option, 171
-1, 247
-DDISABLE_PSI_MEMORY, 171

.mysql_secret, 360
1F, 507
32-bit, 597
=, 290
>ut_allocator(), 171
\u001f, 507
_, 45
` (backtick), 247
||, 334

, 131, 211

## A

Aborted_connects, 360
Abseil, 14
account categories, 360
account management, 290, 360, 408, 448, 529, 597
ACL, 31
Acl_cache, 20
adaptive hash index, 26, 42, 71, 98, 171, 211, 472, 554, 597
ADD COLUMN, 84, 98, 171, 269, 334, 385, 432
ADD INDEX, 269, 314
ADD PARTITION, 84
Address Sanitizer, 195, 211, 247
AddressSanitizer, 448
ADD_GDB_INDEX, 314
add_not_null_conds(), 360
ADMIN, 432
ADMINISTRABLE_ROLE_AUTHORIZATIONS, 290
admin_address, 228, 385
admin_port, 385
admin_tls_version, 131
AES_DECRYPT(), 472
AES_ENCRYPT(), 26
aggregate functions, 195, 195, 211, 290
AggregateIterator, 171
aggregates, 150
aggregation, 114, 131, 150, 195, 211, 334, 472, 554
aggregation functions, 114
AIO, 211
ALGORITHM, 360
ALGORITHM-INSTANT, 98
ALGORITHM=COPY, 597
ALGORITHM=INSTANT, 36, 71, 84, 98, 114, 131, 432
algrind, 114
alias, 290, 360
aliases, 114, 290
ALLOW_INVALID_DATES, 360
Alpine Linux, 448, 472, 529
ALTER COLUMN, 408
ALTER EVENT, 62
ALTER INSTANCE, 247
ALTER INSTANCE RELOAD KEYRING statement, 84, 195
ALTER INSTANCE RELOAD TLS, 62, 247

ALTER RESOURCE GROUP, 432
ALTER SCHEMA, 290
ALTER TABLE, 26, 62, 71, 114, 195, 211, 228, 247, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554, 597
ALTER TABLE ... ALGORITHM=COPY, 36
ALTER TABLE ... IMPORT TABLESPACE, 228
ALTER TABLE statement, 211
ALTER TABLESPACE, 314, 385, 472
ALTER USER, 62, 269, 334, 385, 448
AlternativeIterator, 290
ANALYZE, 42
ANALYZE TABLE, 84, 150, 228, 290, 448, 529
ANALYZE TABLE statement, 195
ANALYZE TABLE UPDATE HISTOGRAM, 408
AND, 290
antijoin, 334
antijoins, 20, 62
ANY, 228
API, 98
AppArmor, 20
append_identifier(), 14
APPLICABLE_ROLES, 290
ARCHIVE, 334, 385
arguments checking, 45
Arg_comparator, 150
ARM, 31
ARM 64-bit, 334, 408
ARRAY, 290, 334
arrays, 472
ASAN, 195, 211, 247
ASC, 408, 432
ASCII, 84, 131
ASCII 31, 507
assert, 42
assertion, 71, 171
ASSERT_COLUMN_MARKED_FOR_READ, 385
asymmetric_encrypt(), 62
async client, 51
asynchronous I/O, 360
asynchronous interface, 26
asynchronous_connection_failover_add_managed(), 14
asynchronous_connection_failover_add_source(), 14
asynchronous_connection_failover_delete_source(), 14
AT TIME ZONE, 228
atomic DDL, 247, 448, 507, 554, 597
atomic operations, 597
atomics, 211
ATTRIBUTE, 247
audit filter, 7
Audit Log, 14
audit log, 62
audit log plugin, 45, 51
audit plugins, 360
AUDIT_ADMIN, 334

audit_api_message_emit, 385
audit_api_message_emit_udf(), 385
audit_log plugin, 45, 51, 62, 71, 84, 98, 131, 150, 171, 195, 228, 247, 290, 314,
334, 360, 385, 408, 432, 448, 472, 554
audit_log_connection_policy, 51
audit_log_disable, 131
audit_log_exclude_accounts, 51
audit_log_filter_id, 247
audit_log_flush, 84
audit_log_format_unix_timestamp system variable, 171
audit_log_include_accounts, 51
audit_log_max_size system variable, 171
audit_log_password_history_keep_days, 334
audit_log_policy, 51
audit_log_prune_seconds system variable, 171
audit_log_read, 7
audit_log_read() function, 228
audit_log_rotate, 71
audit_log_rotate(), 84
audit_log_rotate_on_size system variable, 171
audit_log_statement_policy, 51
audit_null plugin, 334
authentication, 51, 71, 84, 114
authentication plugins, 45, 51, 71, 98, 150, 171, 195, 211, 228, 247, 269, 290, 314,
360, 385, 408, 448, 472, 507
authentication_fido_client plugin, 45, 62
authentication_kerberos, 31
authentication_kerberos authentication plugin, 171
authentication_kerberos_client authentication plugin, 171
authentication_ldap_sasl, 247
authentication_ldap_sasl_log_status, 314
authentication_ldap_sasl_referral system variable, 269
Authentication_ldap_sasl_supported_methods, 247
authentication_ldap_simple_log_status, 314
authentication_ldap_simple_referral system variable, 269
authentication_oci, 51
authentication_oci_client, 62
AUTHENTICATION_PAM_LOG, 42
authenticator data, 114
auth_sock plugin, 448
auth_socket, 71
auto-increment, 314, 507, 529, 597
autocommit, 20, 472
AUTOEXTEND_SIZE, 195, 211
AutomaticThreadConfig, 62
automatic_sp_privileges, 432
autopositioning, 529
autoscale, 507
autowrapping, 507
AUTO_INCREMENT, 62, 114, 247, 334
auto_increment_increment, 334, 385
auto_increment_offset, 385
averages, 31
AVG(), 62, 334, 385, 408

## B

B-tree, 114, 408
backup, 507
back_log, 507
Barracuda, 597
base-64, 432
basedir, 529
batch mode, 334
batched key access, 171
BETWEEN, 114, 408
BETWEEN operator, 211
BIGINT, 290
big_tables, 472
BIN(), 554
BINARY, 150, 334
binary, 408
binary log, 51, 62, 195, 507, 529, 554, 597
binary log compression, 45, 51
binary log events, 51
binary log sender thread, 84
binary logging, 84
bind_address, 228, 408
binlog, 71
binlog_direct_non_transactional_updates, 385
binlog_encryption, 247
binlog_expire_logs_auto_purge, 114
binlog_expire_logs_seconds, 84, 432, 472, 554
binlog_format, 51, 385, 529
binlog_order_commits, 62
binlog_rows_query_log_events, 385
binlog_row_image, 385
binlog_row_value_options, 385, 472, 507
binlog_transaction_dependency_history_size, 554
binlog_transaction_dependency_tracking, 45, 472, 554
BIN_TO_UUID(), 597
Bison, 114
BIT, 84, 131, 597
bit functions, 150
bit operations, 597
bitwise operators, 314
BIT_AND(), 195, 432
BIT_OR(), 195, 334, 432
BIT_XOR(), 195, 432
BKA, 171
BLACKHOLE, 472
BLACKHOLE storage engine, 150
BLOB, 150, 211, 360, 448, 472, 597
bool, 554
boolean, 334
Boost, 114, 150, 228, 247, 314, 334, 360, 385, 432, 448, 472, 529, 554, 597
bootstrap, 472, 529
Bounded_queue, 51
BUCKETS, 448

buffer pool, 84, 150, 171, 195, 228, 269, 408, 432, 448, 472, 507, 554, 597
buffer pool mutex, 597
buffer pool scan, 597
buffers, 51
buf_validate(), 131
bugs, 14
build_id, 84
BUILD_SHARED_LIBS, 26
bulk_insert_buffer_size, 385
BUNDLE_RUNTIME_LIBRARIES, 448

## C

c api, 51
C API, 98, 131, 150, 211, 228, 247, 269, 334, 360, 472, 507, 529, 554, 597
C++11, 597
C++14, 360
C++17, 150
C++23, 45
CACHE INDEX, 334
caching_sha2_password, 360, 448, 472, 507
caching_sha2_password plugin, 195
caching_sha2_password_auto_generate_rsa_keys, 472
caching_sha2_password_digest_rounds, 4
caching_sha2_password_digest_rounds system variable, 195
capture groups, 432
CAP_SYS_NICE, 385
CASE, 150, 195, 334, 554
CASE function, 62
case sensitivity, 269
CAST, 84, 195, 432
cast injection, 247, 314
CAST(), 84, 150, 195, 228, 290, 314, 334, 597
CAST() function, 195
CATS, 408
CDATA, 247
CEIL(), 597
CEILING(), 290
change buffer, 247, 314, 597
change buffering, 472
CHANGE MASTER TO, 71, 228, 472
CHANGE MASTER TO statement, 195
CHANGE REPLICATION FILTER, 529
channel initialization, 529
CHAR, 385
CHAR(), 432
character escapes, 507
CHARACTER SET, 150
character sets, 62, 84, 195, 211, 269, 290, 314, 334, 360, 385, 408, 448, 472, 507,
529, 554, 597
character_set_client, 334
character_set_connection, 472
character_set_database, 385, 554
character_set_filesystem, 385

character_set_server, 36, 554
charsets, 51, 62, 150, 171, 195, 408
CHECK constraints, 171, 269, 290, 360
CHECK TABLE, 360, 385, 408
check-table-functions, 14
checkpoint, 408
checksum, 408, 597
checksum algorithm, 150
checksum mismatch, 432
CHECKSUM TABLE, 597
CHECK_CONSTRAINTS, 360
check_stack_overrun(), 554
Chinese, 360
Clang, 7, 9, 26, 71, 131, 150, 228, 269, 529, 597
clang, 51, 98
client metadata, 36
client/server protocol, 171
CLIENT_NO_SCHEMA, 71
CLIENT_SSL_VERIFY_SERVER_CERT, 448
clock_gettime(), 290
clone, 114
clone plugin, 36, 114, 150, 171, 195, 228, 269, 290, 314, 334
clone_delay_after_data_drop, 114
clone_os_copy_file_to_buf(), 71
Clone_Snapshot, 45
clone_status table, 51
CMake, 20, 42, 45, 84, 98, 290, 334, 360, 385, 408, 448
cmake, 62, 84, 98
CMake 4, 9
CMAKE_BUILD_TYPE, 385, 408
CMAKE_MINIMUM_REQUIRED, 26
CMAKE_SKIP_INSTALL_ALL_DEPENDENCY, 26
COALESCE, 360
COALESCE(), 472
coercibility, 247
COLLATE, 62, 150, 507
collation-server, 62
collations, 51, 84, 98, 150, 195, 211, 228, 247, 269, 290, 314, 334, 360, 408, 448,
507, 529, 554, 597
collation_database, 385, 554
collation_server, 554
column prefixes, 45
column references, 26, 51, 269
COLUMNS table, 211
columns_priv table, 597
COLUMN_FORMAT, 334
column_statistics table, 98, 507, 529
COLUMN_STATISTICS table, 529
column_stats table, 529
COMMENT, 247
comments, 9, 31, 269, 290, 408, 529
common table expressions, 98, 114, 150, 247, 269, 290, 314, 385, 472, 507, 554
compare_pair_for_nulls(), 51
comparisons, 171, 247

