

---

# MySQL 9.6 Release Notes

## Abstract

This document contains release notes for the changes in MySQL 9.6. For information