COMPILATION_COMMENT, 385
COMPILATION_COMMENT_SERVER, 360, 385
compile_cluster, 334
COMPILE_DEFINITIONS, 360
COMPILE_FLAGS, 360
compiling, 4, 9, 36, 42, 45, 51, 62, 71, 84, 98, 114, 131, 150, 171, 195,
195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554,
597
component service, 84, 171
components, 26, 71, 84, 247, 334, 385, 448, 472, 529, 554, 597
component_enterprise_encryption, 62
component_keyring_encrypted_file component, 51, 195
component_keyring_file component, 51, 195
component_keyring_oci, 71, 84
component_keyring_oci component, 84
component_masking component, 62
component_masking_functions component, 62
component_sys_variable_register service, 529
component_sys_variable_unregister service, 529
composite indexes, 385
Compression, 71
compression, 247, 290, 314, 507, 554, 597
COMPRESS_DEBUG_SECTIONS, 228
comp_err, 195, 247, 269, 334
COM_SHUTDOWN, 597
CONCAT(), 247, 360, 472
CONCAT_WS(), 247
concurrent DDL, 84
condition count, 228
condition pushdown, 14, 20, 36, 62, 71, 84, 98, 114, 131, 150, 314, 360
COND_FILTER_STALE, 334
configuration, 228, 247, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554, 597
connection, 71
connection attributes, 290, 334
connection handling, 195, 597
connection management, 228, 247
connections, 9, 26, 31, 71
CONNECTION_CONTROL, 448
CONNECTION_CONTROL plugin, 269, 554
CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS plugin, 554
CONNECTION_ID(), 98
connection_memory_chunk_size, 51
Connectors, 131
const, 26, 228, 290, 334, 360, 432
constant expressions, 314
constant folding, 150, 314, 360
constant propagation, 150, 472
constraints, 290, 314, 334
Contention-Aware Transaction Scheduling (CATS), 269, 314, 334, 360, 408, 432, 448
CONTRIBUTING.md, 26
contribution, 290
contributions, 114
CONV(), 290
conversions, 385

CONVERT TO CHARACTER SET, 334
CONVERT(), 62, 131, 228, 334, 408, 597
CONVERT_TZ(), 131
copy functions, 171
core file, 385
correlated subqueries, 195
corruption, 597
cost model, 554
cost tables, 554
COUNT(), 195, 334, 360, 385, 472, 529
COUNT(*), 408
COUNT_SECONDARY, 114
crc32, 597
CREATE DATABASE, 150
CREATE EVENT, 62
CREATE FUNCTION, 114, 529
CREATE INDEX, 131, 432, 554
CREATE MASTER, 597
CREATE PROCEDURE, 114
CREATE RESOURCE GROUP, 432
CREATE ROLE, 45
CREATE SPATIAL REFERENCE SYSTEM, 432, 472
CREATE TABLE, 9, 114, 195, 314, 334, 360, 385, 432, 448, 472, 507, 529, 554, 597
CREATE TABLE ... SELECT, 247, 448
CREATE TABLE ... SELECT statement, 150
CREATE TABLE AS SELECT, 26
CREATE TABLE LIKE, 448, 472
CREATE TABLE SELECT, 9, 448
CREATE TABLE statement, 211
CREATE TABLESPACE, 195, 247, 290, 334, 385, 529
CREATE TEMPORARY TABLE, 472, 597
CREATE TRIGGER, 114
CREATE USER, 36, 62, 84, 150, 228, 314, 334, 360, 385, 448, 507
CREATE VIEW, 150, 228, 314, 448, 472
create_admin_listener_thread, 228, 269, 385
CREATE_OPTIONS, 472
crypto policy, 62
CSV, 171
CTE, 71, 114, 171
CTEs, 42, 45, 51, 62, 98
cte_max_recursion_depth, 472, 507
CTX_NONE, 472
CTX_WINDOW, 472
curl, 9, 14, 20, 26, 45, 51, 62, 71, 84, 98, 131, 171, 195, 195, 269, 290,
314, 360, 408
CURRENT_ROLE(), 448
CURRENT_USER, 360
CURRENT_USER(), 51
CURSOR_TYPE_READ_ONLY, 448

## D

daemon_keyring_proxy_plugin plugin, 195

data dictionary, 62, 71, 131, 211, 247, 269, 314, 334, 360, 385, 385, 408, 432, 448, 472, 507, 529, 554, 597
data dictionary cache, 529
DATA DIRECTORY, 247, 385
data encryption, 114, 171, 290
data file, 597
data masking, 62, 114, 150, 408
data types, 71, 290, 314, 408, 507, 597
DATABASE(), 26
data_locks table, 31, 36, 385, 408
data_locks_waits table, 31
data_lockwaits table, 408
data_masking, 71, 98
data_masking plugin, 62
DATE, 114, 195, 314, 334, 360, 448
date arithmetic, 472
DATETIME, 114, 195, 360, 448
datetime, 290
datetime_format, 507
DATE_ADD(), 131
date_format, 507
DATE_SUB(), 131
db table, 597
DBL_MAX, 290
DBSPJ, 269
DBUG package, 314
DBUG_ABORT(), 408, 529
DDL, 9, 14, 51, 114, 131, 150, 228, 360, 408, 432, 448, 472, 507, 554
DDL log, 247
ddl_rewriter, 360
dd_properties table, 529
de-identification, 62
deadlock, 36, 150, 171, 247, 269, 290, 314, 385, 432, 448, 597
deadlock checks, 269
deadlock detection, 385
debug, 31, 36, 385, 408, 432, 448, 472, 597
Debug Sync, 554
debugging, 314, 334
debug_sync, 385
dec, 131
DECIMAL, 51, 114, 290, 314, 334
decimal precision, 150
decimals, 195
DECODE(), 507
deduplication, 269, 334
DEFAULT, 7, 51, 269, 334, 408
DEFAULT ROLE, 314
default values, 247, 408
DEFAULT(), 114, 150
DEFAULT() function, 62
default_authentication_plugin, 150, 472
default_collation_for_utf8mb4, 385, 448
default_roles, 247
default_table_encryption, 360

definer, 36
degenerate hash joins, 247
deinitialization, 36
delay, 31
DELETE, 71, 131, 150, 247, 360, 385, 432, 472
DELETE IGNORE, 247
DENSE_RANK(), 62, 472
DENSE_RANK() OVER, 472
deprecation, 6, 26, 51, 62, 71, 84, 98, 114, 131, 150, 171, 195, 211, 228, 247,
269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554, 597
deprecations, 51
derived condition pushdown, 228
derived tables, 62, 84, 98, 195, 314, 334, 360, 385, 472, 554
DESC, 195, 408, 432
descending indexes, 554
DESCRIBE, 45, 448
DES_DECRYPT(), 507
DES_ENCRYPT(), 507
detached transactions, 114
dictionary object cache, 597
dict_sys, 171
disabled_storage_engines, 432
DISABLE_PSI_MEMORY, 211
DISABLE_SHARED, 314
DISCARD OLD PASSWORD, 385
DISCARD TABLESPACE, 84, 131, 290, 314, 432
disk is full, 472
disk space, 472
display width, 334
display widths, 290
DISTINCT, 42, 114, 150, 171, 211, 247, 269, 290, 314, 334, 360, 432, 529
DIV operator, 195
division, 150
DML, 9, 114, 131, 150, 195, 228, 597
Docker, 150, 195, 360, 385, 507
Docker image, 247
dollar sign, 71
DOM, 554
DOUBLE, 114, 195, 334
double, 554
doublewrite, 290
doublewrite buffer, 26, 62, 98, 171, 247, 269
downgrade, 448
downgrades, 45
Doxygen, 62, 114, 211
doxygen, 448, 597
dragnet.log_error_filter_rules, 432, 472
dragnet.Status, 432
DROP COLUMN, 71, 84, 98, 114, 269
DROP DATABASE, 171, 247, 432, 472, 507, 554
DROP DATAFILE, 529
DROP FUNCTION, 529
DROP INDEX, 26, 554
DROP PARTITION, 228

DROP privilege, 211
DROP RESOURCE GROUP, 432
DROP ROLE, 45, 360
drop role, 360
DROP SPATIAL REFERENCE SYSTEM, 472
DROP TABLE, 84, 98, 211, 247, 432, 507, 529, 554
DROP TABLESPACE, 131, 314, 529
DROP USER, 20, 228
DROP VIEW, 20, 507
DST, 334
DTrace, 554
dual passwords, 385
DUMP 103003, 314
DUMP 103004, 314
DUMP 103005, 314
duplicate key error, 290, 334
duplicate key errors, 9
duplicate keys, 334, 507
duplicate weedout, 269
dynamic metadata, 448
DYNAMIC_STRING, 228

## E

EL7, 51
embedded server, 554
empty string, 597
empty transactions, 14
ENABLED_ROLES, 290
ENABLE_DOWNLOADS CMake option, 171
ENCODE(), 507
ENCRYPT(), 507
encryption, 98, 114, 131, 150, 171, 195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 432, 448, 472, 507, 529, 554
end-range condition, 472
ENUM, 150, 385, 529
EOF, 51
epoch, 269
EPSG, 432, 472
equi-join, 314
equi-joins, 472
equijoins, 84
eq_ref, 84, 114, 334, 360
error handling, 84, 114, 150, 195, 211
error log, 171, 448
error logging, 98, 269, 472, 529
error messages, 62, 314
errors, 14, 20, 26, 31, 36, 62, 71, 84, 131, 150, 171, 195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554, 597
error_log, 228
ER_DUP_ENTRY, 334
ER_FILESORT_TERMINATED, 290
ER_FILSORT_ABORT, 290
ER_GEOMETRY_PARAM_LATITUDE_OUT_OF_RANGE, 432

ER_GEOMETRY_PARAM_LONGITUDE_OUT_OF_RANGE, 432
ER_IB_MSG_720, 385
ER_INCONSISTENT_ERROR, 529
ER_INVALID_CHARACTER_STRING error, 171
ER_JSON_SCHEMA_VALIDATION_ERROR_WITH_DETAILED_REPORT, 290
ER_LATITUDE_OUT_OF_RANGE, 432
ER_LONGITUDE_OUT_OF_RANGE, 432
ER_NEED_REPREPARE, 385
ER_REGEXP_ILLEGAL_ARGUMENT, 360
ER_REGEXP_INVALID_CAPTURE_GROUP_NAME, 448
ER_REGEX_NUMBER_TOO_BIG, 228
ER_WRONG_FIELD_WITH_GROUP, 360
ER_WRONG_VALUE_FOR_VAR, 385
ESCAPE, 195
Event Scheduler, 84, 131, 171, 228, 529
EventMutex, 228
events, 9, 290
events table, 597
events_statements_current, 385
events_statements_history, 385
events_statements_history_long, 385
events_statements_summary_by_digest, 507
event_scheduler, 507
EXCEPT, 84
exception handling, 150
EXCHANGE, 334, 448
EXCHANGE PARTITION, 211, 290, 360, 385
EXECUTION_ENGINE, 114
EXISTS, 51, 62, 150, 195, 247, 360
expired passwords, 150
expire_logs_days, 432, 472
EXPLAIN, 14, 31, 71, 98, 114, 131, 150, 228, 247, 269, 314, 334, 360, 408, 432
EXPLAIN ANALYZE, 62, 84, 98, 171, 247, 269, 290, 314
EXPLAIN EXTENDED, 448, 507
EXPLAIN FOR CONNECTION, 385
EXPLAIN FORMAT=JSON, 314
EXPLAIN FORMAT=TREE, 62, 269
EXPLAIN PARTITIONS, 507
EXPLAIN statement, 171
explicit_defaults_for_timestamp, 228, 385, 472, 529
exponents, 408
expression default values, 408
expressions, 360
EXTENDED, 529
extent, 597

## F

FAILED_LOGIN_ATTEMPTS, 290
failover, 62
false, 360
FALSE AND ..., 228
fastcov, 314
FEDERATED, 385, 408, 432, 554

FEDERATED storage, 98
FEDERATED storage engine, 171
Fedora, 20, 26
FIDO pluggable authentication, 45, 62, 114, 150
fido2, 98
Field_json::cmp_binary(), 472
field_phy_pos, 98
Field_tiny::pack(), 314
file operation error, 114
file space allocation, 334
file writes, 269
FILES, 247, 448, 472
filesort, 51, 114, 228, 247, 269, 385, 432
filtering, 14
fil_shard mutex, 36, 150
fil_space_acquire(), 150, 171
fil_space_acquire_silent(), 150
fil_tablespace_redo_extend(), 171
find_in_group_list(), 150
FIND_IN_SET(), 114, 408
FIPS, 314, 448
firewall, 114, 150, 171, 195, 211, 269, 385, 432
FIREWALL_EXEMPT, 98
firstmatch, 334
FIRST_VALUE(), 448
FLOAT, 71, 98, 131, 195, 314, 334, 408
floating-point, 71, 195, 211
floating-point numbers, 472
FLOOR(), 290, 597
flush, 472
FLUSH DES_KEY_FILE, 507
FLUSH HOSTS statement, 211
FLUSH LOGS, 529
flush method, 597
FLUSH OPTIMIZER_COSTS statement, 211
FLUSH QUERY CACHE, 507
FLUSH STATUS, 360
FLUSH STATUS statement, 211
FLUSH TABLES, 472, 507, 529
FLUSH TABLES FOR EXPORT, 408
FLUSH TABLES statement, 211
FLUSH TABLES WITH READ LOCK, 247
FLUSH TABLES WITH READ LOCK statement, 211
FLUSH USER_RESOURCES statement, 211
FLUSH_OPTIMIZER_COSTS privilege, 211
FLUSH_STATUS privilege, 211
FLUSH_TABLES privilege, 211
FLUSH_USER_RESOURCES privilege, 211
FOLLOWING, 385
FOR SHARE, 228
FORCE INDEX, 507
FORCE_INSOURCE_BUILD, 385
foreign key check, 408
foreign key constraint, 360, 408, 507, 554

foreign keys, 26, 171, 247, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554, 597
foreign_key_checks, 114
FORMAT=TREE, 150, 314
formatting, 529
FORMAT_BYTES(), 290, 360
FORMAT_PICO_TIME(), 290, 360
FOUND_ROWS(), 334
FPROFILE_GENERATE, 290
FPROFILE_USE, 290
fragment page, 597
frame buffer partition offset, 211
FreeBSD, 14, 269, 554
frm files, 334, 408
FROM_DAYS(), 247
FROM_UNIXTIME(), 131, 228, 360, 472
fsync, 408
FTS, 20, 26
FULL, 71
full-text index, 114, 131, 150, 171
full-text search, 131, 171, 195, 211, 228, 290, 334, 385, 432, 448, 472, 507, 529, 554, 597
FULLTEXT, 84, 171
fulltext, 211
fulltext index, 150
FULLTEXT index, 211, 228, 269, 334, 408, 432, 529, 554
func table, 597
functional dependencies, 448
functional indexes, 14, 31, 98, 195, 211, 228, 247, 290, 314, 334, 385, 408
functional key parts, 408
functions, 42, 62, 131, 290, 360, 554
futex, 554

## G

gap lock, 554
gap locks, 448
garbage collection, 31
GCC, 20, 98, 114, 171, 195, 269
gcc, 62
GCS, 26
general tablespace, 507
generated column, 98
generated columns, 62, 131, 150, 171, 247, 290, 334, 360, 385, 408, 432, 448, 472, 529, 554, 597
generated expressions, 228
generated invisible primary keys, 98
GenError, 31
gen_blacklist() function, 211
gen_blocklist() function, 211
gen_dictionary() function, 150
gen_range(), 31, 247
gen_range() function, 150
gen_rnd_pan(), 26

gen_rnd_pan() function, 150
GeomCollection, 448
GeomCollection(), 448
GEOMETRY, 269, 290
geometry, 269, 554
GeometryCollection, 448
GeometryCollection(), 448
GET_LOCK(), 385, 472
get_mysql_time_from_str_no_warn(), 360
GIPK mode, 98
GIS, 71, 98, 150, 171, 195, 211, 228, 247, 269, 314, 334, 360, 385, 408, 432, 448,
472, 507, 529, 554, 597
glibc, 51, 554
GLOBAL_STATUS, 554
global_variables, 472
GLOBAL_VARIABLES, 554
Google Test, 171
googlemock, 31
googletest, 247, 314
GPG key, 42, 131
GRANT, 51, 360, 448, 472
GRANT OPTION, 334
grants, 45
GREATEST(), 84, 290, 314, 334, 448, 472, 597
grep, 9
group, 84, 114, 150, 171, 211, 247, 269, 290, 314, 360, 385, 408, 432, 448, 472, 507,
529, 554
Group, 507
GROUP BY, 26, 36, 71, 171, 211, 247, 269, 290, 334, 360, 408, 432, 448, 472
group commit, 554, 597
group recovery, 36
Group Replication, 9, 14, 20, 26, 31, 31, 36, 42, 45, 51, 62, 71, 84, 98,
114, 131, 150, 171, 195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 385, 408, 432,
448, 472, 507, 529, 554
grouping, 150, 432, 472
GROUPING(), 171, 554
GROUP_CONCAT(), 45, 131, 195, 472, 529, 554
GROUP_CONCAT() function, 171
group_concat_max_len, 131, 472
group_concat_max_len system variable, 171
group_replication_components_stop_timeout, 150
group_replication_exit_state_action, 36
group_replication_paxos_single_leader, 9
group_replication_recovery_get_public_key, 472
group_replication_recovery_public_key_path, 472
group_replication_set_as_primary(), 9, 36
GSSAPI, 71, 269
GTID, 211, 269, 529, 554, 597
GTID sets, 36
GTID skip, 529
GTIDs, 84
gtid_executed, 14, 314, 432, 472
gtid_executed_compression_period, 211
gtid_mode, 507, 529

gtid_next, 385
Gtid_set::Interval_chunk, 26
GTID_SUBSET(), 98

## H

HANDLER, 334
HANDLER READ, 211
handler::position(), 314
Handler::rnd_pos(), 448
Handler_read_key, 314
Handler_read_next, 314
hash functions, 98
hash indexes, 269
hash join, 269, 290, 314
hash joins, 114, 131, 150, 171, 211, 228, 269, 290
hash outer joins, 62
hash scans, 62
hash semijoin, 472
hashes, 211
hashing, 42
hash_join, 290
HASH_SCAN, 14
have_crypt, 507
have_openssl system variable, 171
have_query_cache, 507
have_ssl system variable, 171
have_symlink, 529
HAVING, 42, 51, 62, 131, 171, 228, 334, 472
heap size, 529
heap-use-after-free, 448
Henry Spencer Library, 360
HEX(), 597
hexadecimal, 432
HISTOGRAM, 42, 472
histogram, 472
histogram sampling, 228, 247, 269
histogram smapling, 84
histogram statistics, 171, 290, 507, 529
histograms, 84, 98, 114, 131, 360, 408
histogram_generation_max_mem_size, 385, 529
history list, 472
hole punch, 131
host cache, 211
host name maximum length, 334
host_application_signal component service, 385
host_cache table, 211
host_cache_size, 314
hotfix, 171
hp8, 131

## I

ib::fatal(), 150
ibd2sdi, 432, 448, 507, 529, 597

ibtmp1, 597
ICU, 9, 51, 131, 247, 472
ICU_VERSION(), 472
identifiers, 14, 71
IF, 150, 195, 334
IF EXISTS, 98
IF NOT EXISTS, 114
IF(), 195, 472
IFNULL(), 314, 472
IGNORE, 385
IGNORE UNKNOWN USER, 98
ignore_builtin_innodb, 507
IGNORE_SPACE, 529
implicit commits, 131
implicit tablespace, 472
import, 472, 597
IMPORT TABLE, 554
IMPORT TABLESPACE, 84, 247, 290
Important Change, 9, 14, 26, 36, 42, 45, 51, 62, 71, 84, 98, 114, 131, 150,
228, 247, 269, 290, 385, 408, 432, 448, 472, 554, 597
Important Note, 9, 114, 195
impossible ON, 62, 334
impossible WHERE, 269
IN, 36, 62, 84, 195, 228, 269, 385, 472
IN subqueries, 360
IN subquery, 334
IN(), 150, 171
in-place update, 529
Incident event, 529
includes, 9
Incompatible Change, 20, 45, 84, 114, 131, 150, 171, 195, 247, 269, 314, 360, 408,
432, 448, 472, 507, 529, 554, 597
incomplete predicates, 334
incremental backup, 114
index, 554
index condition pushdown, 195, 472
index dives, 507
index extensions, 171
Index Merge, 51
index merge, 150
index merge scan, 114
index prefixes, 247
index range scans, 26, 131
index scan, 171
index scans, 98, 314
index skip scans, 51
index statistics, 114, 211, 597
index tree, 195
Index.xml, 448
indexes, 20, 290, 408, 432, 448
IndexRangeScanIterator, 131
index_id, 597
INDEX_MERGE, 62, 554
index_subquery, 20

Information Schema, 114, 195, 472
INFORMATION_SCHEMA, 150, 211, 228, 247, 269, 290, 314, 360, 385, 408, 432, 448, 472,
507, 529, 554, 597
INFORMATION_SCHEMA.COLUMNS, 507
INFORMATION_SCHEMA.COLUMN_STATISTICS, 472
INFORMATION_SCHEMA.FILES, 98, 114, 150, 554
INFORMATION_SCHEMA.INNODB_BUFFER_PAGE, 195
INFORMATION_SCHEMA.INNODB_COLUMNS, 314, 432
INFORMATION_SCHEMA.INNODB_SYS_FOREIGN, 507
INFORMATION_SCHEMA.INNODB_TABLESPACES, 211
INFORMATION_SCHEMA.KEYWORDS, 408
INFORMATION_SCHEMA.KEY_COLUMN_USAGE, 114
INFORMATION_SCHEMA.PROCESS_LIST, 131, 150
INFORMATION_SCHEMA.STATISTICS, 448
INFORMATION_SCHEMA.TABLES, 114, 360, 472
init file, 9
initialization, 84, 195, 334
initialize, 408
init_connect, 448
init_file, 269, 314
init_file system variable, 211
inner hash joins, 98
innochecksum, 360, 472, 597
InnoDB, 6, 9, 14, 20, 26, 31, 31, 36, 42, 45, 51, 62, 71, 84, 98,
114, 131, 150, 171, 195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 385, 408, 432,
448, 472, 507, 529, 554, 597
InnoDB memcached plugin, 171, 228
InnoDB system tablespace, 98
innodb-dedicated-server, 171, 385
innodb_buffer_pool_evict, 269
innodb_buffer_pool_instances, 290
Innodb_buffer_pool_resize_status_code, 84
Innodb_buffer_pool_resize_status_progress, 84
INNODB_CACHED_INDEXES, 597
INNODB_COLUMNS, 247
Innodb_data_pending_fsyncs, 62
innodb_ddl_log table, 448
innodb_directories, 171, 334
innodb_doublewrite, 62, 98
innodb_dynamic_metadata table, 448
innodb_extend_and_initialize, 195
innodb_fast_shutdown, 228
innodb_file_format, 597
innodb_file_format_check, 597
innodb_file_format_max, 597
innodb_file_per_table, 408, 472
innodb_flush_log_at_trx_commit, 45, 408
innodb_flush_method, 360, 385
innodb_force_recovery, 290
INNODB_FOREIGN, 360
innodb_fsync_threshold, 408
innodb_idle_flush_pct, 314
innodb_interpreter, 131
innodb_large_prefix, 597

innodb_limit_optimistic_insert_debug, 314
innodb_log_buffer_size, 98
innodb_log_checkpoint_fuzzy_now, 432
innodb_log_files_in_group, 98
innodb_log_file_size, 98
innodb_log_writer_threads, 171, 195
INNODB_METRICS, 360
INNODB_METRICS table, 247, 314, 334
innodb_numa_interleave, 554
innodb_open_files, 131, 150, 195
innodb_redo_log_capacity, 98
innodb_redo_log_consumer_advance(), 84
innodb_redo_log_encrypt, 171
innodb_segment_reserve_factor, 171
innodb_spin_wait_delay, 385
innodb_spin_wait_pause_multiplier, 360
innodb_status_output, 114
innodb_strict_mode system variable, 171
innodb_tmpdir, 334
INNODB_TRX, 211
INNODB_TRX table, 408
innodb_validate_tablespace_paths, 195
INPLACE, 334
INSERT, 45, 71, 131, 228, 334, 432, 472, 554
INSERT ... ON DUPLICATE KEY UPDATE, 269, 360, 448
INSERT IGNORE, 472
INSERT SELECT, 131, 171
INSERT(), 171, 211
INSTALL COMPONENT, 14, 20, 62, 448, 472, 597
INSTALL PLUGIN, 334
installing, 42, 131, 171, 228, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472,
529, 554
INSTALL_STATIC_LIBRARIES, 597
integer, 114
internal SQL parser, 597
internal temporary table, 228, 554
internal_tmp_disk_storage_engine, 360
internal_tmp_mem_storage_engine, 114, 150, 211
INTERSECT, 45, 71, 84
introducers, 247, 472
INT_MAX, 9
inversion, 269
invisible columns, 195, 211
invisible indexes, 195, 228
IPv6, 314
IS NOT NULL, 360
IS NOT TRUE, 334
IS NULL, 228
IS TRUE, 247
isl file, 597
ISNULL(), 334
IS_NULL(), 195
IS_UUID(), 597
Item_subselect::walk_body(), 360

iterator, 290, 314
iterator executor, 269

## J

Japanese, 529, 554
JOIN, 36
join buffer, 131
joins, 26, 62, 84, 114, 150, 195, 211, 334, 360
join_buffer_size, 211
JOIN_CACHE::join_record(), 334
JSON, 31, 36, 45, 51, 71, 131, 150, 171, 195, 211, 228, 247, 269, 290, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554, 597
json, 472
JSON functions, 385
Json_array, 507, 529
JSON_ARRAYAGG(), 71, 195, 211, 385, 408, 554
JSON_ARRAY_INSERT(), 507
JSON_CONTAINS(), 597
JSON_CONTAINS_PATH(), 554
Json_dom, 507
JSON_EXTRACT(), 171, 507, 597
JSON_INSERT(), 472, 507
JSON_LENGTH(), 171
JSON_MERGE(), 507, 597
JSON_MERGE_PATCH(), 507
JSON_MERGE_PRESERVE(), 507
Json_object, 507
JSON_OBJECT(), 228, 472
Json_object::consume(), 597
JSON_OBJECTAGG(), 71, 334, 385, 408, 554
JSON_OVERLAPS(), 334
Json_path, 448
JSON_PRETTY(), 554
JSON_QUOTE(), 448, 554
JSON_REMOVE(), 529
JSON_REPLACE(), 529
JSON_SCHEMA_VALID(), 51, 269, 290, 334
JSON_SCHEMA_VALIDATION_REPORT(), 269, 314, 334
JSON_SEARCH, 554
JSON_SEARCH(), 195, 448, 597
JSON_SET(), 408, 472, 529, 597
JSON_STORAGE_FREE(), 529
JSON_STORAGE_SIZE(), 529
JSON_TABLE, 290
JSON_TABLE(), 131, 150, 247, 269, 290, 334, 385, 408, 432, 448, 472
JSON_TABLE() function, 150
JSON_TYPE(), 597
JSON_UNQUOTE, 554
JSON_UNQUOTE(), 408, 554, 597
JSON_VALUE(), 51, 228, 247
JT_CONST, 472

## K

kana sensitivity, 529
Kerberos, 71, 98, 150, 171, 211, 228, 269
KEY, 247
key sort, 71
keyring, 71, 84, 98, 114, 131, 150, 195, 211, 228, 247, 290, 314, 334, 360, 385, 432, 448, 472
keyring migration, 114, 131, 472
keyring_aws plugin, 98, 290, 314, 334
keyring_aws_region, 314, 334
keyring_component_status Performance Schema table, 84, 195
keyring_encrypted_file plugin, 51, 195, 334, 385, 432, 472
keyring_encrypted_file_password, 290
keyring_file plugin, 51, 195, 448, 554
keyring_hashicorp plugin, 150, 228, 247, 314
keyring_hashicorp_update_config() function, 150
keyring_keys, 360
keyring_oci, 98
keyring_oci plugin, 84, 228
keyring_okv, 9
keyring_okv plugin, 131, 385, 432, 554
keyring_operations, 432
keys, 269
KEYWORDS, 314, 448
keywords, 314, 360
KEYWORDS table, 211
KEY_BLOCK_SIZE, 529
KEY_COLUMN_USAGE, 247, 385
key_hint, 228
KILL, 36, 51, 71, 98, 360, 448
KILL QUERY, 269, 290

## L

LAG(), 472
large object data, 432, 472
last (keyword), 529
last_committed, 529
LAST_INSERT_ID(), 150, 360
Last_query_cost, 360
LAST_VALUE(), 448, 472
latch, 529
latch debugging, 554
later derived tables, 150
LATERAL, 36, 71, 195, 228, 360, 385
lateral derived tables, 385
lateral joins, 334
latin1, 529, 554
latin1_swedish_ci, 554
launchd, 334
lc_messages, 529
lc_messages_dir, 529
LDAP, 211, 228, 247, 269, 290, 314, 360, 385, 408, 448, 472
ldap.conf, 228

LDML, 448
LEAD(), 472
LEAST(), 247, 290, 334, 448, 472, 597
LEFT JOIN, 84, 98, 195, 314, 334, 448
left outer join, 472
LEFT(), 597
libc, 42
libcurl, 9, 26, 211
libedit, 20, 45, 131, 195, 269
libevent, 247, 408, 597
libfido, 26
libmysqlclient, 62
libmysqlclient version, 597
libmysqld, 554, 597
libtirpc, 448
LIBWRAP, 385
licensing, 31
LIKE, 51, 71, 195, 228, 334, 385, 448
LIKE ... ESCAPE, 360
LIKE ESCAPE, 98
LIMIT, 14, 71, 84, 228, 247, 290, 334, 597
limit optimization, 247
LimitNOFILE, 448
LINK_RANDOMIZE, 432
Linux, 14, 171, 448, 472, 529
linux, 334, 360
LIST, 507
literal values, 114, 597
LOAD DATA, 195, 228, 247, 314, 472, 529, 554
LOAD DATA INFILE, 71, 114
LOAD DATA LOCAL, 247, 290
LOAD XML, 247, 529
loadable functions, 131, 171, 195, 211, 290, 334, 360, 448, 507
loading data, 150
LOAD_DATA(), 360
LOAD_FILE() function, 211
LOB, 150, 228, 247, 269, 290, 360, 408, 432, 448, 554
LOB pages, 385
local_infile, 529
LOCATE(), 171, 554
LOCK INSTANCE FOR BACKUP, 448
lock system, 114, 195, 247
LOCK TABLES, 228, 269, 408, 472
lock-free hash table, 98
locking, 51, 84, 98, 114, 131, 150, 171, 195, 211, 228, 247, 269, 290, 314, 334,
360, 385, 408, 448, 472, 507, 529, 554, 597
locking read, 554
locks, 171, 228, 360, 385, 408, 529
LOCK_ORDER, 211, 247
LOCK_ORDER tool, 211, 334
lock_wait_timeout, 408
log, 247
log buffer, 432, 597
log rotation, 554

log system lock, 228
log writer, 211
log writer threads, 228
log-error, 385
logging, 62, 71, 98, 114, 150, 171, 195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554, 597
logical backup, 131
logs, 6
log_bin_trust_function_creators, 51
log_builtin_as_identified_by_password, 448
log_error_services, 98, 247, 507, 529
log_error_suppression_list, 408
log_error_verbosity, 247, 472, 507
log_filter_dragnet, 472
log_filter_internal, 472
log_free_check(), 195
log_raw, 290
log_sink_internal, 472
log_sink_json, 269
log_sink_syseventlog, 408
log_slow_extra, 385
log_statements_unsafe_for_binlog, 51
log_status, 14, 448
log_syslog, 408
log_syslog_facility, 408
log_syslog_include_pid, 408
log_syslog_tag, 408
log_warnings, 507
long messages, 314
long semaphore wait, 98, 114, 131, 290
loose index scan, 36, 150
Loose Index Scan, 472
lowercase_table_names, 385, 408
lower_case_table_names, 228, 290, 334, 385, 448
low_priority_updates system variable, 150
LPAD(), 62, 290, 408
LRU mutex, 171
LWG-3865, 45
lz4, 26, 171
LZ4, 269
lz4_decompress, 597

## M

macce, 131
macOS, 9, 20, 171, 269, 290, 334, 360, 408, 432, 448, 472
macroman, 131
maintainer mode, 36, 290
MAKETIME(), 62
mandatory_roles, 150, 290
mask_ssn(), 26, 114
master key rotation, 314
MASTER_COMPRESSION_ALGORITHMS, 71
MASTER_POS_WAIT(), 14

MATCH, 171
MATCH(), 71, 131
materialization, 211, 314
materialized tables, 84, 472
MAX(), 114, 290, 334, 472
maximum prefix, 71
max_allowed_packet, 507
max_connections, 290, 507
max_delayed_threads system variable, 150
max_error_count, 507
max_error_count system variable, 150
max_execution_time, 98, 408
MAX_EXECUTION_TIME, 290
max_join_size, 9, 71
max_length_for_sort_data, 269, 554
max_prepared_stmt_count, 314
MAX_SIZE_LOG_EVENT_STATUS, 84
max_sort_length, 228
max_tmp_tables, 507
MAX_USER_CONNECTIONS, 269
mbind, 408
MBRContains(), 131, 507, 529
MBRCoveredBy(), 507, 529
MBRCovers(), 507, 529
MBRDisjoint(), 507, 529
MBREquals(), 507, 529
MBRIntersects(), 507, 529
MBROverlaps(), 472, 507, 529
MBRTouches(), 507, 529
MBRWithin(), 507, 529
mbr_utils.cc, 150
MD5() encryption function, 42
MeCab, 314, 334, 432, 472
mecab_charset, 150
MEMBER, 290
MEMBER OF(), 51, 290, 334
memcached, 171, 269, 314, 334, 408, 432, 472, 507, 529, 554, 597
memory, 131, 529, 554
MEMORY, 408, 472
memory allocation, 171, 472
memory allocator, 131
memory management, 131
memory usage, 114, 131
memory_summary_global_by_event_name, 211
mem_heap_alloc(), 114
MEM_ROOT, 51, 314
MERGE, 269, 408, 597
MESSAGE_TEXT, 42
metadata, 385, 529, 597
Metadata locking, 20
metadata locking, 472
metadata logging, 334
metadata_locks table, 171
metadata_locks_cache_size, 408

metadata_locks_hash_instances, 408
Microsoft Windows, 36, 42, 51, 62, 71, 84, 114, 131, 150, 171, 195, 211, 228, 290, 334, 360, 385, 408, 432, 448, 472, 529, 554, 597
MIN(), 114, 334
min_examined_row_limit system variable, 150
monitor thread, 114
MRR, 36, 171
MSVC, 360
MTR_PORT_EXCLUDE, 247
MTR_UNIQUE_IDS_DIR, 360
MTS, 472, 507, 529, 554
multi-source, 529
multi-value indexes, 36
multi-valued index, 98
multi-valued indexes, 26, 114, 131, 150, 195, 211, 228, 334
multifactor authentication (MFA), 45, 71, 150
multithreaded slave, 507, 529, 554
multithreading for replica servers, 150
multivalued indexes, 14
multi_range_count, 507
mutex, 171, 314, 334, 385, 432, 597
MVCC, 228, 385, 408, 507
MyISAM, 31, 84, 432
myisam_repair_threads, 98, 114
myisam_use_mmap, 472
mysql client, 26
MySQL Enterprise Encryption, 62, 84, 98
MySQL NDB ClusterJ, 31, 36
mysql-test-run.pl, 228, 247, 360, 529, 554, 597
mysql-test.run.pl, 554
mysql.gtid_executed, 472, 529
mysql.innodb_table_stats.table_name, 597
mysql.plugin, 334
mysql.session, 432
mysql.session account, 529
mysqladmin, 269, 360
mysqladmin flush-hosts command, 211
mysqlbinlog, 51, 62, 334, 385, 507, 529
mysqlcheck, 84
mysqld, 408
mysqld-auto.cnf, 408, 448
mysqldump, 14, 26, 36, 51, 62, 71, 98, 114, 131, 211, 247, 269, 314, 334, 385, 408, 432, 472, 507, 529, 554, 597
mysqldump command, 195
mysqld_multi, 62, 385
mysqld_safe, 114, 247, 385, 554
mysqld_safe.pid, 554
mysqlimport, 62, 228
mysqlpump, 51, 62, 84, 114, 195, 211, 228, 269, 290, 314, 334, 360, 385, 432, 448, 472, 507, 529, 554
mysqlslap, 269
mysqltest, 247, 472, 529, 554, 597
mysqltest_safe_process, 360
mysqlxtest, 472, 529

mysqlxtext, 432
mysqlx_bind_address, 228
mysqlx_interactive_timeout, 472
mysqlx_read_timeout, 472
mysqlx_wait_timeout, 472
mysqlx_write_timeout, 472
mysql_bind_param() C API function, 62, 211
mysql_command_*, 14
mysql_config, 290, 448, 554
mysql_config_editor, 554
mysql_config_editor command, 195, 211
mysql_current_thread_reader component service, 334
mysql_fetch_row_nonblocking(), 247, 360
mysql_fetch_row_nonblocking() C API function, 247
MYSQL_FIREWALL plugin, 42
mysql_free_result_nonblocking(), 360
MYSQL_GROUP_SUFFIX, 472
MYSQL_HOME, 360
mysql_insert_id(), 554
mysql_install_db, 360, 597
mysql_install_plugin, 408
mysql_list_fields(), 150
mysql_migrate_keyring, 171
mysql_migrate_keyring utility, 195
mysql_native_password, 31, 472
mysql_native_password plugin, 51
mysql_next_result_nonblocking(), 360
mysql_num_rows(), 247
mysql_options(), 247, 529
mysql_options() C API function, 98, 247
MYSQL_OPT_COMPRESS, 71
MYSQL_OPT_COMPRESSION_ALGORITHMS, 71, 247
MYSQL_OPT_LOAD_DATA_LOCAL_DIR, 228, 247
MYSQL_OPT_LOCAL_INFILE, 247
MYSQL_OPT_RECONNECT, 51, 98
MYSQL_OPT_SSL_MODE, 529
MYSQL_OPT_ZSTD_COMPRESSION_LEVEL, 247
mysql_plugin, 597
MYSQL_PWD, 314
mysql_query_attribute_string() function, 211
mysql_real_connect(), 71, 247
mysql_real_connect() C API function, 228
mysql_real_connect_dns_srv() C API function, 228
mysql_real_connect_nonblocking(), 247, 360
mysql_real_connect_nonblocking() C API function, 269
mysql_real_query_nonblocking(), 360
mysql_refresh() function, 211
mysql_reset_server_public_key(), 472
mysql_result_metadata(), 408
mysql_secure_installation, 360, 432, 554
MYSQL_SERVER_PUBLIC_KEY, 448
mysql_service_udf_registration service, 529
mysql_service_udf_registration_aggregate service, 529
mysql_shutdown(), 554, 597

mysql_ssl_rsa_setup, 51, 360, 472
mysql_ssl_set() C API function, 45
mysql_stmt_close(), 529
mysql_stmt_errno(), 529
mysql_stmt_error(), 529
mysql_stmt_sqlstate(), 529
mysql_store_result(), 247
mysql_store_result_nonblocking(), 360
mysql_string component, 529
mysql_udf_metadata component service, 290
mysql_upgrade, 360, 448, 472, 529
mysql_upgrade_info, 334
mysql_use_result(), 247
MYSQL_VERSION file, 228
mysys, 597
my_bool, 554
my_convert_internal(), 26
my_init(), 529
my_print_help(), 20
my_row_id, 195
my_time.cc, 84
my_ulonglong, 314
m_flush_bit, 98

## N

named_pipe_full_access_group, 360, 385
NAME_CONST(), 247
natural joins, 71
NDB Client Programs, 408
NDB Cluster, 9, 51, 62, 269, 290, 314, 334, 360, 408, 597
NDB Cluster APIs, 9
ndb_binlog_index, 597
ndb_cache_check_time, 507
negation, 150, 247
nested joins, 290
network, 26
network namespaces, 211, 228
NETWORK_NAMESPACE, 228
ngram, 432, 507
NO ACTION, 507
NO PAD collations, 334, 385, 529, 554
nondeterministic, 290
NOOP, 472
NOT EXISTS, 334
NOT IN, 171, 334
NOT NULL, 448, 472
NOTIFY_SOCKET, 290, 448
NOW(), 14
NOWAIT, 171, 228, 554
NO_AUTO_CREATE_USER, 448
NO_AUTO_VALUE_ON_ZERO, 314
NO_ENGINE_SUBSTITUTION, 432
NO_HASH_JOIN, 290

NO_INDEX_MERGE, 554
NO_MERGE, 597
NO_SKIP_SCAN, 84
NO_ZERO_DATE, 448
NTH_VALUE(), 472
NTILE(), 62
NULL, 36, 45, 62, 98, 114, 131, 171, 211, 228, 247, 269, 334, 360, 432, 448, 472, 507, 554, 597
null pointer, 98, 114
NULL-safe, 114
null-safe equals, 84
NULLIF(), 171, 195, 408, 529
NUMA, 131, 211, 408, 529, 597
numeric data types, 314

## O

offline_mode, 36, 385
OFFSET, 334, 360
old-style-user-limits, 98
old_passwords, 448, 472
ON DUPLICATE KEY UPDATE, 114, 290, 472, 554
ON EMPTY, 269
ON ERROR, 269
one_or_all, 554
online DDL, 36, 114, 131, 150, 385, 432, 448, 472
only_full_group_by, 448
ONLY_FULL_GROUP_BY, 554
open files limit, 171
opening tables, 42
OpenSSL, 14, 51, 84, 98, 114, 131, 150, 195, 211, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 529, 554
openssl_udf plugin, 98
open_files_limit, 290
operators, 385, 597
optimization, 360
OPTIMIZE PARTITION, 529
OPTIMIZE TABLE, 228, 408
optimizer, 4, 6, 7, 150, 195, 228, 247, 269, 334, 385, 408, 432, 472, 507, 529, 554, 597
optimizer hints, 14, 26, 269, 334, 507, 554, 597
optimizer statistics, 290
optimizer trace, 131, 150, 195
optimizer_search_depth, 195
optimizer_switch, 247, 334, 507
optimizer_trace_max_mem_size, 472
optimizer_trace_offset, 448
options, 71, 84, 98, 114, 131, 150, 171, 228, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554, 597
OR, 51, 150, 290
ORDER BY, 14, 42, 45, 51, 62, 71, 84, 98, 114, 150, 171, 211, 228, 247, 269, 314, 334, 360, 385, 432, 472, 554
ORDER BY DESC, 51
ordered index scans, 62

original_commit_timestamp, 385, 529
os_event, 150
outer joins, 360, 385
outer references, 114, 269
OVER(), 36

## P

packaging, 4, 6, 7, 9, 14, 20, 26, 36, 42, 45, 51, 62, 71, 84, 98,
114, 131, 150, 171, 195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 385, 408, 432,
448, 472, 507, 529, 554, 597
Packaging, 36, 51, 98, 131, 228, 432, 472, 554
PAD SPACE, 290
PAD SPACE collations, 334, 554
padding, 150
PAD_CHAR_TO_FULL_LENGTH, 408
page, 597
page allocation, 131
page cleaner, 597
page cleaner thread, 114, 334, 554
page compression, 114, 150, 211, 228, 448, 507, 529
page flushing, 195, 211, 228, 314, 507
page tracker, 269
page tracking, 114, 290, 360
parallel index read, 334, 385
parallel read, 211, 269, 290
parallel read threads, 228, 247
parallel reader, 4
parallel scan, 150
parallelization, 472, 554
Parallel_reader interface, 228
parser, 4, 98, 228, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529,
554, 597
parser_max_mem_size, 408
partial indexes, 448
partial revokes, 360
partial update, 472, 529
partial updates, 472, 507
partial_revokes, 211
partial_revokes system variable, 247
PARTITION, 42
partition, 507, 597
PARTITION BY, 42
partition handler, 114
partition ID, 507
partition pruning, 114
Partitioning, 9, 14, 26, 62, 84, 114, 131, 171, 195, 211, 247, 290, 334, 360, 385,
408, 432, 448, 472, 507, 529, 554, 597
partitioning, 150, 195, 290, 314, 385, 432, 472, 507, 529, 554
password expiration, 472
password history, 507
password verification, 408
PASSWORD(), 360, 448, 472
passwords, 20, 290, 314, 385, 408, 507

password_history, 507
PASSWORD_LOCK_TIME, 290
password_require_current, 408
password_reuse_interval, 507
PATH, 408
path expressions, 507
paths, 597
PERCENT_RANK(), 62
perdicate elimination, 385
Performance, 14, 26, 42, 45, 171, 247, 269, 507, 529
performance, 432, 597
Performance Schema, 20, 26, 31, 36, 62, 71, 84, 98, 114, 131, 150, 171, 195,
211, 228, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554, 597
PERFORMANCE_SCHEMA.DATA_LOCKS, 360
performance_schema.events_statements_summary_by_digest, 334
performance_schema.procsslist, 114
performance_schema.threads, 114
performance_schema_max_cond_classes, 150
performance_schema_max_mutex_classes, 507
performance_schema_max_thread_classes system variable, 195
performance_schema_max_thread_instances system variable, 195
performance_schema_show_processlist, 228
Performansce Schema, 131
perform_order_index, 31
PERIOD_ADD(), 360, 472
PERIOD_DIFF(), 472
perror, 448
PERSIST, 360
persisted variables, 150, 448
PERSIST_ONLY, 360
persist_only_admin_x509_subject, 385
PERSIST_RO_VARIABLES_ADMIN, 432, 529
PERSIST_RO_VARIABLES_ADMIN privilege, 507
pid_file, 385
PIPES_AS_CONCAT, 314
pkg-config, 554
pluggable authentication, 45, 171, 360, 385, 448, 472, 507, 529
plugin service, 507
plugins, 42, 45, 51, 62, 71, 98, 150, 171, 195, 211, 228, 247, 269, 290, 314,
334, 360, 385, 408, 432, 448, 472, 507, 529, 554
plugin_registry_service, 554
pointers, 51, 84, 507
posix_fallocate(), 211
precision, 150
prefer_ordering_index, 26, 171, 247
prefetch chache, 131
prefixes, 408, 448
preload_buffer_size system variable, 150
PREPARE, 45, 195, 432
prepared statements, 26, 31, 36, 42, 62, 71, 84, 98, 114, 131, 150, 171, 195,
195, 211, 228, 269, 314, 334, 360, 385, 432, 448, 472, 529, 597
primary key, 150, 408
PRIMARY KEY, 171
primary key extensions, 211

primary passwords, 385
print_identified_with_as_hex, 334
privileges, 31, 62, 71, 84, 98, 131, 150, 171, 195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554, 597
proc table, 597
procedure, 597
PROCEDURE ANALYSE(), 554
processlist, 20, 228
PROP_SUBQUERY, 62
protobuf, 9, 26
Protobuf, 247, 290, 314, 334
protocol_compression_algorithms, 247
proxies_priv table, 597
PROXY, 529
pruning, 31, 360, 597
pseudo_slave_mode, 385
pseudo_thread_id, 385
PS_CURRENT_THREAD_ID(), 290, 360
ps_setup_reset_to_default(), 269
PS_THREAD_ID(), 290, 360
punch hole, 171
purge, 98, 114, 131, 150, 171, 211, 247, 290, 360, 385, 597
pushdown, 211, 228, 360

## Q

Qcache_free_blocks, 507
Qcache_free_memory, 507
Qcache_hits, 507
Qcache_inserts, 507
Qcache_lowmem_prunes, 507
Qcache_not_cached, 507
Qcache_queries_in_cache, 507
Qcache_total_blocks, 507
query attributes, 171, 211
query cache, 507, 529, 554
query caching, 45
query cost, 314, 334, 360, 448
query end, 228
query rewrite, 9
query rewrite plugin, 150
query rewrite plugins, 150, 360, 432, 529
query_attributes component, 211
query_cache_limit, 507
query_cache_min_res_unit, 507
query_cache_size, 507, 554
query_cache_type, 507
query_cache_wlock_invalidate, 507
query_digest audit log function, 171
query_prealloc_size, 114
que_eval_sql, 6
QUICK_GROUP_MIN_MAX_SELECT, 472
QUOTE, 45
QUOTE(), 45, 360, 472, 554

quoting, 507

## R

R-tree, 269, 448, 554
race condition, 20, 597
RAND(), 228, 269
random generator functions, 98
random password generation, 314
RANDOM_BYTES(), 51, 62
range, 448
range access, 71, 472
range checks, 114
range expressions, 150
range frames, 360, 385
range index access, 385
range indexes, 131
RANGE INTERVAL, 131
range optimization, 62
range optimizer, 114, 150, 228
range scans, 131
ranges, 529
range_optimizer_max_mem_size, 408
RANK(), 36
RAPID, 334
rapidjson, 269, 554
RapidJSON, 408, 554
raw disk partition, 228
rbr_exec_mode, 385
READ ONLY, 228
read-ahead, 554
READ-COMMITTED, 131
READ_BACKUP, 269
read_only, 228, 290, 529
read_only system variable, 131, 171
REAL, 334
receiver thread, 529
record buffer, 26
records_in_range, 45
recovery, 84, 98, 114, 131, 150, 171, 195, 290, 314, 334, 360, 432, 448, 472, 507, 529, 554, 597
recursion, 150, 314
rec_convert_dtuple_to_rec_old(), 114
rec_get_instant_row_version_old(), 98
rec_get_offsets(), 171
redo, 385, 408
redo log, 26, 98, 131, 228, 247, 314, 360, 385, 408, 448, 472, 529, 554
redo log archiving, 228, 334
redo log format, 269
redo logs, 45, 269
ref access, 385, 472
reference counts, 31
references, 71
regex, 360

regexp, 26, 448
REGEXP, 472
regexp functions, 360, 385
regexp::EvalExprToCharset(), 408
REGEXP_INSTR(), 150, 432, 472
REGEXP_LIKE(), 432, 472
REGEXP_REPLACE(), 211, 334, 432, 472
regexp_stack_limit, 472
REGEXP_SUBSTR(), 247, 334, 432, 472
regexp_time_limit, 472
register, 597
regular accounts, 360
regular expression functions, 211
regular expressions, 448, 472
regular sessions, 360
rejoining, 20
relay log, 31, 62, 529, 554
relay_log_space_limit, 31, 71
RELOAD privilege, 211
rematerialization, 314
removing WHERE, 269
RENAME, 334
RENAME COLUMN, 84, 131, 507
RENAME TABLE, 314, 334, 360, 385, 408, 448, 529, 554
RENAME USER, 228, 432, 448
REPAIR, 334
REPEAT(), 334, 597
REPEATABLE-READ, 529
REPLACE, 432, 448, 472
REPLACE(), 290, 408, 507, 554, 597
replace_index_subquery(), 228
replace_numeric_round, 529
replica reconnection, 20
Replication, 9, 14, 20, 31, 42, 45, 51, 62, 71, 84, 98, 114, 131, 150, 171,
195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 529, 554,
597
replication, 84, 211, 472, 507, 597
replication filtering, 84
replication_applier_status_by_worker, 507, 529
replication_connection_configuration, 228, 472
replication_group_communication_information, 71
replication_group_members, 14
replica_parallel_type, 114, 150
replica_parallel_workers, 98, 150
replica_preserve_commit_order, 150
REPRODUCIBLE_BUILD, 432
require_row_format, 84
REQUIRE_TABLE_PRIMARY_KEY_CHECK, 71
reserved accounts, 228, 432, 472, 529
reserved words, 314, 554, 597
RESET PERSIST, 71, 385, 432, 448, 554
RESET QUERY CACHE, 507
resolveip, 385
resolver, 150, 269

resolve_stack_dump, 385
resolve_type(), 20
resource allocation, 131, 269
resource groups, 71, 84, 314, 385, 507
resource usage, 597
RESTART, 334, 432, 448, 472
restart, 529
restore_last_record(), 334
RESTRICT, 507
RETAIN CURRENT PASSWORD, 385
return values, 150
REVOKE, 98, 472
rewriter, 62
Rewriter plugin, 84
RFC 7159, 507
RIGHT JOIN, 228
RIGHT(), 597
RLIKE, 472
roles, 150, 211, 228, 247, 269, 290, 314, 334, 360, 408, 432, 448, 472, 507, 529, 554, 597
ROLES_GRAPHML(), 448
ROLE_COLUMN_GRANTS, 290
role_edges, 247
ROLE_ROUTINE_GRANTS, 290
ROLE_TABLE_GRANTS, 290
ROLLBACK, 9
rollback, 290, 360, 408, 472, 597
rollback segment, 269
rollback segments, 171, 314, 529, 554
rollbacks, 171
ROLLUP, 20, 26, 42, 51, 62, 114, 131, 150, 171, 211, 228, 247, 334, 360, 385, 408, 448, 472, 554
root, 195
Rotate_log_event, 529
ROUND(), 228
rounding, 150
row comparison, 51
row format, 171, 228, 597
row size, 114
row-based, 554
row-based replication, 448, 472
ROW_COUNT(), 62
ROW_FORMAT, 334, 472, 529, 554
row_table_add_foreign_constraints(), 507
RPAD(), 62, 211, 408
RPC, 385, 448
rpcgen, 448
RPM, 529
Russian, 507
rw_lock_stats counter, 195

## S

SAFE_MUTEX, 507

sandbox mode, 150
SASL LDAP plugin, 98
savepoints, 408
SCAN_NEXTREQ, 9
SCHEMATA_EXTENSIONS, 228
scientific notation, 269
scope, 507
SCRAM-SHA-256, 211
scramble generator, 62
SDI, 195
secondaries, 20
secondary engines, 290
secondary indexes, 131, 171
secondary passwords, 385
secure_auth, 507
secure_file_priv, 554
secure_file_priv system variable, 211
security, 45, 51, 62, 228, 290, 314, 360, 385, 408, 529
SECURITY.md, 26
SELECT, 42, 84, 114, 269, 385, 408, 448, 554
SELECT COUNT(*), 36, 171, 195, 211, 269
SELECT INTO DUMPFILE, 228
SELECT INTO OUTFILE, 228
select_into_buffer_size, 228
select_into_buffer_size system variable, 150
select_into_disk_sync, 228
select_into_disk_sync_delay, 228
select_into_disk_sync_delay system variable, 150
select_lex, 408
SEL_ROOT, 14
semijoin, 228, 334, 472
semijoins, 20, 98, 269, 290, 314, 334, 360
semisync, 9
semisynchronous replication, 211
send buffer, 314
SENSITIVE_VARIABLES_OBSERVER, 98
SERIAL, 131
serialization, 472
serialized dictionary information (SDI), 290, 314, 334, 360, 408, 448, 472, 507, 529
server ID, 71
server initialization, 448
session categories, 360
session temporary tablespace, 195
session tracking, 114
SESSION_STATUS, 554
session_track_gtids, 51
SESSION_VARIABLES, 554
SESSION_VARIABLES_ADMIN privilege, 385
SET, 14, 36, 150, 247, 529
SET DEFAULT ROLE, 472
SET GLOBAL, 529
set operations, 62
SET PASSWORD, 114, 360, 385, 448
SET PERSIST, 269, 290, 314, 385, 448, 472, 507, 529, 597

SET PERSIST_ONLY, 290, 314, 334, 360, 385, 432, 448, 472, 529
SET PERSIST_ONLY statement, 171
SET RESOURCE GROUP, 385
SET ROLE, 360, 432
SET statement, 150, 211, 408
setuid, 432
setup_instruments, 507
setup_threads, 507
set_parse_error_message, 131
SET_USER_ID, 228
SET_VAR, 228, 472, 507
sha256 cache, 472
sha256_password, 360, 472, 507
sha256_password_auth_client_nonblocking(), 71
sha256_password_auto_generate_rsa_keys, 472
shared-memory connections, 211
SHOW, 114, 314
SHOW CHARACTER SET, 247
SHOW COLLATION, 247
SHOW COLUMNS, 211, 247, 269, 314, 507, 529
SHOW CREATE EVENT statement, 171
SHOW CREATE PROCEDURE statement, 171
SHOW CREATE TABLE, 195, 290, 314, 385, 408, 432, 448, 472, 529, 597
SHOW CREATE TABLE statement, 195
SHOW CREATE TRIGGER, 247, 360
SHOW CREATE USER, 334, 432, 507, 597
SHOW CREATE USER statement, 195
SHOW CREATE VIEW, 269, 507
SHOW CREATE VIEW statement, 195
SHOW ENGINE INNODB MUTEX, 228
SHOW ENGINE INNODB STATUS, 195, 597
SHOW ENGINE PERFORMANCE SCHEMA STATUS statement, 195
SHOW EVENTS, 247
SHOW FUNCTION STATUS, 247
SHOW GRANTS, 150, 247, 290, 432, 448
SHOW INDEX, 529
SHOW INDEXES, 408
SHOW KEYS, 247
SHOW PLUGINS, 472
SHOW PROCEDURE STATUS, 247
SHOW PROCESSLIST, 131, 195, 228, 314, 448
SHOW RELAYLOG EVENTS, 529
SHOW REPLICA STATUS, 228
SHOW SCHEMAS, 247
SHOW SLAVE STATUS, 448, 472, 529
SHOW TABLE STATUS, 131, 247, 432, 472
SHOW TABLE STATUS statement, 211
SHOW TABLES, 247, 529
SHOW TRIGGERS, 247
show_create_table_verbosity, 448
SHOW_ENGINE_INNODB_STATUS, 71
show_old_temporals system variable, 150
SHOW_ROUTINE, 269
shutdown, 84, 114, 195, 228, 247, 314, 432, 448, 529

SHUTDOWN, 334, 597
SIGHUP, 269, 290, 385
signal handling, 14
signals, 385, 432
signed variables, 114
signedness, 247
SIGUSR1, 290
single user mode, 269
SKIP LOCKED, 228, 554
Skip Scan, 408
skip-grant-tables, 26
skip-slave-start, 529
skip_name_resolve, 360, 385
skip_networking, 507
slave, 597
slave applier, 529
slave_compressed_protocol, 71
Slave_heartbeat_period, 554
Slave_last_heartbeat, 554
slave_master_info, 228
slave_pending_jobs_size_max, 432, 529
Slave_received_heartbeats, 554
Slave_retried_transactions, 554
slave_rows_search_algorithms, 314
Slave_running, 554
slave_skip_errors, 529
slow query log, 114, 290, 385
slow shutdown, 269, 597
Solaris, 20, 269, 385, 408, 448, 529, 554, 597
sort, 171, 228, 554
sort buffer, 360
sort elimination, 334
sort keys, 408
sorting, 131, 269, 314
sort_buffer_length, 228
sort_buffer_size, 314
source code, 114
source_heartbeat_period, 71
space ID, 247
SPACE(), 62, 150
space_id, 597
SPATIAL, 432
spatial index, 334, 408, 472, 507
spatial indexes, 31, 98, 269, 290
spill to disk, 290
spin lock polling, 360
SQL, 529
SQL syntax, 597
sql_auto_is_null, 360
SQL_BIG_RESULT, 84
sql_buffer_result, 507
SQL_CACHE, 507, 597
SQL_CALC_FOUND_ROWS, 334
sql_executor.cc, 26

sql_help.cc, 150
sql_log_bin, 385, 507
sql_log_off, 385
sql_mode, 385, 408, 432, 448, 529, 554
SQL_NO_CACHE, 472, 507, 597
sql_require_primary_key, 314, 360, 385, 408
sql_safe_updates, 408
SQL_SMALL_RESULT, 84
SRID attribute, 507
SRS, 71
Srv_session, 360
SSL, 14, 26, 36, 42, 45, 51, 62, 71, 84, 98, 114, 131, 150, 171, 195, 211,
247, 269, 314, 334, 360, 385, 408, 432, 448, 472, 529, 554
SSL session, 114
ssl-fips-mode, 150
SSL_connect(), 14
ssl_fips_mode, 408
stack, 114
stack size, 554
stack trace, 20
standard monitor, 448
standards compliance, 247, 314
START GROUP_REPLICATION, 42
START TRANSACTION, 228
start-position, 62
startup, 84, 150, 171, 211, 290, 314, 334, 360, 408, 432, 448, 472, 529, 597
startup configuration, 448
statement event tables, 114
statement summary tables, 114
STATEMENT_DIGEST(), 334, 472
statement_digest(), 334
STATEMENT_DIGEST_TEXT(), 334, 472
statement_digest_text(), 334
statistics, 554, 597
status variables, 290
status_by_thread, 334
status_variable_registration service, 529
std::vector, 529
STOP REPLICA, 71
STOP SLAVE, 71
stopping channel, 20
storage, 171
storage engine attributes, 247
storage engine initailization, 314
storage engines, 597
stored functions, 31, 45, 98, 131, 150, 290
stored objects, 228, 269
stored procedures, 26, 42, 84, 98, 131, 150, 195, 228, 360, 408, 432, 472
stored programs, 42, 45, 71, 131, 150, 171, 195, 211, 269, 290, 334, 385, 432, 448,
472, 507, 529, 597
stored routines, 14, 51, 84, 472
STRAIGHT_JOIN, 269, 597
strict mode, 360
STRICT_TRANS_TABLES, 360

string, 114, 472
string data types, 247
string functions, 84
strings, 171, 360
STR_TO_DATE(), 45, 228, 314
ST_Area(), 408, 472
ST_AsBinary(), 554
ST_AsGeoJSON(), 472, 597
ST_AsText(), 472, 554
ST_AsWKB(), 334, 554
ST_AsWKT(), 554
ST_Buffer(), 472
ST_Buffer() function, 171
ST_Centroid(), 472
ST_Collect() function, 195
ST_Contains(), 269, 507, 529
ST_ConvexHull(), 472
ST_Crosses(), 472, 507, 529
ST_Difference(), 472
ST_Difference() function, 171
ST_Dimension(), 472
ST_Disjoint(), 507, 529
ST_Distance(), 314, 385
ST_Distance_Sphere(), 472
ST_EndPoint(), 472
ST_Envelope(), 472
ST_Equals(), 507, 529
ST_ExteriorRing(), 472
ST_FrechetDistance() function, 211
ST_GeoHash(), 472
ST_GeomCollFromText(), 472, 554
ST_GeomCollFromTXT(), 472
ST_GeomCollFromTxt(), 472, 554
ST_GeomCollFromWKB(), 472, 554
ST_GeometryCollectionFromText(), 472, 554
ST_GeometryCollectionFromWKB(), 472, 554
ST_GeometryFromText(), 472, 554
ST_GeometryFromWKB(), 472, 554
ST_GeometryN(), 472
ST_GeometryType(), 448, 472
ST_GeomFromGeoJSON(), 472, 597
ST_GeomFromText(), 472, 554
ST_GeomFromWKB(), 472, 554
ST_HausdorffDistance() function, 211
ST_InteriorRingN(), 472
ST_Intersection(), 472
ST_Intersection() function, 150
ST_Intersects(), 507, 529
ST_IsClosed(), 472
ST_IsEmpty(), 472
ST_IsSimple(), 472
ST_IsValid(), 448, 472
ST_Latitude(), 432
ST_Length(), 360, 472

ST_LineFromText(), 472, 554
ST_LineFromWKB(), 472, 554
ST_LineInterpolatePoint() function, 195
ST_LineInterpolatePoints() function, 195
ST_LineStringFromText(), 472
ST_LinestringFromText(), 554
ST_LineStringFromWKB(), 472
ST_LinestringFromWKB(), 554
ST_Longitude(), 432
ST_MakeEnvelope(), 472
ST_MLineFromText(), 472, 554
ST_MLineFromWKB(), 472, 554
ST_MPointFromText(), 472, 554
ST_MPointFromWKB(), 472, 554
ST_MPolyFromText(), 472, 554
ST_MPolyFromWKB(), 472, 554
ST_MultiLineStringFromText(), 472
ST_MultiLinestringFromText(), 554
ST_MultiLineStringFromWKB(), 472
ST_MultiLinestringFromWKB(), 554
ST_MultiPointFromText(), 472, 554
ST_MultiPointFromWKB(), 472, 554
ST_MultiPolygonFromText(), 472, 554
ST_MultiPolygonFromWKB(), 472, 554
ST_NumGeometries(), 472
ST_NumInteriorRing(), 472
ST_NumInteriorRings(), 472
ST_NumPoints(), 472
ST_Overlaps(), 507, 529
ST_PointAtDistance() function, 195
ST_PointFromGeoHash(), 472
ST_PointFromText(), 472, 554
ST_PointFromWKB(), 472, 554
ST_PointN(), 472
ST_PolyFromText(), 472, 554
ST_PolyFromWKB(), 472, 554
ST_PolygonFromText(), 472, 554
ST_PolygonFromWKB(), 472, 554
ST_Simplify(), 408, 472
ST_SPATIAL_REFERENCE_SYSTEMS, 472
st_spatial_reference_systems, 472
ST_SRID(), 597
ST_StartPoint(), 472
ST_SwapXY(), 472, 554
ST_SymDifference(), 472
ST_SymDifference() function, 150
ST_Touches(), 507, 529
ST_Transform(), 71, 408
ST_TRANSFORM(), 98
ST_Union(), 472
ST_Union() function, 171
ST_UNITS_OF_MEASURE, 360, 385
ST_Validate(), 408, 472
ST_Within(), 507, 529

ST_X(), 432, 472, 597
ST_Y(), 432, 472, 597
subqueries, 9, 14, 20, 36, 42, 51, 62, 71, 98, 114, 131, 150, 195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 408, 448, 472
subquery, 269
subquery materialization, 290
subquery_to_derived, 9, 51, 62, 150, 211, 228, 247
subselects, 36, 131, 334
SUBSTR(), 131, 597
SUBSTRING_INDEX(), 290, 597
suitably aligned types, 131
SUM(), 334, 408, 554
SUPER privilege, 385, 432, 507, 554
super_read_only, 228, 290, 314
super_read_only system variable, 131, 171
switchover, 20
SyncFileIO::execute, 114
sync_array_detect_deadlock, 131
sync_binlog, 408
sync_frm, 597
sync_relay_log_info, 51
sys schema, 26, 269, 290, 314, 360, 448
sys.diagnostics(), 314
sys.format_bytes(), 290, 360
sys.format_time(), 290, 360
sys.processlist, 314
sys.ps_is_consumer_enabled(), 314
sys.ps_thread_id(), 290, 360
sys.schema_index_statistics, 314
sys.schema_unused_indexes, 314
sys.session, 314
sys.version, 314
syseventlog.facility, 408
syseventlog.include_pid, 408
syseventlog.tag, 408
sysschema.processlist, 114
sysschema.x$processlist, 114
system accounts, 360
system command, 26
system database, 360
system sessions, 360
system tablespace, 211, 334
system variables, 84, 98, 114, 385
systemd, 228, 290, 314, 448
system_time_zone system variable, 171
SYSTEM_USER, 228
SYSTEM_VARIABLES_ADMIN, 432, 529
SYSTEM_VARIABLES_ADMIN privilege, 385, 507
sysvar_audit_log_strategy, 26

## T

table encryption, 360
table functions, 472

table locks, 432
table references, 7, 290
table share, 195
TABLE statement, 290
table statistics, 131, 472, 507
table value constructors, 290
tablespace, 211, 472, 507, 554, 597
TABLESPACE, 472
tablespace encryption, 211, 408, 432, 529, 554
tablespace file, 171
tablespace file size, 290
tablespace import, 269, 290, 314, 448
tablespace path validation, 247
tablespace size limit, 98
TABLESPACES, 228
tablespaces, 472, 529
tables_priv table, 597
table_definition_cache, 472
TABLE_ENCRYPTION_ADMIN, 290, 360
table_encryption_privilege_check, 360
table_io_waits_summary_by_index_usage, 314
TABLE_LIST, 290
table_open_cache, 472
tcmalloc, 31, 247
tcmalloc(), 31
Telemetry traces, 62
telemetry_log plugin, 62
TELEMETRY_LOG_ADMIN, 62
temporal data types, 314, 472
temporal functions, 195
temporary directory, 472
temporary table, 131
temporary tables, 62, 71, 150, 290, 408, 448, 472, 529, 554, 597
temporary tablespace, 408
TempTable, 114, 150
TempTable storage engine, 171, 195, 211, 228, 247, 290, 314, 360, 385, 408, 448, 507
temptable::Allocator, 385
temptable_max_mmap system variable, 211
temptable_max_ram, 211
temptable_use_mmap, 171
Temp_table_param, 269
terminology, 62
test suite, 98, 114, 228, 247, 269, 507, 529, 554, 597
tests, 114
text protocol, 290
thread management, 597
thread pool plugin, 84, 131, 211, 247, 290, 385, 448
thread switching, 360
threads, 529
thread_cache_size, 360
THREAD_ID, 71
thread_mutex, 408
thread_pool, 360
thread_pool plugin, 247, 290, 314, 334

thread_pool_max_active_query_threads system variable, 290
thread_pool_transaction_delay, 84
thread_stack, 150, 432
TI-RPC, 26
TIME, 114, 228, 432
time zone offset, 290
time zone offsets, 131
time zone support, 171
time zones, 114
timeouts, 71
TIMESTAMP, 114, 171, 228, 334, 472, 529
TIMESTAMPADD(), 314
time_format, 507
TIME_TRUNCATE_FRACTIONAL, 554
time_zone, 334
TLS, 62, 114, 171, 247, 334, 360
TLS v1.0, 131
TLS v1.1, 131
TLSv1, 171
TLSv1.1, 171
TLSv1.3, 360
tls_channel_status, 247
tls_version, 131
tmp_table_size, 131
to (keyword), 529
TO_SECONDS(), 360
tp_thread_state table, 71
TP_THREAD_TYPE, 71
trace log, 114
transaction, 507
transaction locks, 507
TransactionDeadlockDetectionTimeout, 9
transactions, 171, 195, 211, 408, 472, 507, 529, 554
transaction_isolation, 507
transaction_prealloc_size, 114
transaction_read_only, 507
transaction_write_set_extraction, 385
transportable tablespace, 432, 529, 597
TREE, 314, 360
triggers, 9, 14, 26, 98, 114, 131, 195, 334, 432, 472, 529, 554
TRIM(), 171, 290
trivial conditions, 385
truncate, 4
TRUNCATE, 150, 432
TRUNCATE PARTITION, 114, 432, 597
TRUNCATE TABLE, 84, 195, 211, 432, 448, 472, 529, 554, 597
TRUNCATE(), 211, 228
truncation, 554
TRX_FORCE_ROLLBACK_ASYNC, 171
trx_sys, 171
trx_t::is_recovered, 171
trx_undo_prev_version_build(), 84
tuning, 507
two-phase commit, 597

tx_isolation, 507
tx_read_only, 507
type comparisons, 247
type conversion, 84, 290, 360, 472, 597
type conversions, 71
type handling, 131
type-checking, 171
types, 62, 71

## U

UBSAN, 195, 211, 247, 314, 334, 408, 448, 554
UBSan, 247, 334, 529
Ubuntu, 26
ucs2, 131
UDF, 51
UDFs, 62, 150
udf_example, 472
uint64_t, 314
ULLONG_MAX, 247
UMASK, 472
UMASK_DIR, 472
Undefined Behavior Sanitizer, 211, 247, 314, 334, 408, 448
underflow, 314
undo, 269, 529
undo log, 211, 360, 385, 408, 472, 554
undo log encryption, 334, 360
undo tablespace, 150, 171, 195, 211, 228, 247, 269, 290, 314, 334, 360, 385, 408, 448,
472, 507, 554, 597
UNHEX(), 247
UNICODE, 131
Unicode, 334, 360, 507, 529, 554, 597
UNINSTALL COMPONENT, 448, 472, 597
UNINSTALL PLUGIN, 334
UNION, 20, 26, 62, 71, 84, 114, 131, 195, 247, 269, 334, 385, 408, 432, 448,
472, 554, 597
UNION ALL, 314
UNION statement, 171
UNIQUE, 114, 334, 448
Unique::io_cache, 408
UNIV_INLINE, 171
UNIV_INNOCHECKSUM, 597
UNIV_MATERIALIZE, 171
UNIX_TIMESTAMP(), 131, 290, 472
UNSIGNED, 314, 334
UPDATE, 14, 114, 131, 228, 247, 269, 334, 385, 432, 472, 554
UPDATE HISTOGRAM, 150
UpdateXML(), 31, 334
upgrade, 51, 71, 84, 98, 131, 150, 195, 211, 228, 247, 269, 290, 314, 334, 360,
408, 432, 448, 472
UPGRADE DATA DIRECTORY NAME, 84
upgrades, 14, 20, 31, 71, 171, 195, 290, 314, 334, 360, 385, 472, 507, 529, 554
upgrading, 290
usability, 554

user table, 597
user variables, 71, 114, 131, 360, 408
user-defined functions, 71
user-defined variables, 195
USER_ATTRIBUTES, 228
user_defined_functions, 211
user_defined_functions table, 529
user_variables, 472
use_invisible_indexes, 228, 507
USE_LD_LLD, 360
Using index, 290
ut::aligned, 171
UTC, 228
UTF, 195
UTF-16, 62, 554
utf32, 114, 385, 448
UTF32, 314
UTF8, 14
utf8, 408, 448, 472
utf8 character set, 171, 195
utf8mb3, 408, 448
utf8mb3 character set, 171, 195
utf8mb4, 290, 334, 360, 408, 448, 507, 529, 554
utf8mb4_0900_ai_ci, 554, 597
utf8mb4_0900_as_ci, 529
utf8mb4_0900_bin, 334
utf8mb4_general_ci, 554
utf8mb4_ja_0900_as_cs_ks, 529
utf8mb4_ru_0900_ai_ci, 507
utf8mb4_ru_0900_as_cs, 507
utf8mb4_zh_0900_as_cs, 360
ut_allocator, 171
ut_allocator(), 171
UT_DELETE, 171
ut_list, 171
UT_LOCATION_HERE, 98
ut_rnd_ulint_counter, 597
ut_time(), 131
UUID(), 507
UUIDs, 597
UUID_TO_BIN(), 597

## V

Valgrind, 84, 114, 269, 507, 597
valgrind, 269
validate_password, 385, 432, 448, 472
validate_password component, 51
validate_password plugin, 597
VALUES, 36, 114, 150, 211
VALUES statement, 114, 290
VALUES(), 269
VARCHAR, 114, 228
variable conversions, 131

variables, 385
variables_by_thread, 385
variables_info, 385, 448, 472, 507
VARIANCE(), 448
Variance-Aware Transaction Scheduling (VATS), 472, 507
VAR_POP(), 448
VAR_SAMP(), 448
version comments, 51
VERSION file, 228
version table, 529
version_compile_zlib, 448
version_tokens plugin, 554
version_tokens_session, 334
view references, 360
views, 9, 14, 20, 42, 51, 62, 71, 84, 98, 114, 150, 171, 211, 228, 290, 385,
472, 554, 597
View_change_log_event, 71
VIEW_ROUTINE_USAGE, 408
VIEW_TABLE_USAGE, 408
virtual column, 195
virtual columns, 150, 171, 195, 269, 314, 334, 360, 385, 432, 448, 472, 507, 529, 554,
597
virtual indexes, 360, 448, 554, 597
virtual memory, 84
Visual C++ Redistributable, 554
Visual Studio, 42, 195

## W

WAIT_FOR_EXECUTED_GTID_SET(), 62
wait_timeout system variable, 195
WAIT_UNTIL_SQL_THREAD_AFTER_GTIDS(), 9
walk and replace, 171
WalkAndReplace(), 150
warnings, 408
WebAuthn pluggable authentication, 45
weedout, 269, 290, 314
WEIGHT_STRING(), 247, 472, 554
WHERE, 290, 314, 334
WHERE_CONDITION optimization, 211
whitespace, 529
widen_fraction, 360
wildcards, 45, 334, 507
window function, 62
window functions, 42, 45, 51, 62, 71, 114, 150, 171, 195, 211, 228, 314, 360, 385,
408, 432, 448, 472, 507, 529
windowing functions, 9, 42, 71, 131, 334, 360, 385, 472
windowing_use_high_precision, 448
windows, 26, 42, 45, 98
Windows, 334, 385
Windows 32-bit, 98
WITH, 554
WITH ADMIN, 334
WITH ADMIN OPTION, 247

WITH CHECK OPTION, 114
WITH RECURSIVE, 290
WITH ROLLUP, 211, 247, 432
WITHOUT_SERVER, 26
WITH_AUTHENTICATION_CLIENT_PLUGINS CMake option, 171
WITH_DEFAULT_FEATURE_SET, 228
WITH_EDITLINE, 448
WITH_GMOCK, 408
WITH_GMOCK CMake option, 171
WITH_ICU, 211, 448, 472
WITH_INNODB_EXTRA_DEBUG, 448
WITH_JEMALLOC, 360
WITH_LIBEVENT, 247, 448
WITH_LOCK_ORDER, 334
WITH_LSAN, 360
WITH_LTO, 269, 290, 408
WITH_LZ4, 448
WITH_LZMA, 360, 472
WITH_MYSQLX, 360
WITH_PACKAGE_FLAGS CMake option, 171
WITH_PROTOBUF, 448
WITH_RAPIDJSON, 408
WITH_RE2, 314, 472
WITH_ROUTER, 360
WITH_SSL, 290, 448, 472
WITH_SYSTEMD_DEBUG, 228
WITH_SYSTEM_LIBS, 448
WITH_TCMALLOC, 228
WITH_UNIT_TESTS CMake option, 171
WITH_VALGRIND, 98
WITH_WIN_JEMALLOC, 71
WITH_ZLIB, 448
WITH_ZSTD, 51, 228
wolfSSL, 314, 448
WRITE_CONSENSUS_SINGLE_LEADER_CAPABLE, 71
WSAPoll, 84

## X

X DevAPI, 334, 385
X Plugin, 385, 448
X Protocol, 4, 114, 171, 211, 228, 247, 269, 290, 314, 334, 360, 385, 408, 432, 448, 472, 507, 554, 597
X509, 472
XA, 62, 171, 195, 228, 290, 314, 408, 448, 472, 529, 554, 597
XA RECOVER, 529
XA statement, 314
XA transaction, 211
XA transactions, 84, 290, 408, 472, 529, 554, 597
XA Transactions, 211
XA_RECOVER_ADMIN, 529
XCode, 20
xcodebuild, 20
XDR, 385

XML, 554
XOR, 42
XPath syntax, 529
xxHash, 14
xxhash, 14

## Y

yassl, 408, 432
yaSSL, 448, 472, 507, 554
yaSSL support, 51
YEAR, 84, 150, 195, 228, 290, 334, 385, 408
YEAR(), 385
Yum repository, 472
YYYYMMDD, 448

## Z

zero-length columns, 36
ZEROFILL, 290, 314, 334
zlib, 4, 26, 71, 448, 507
zlib_decompress, 597
zstd, 9, 290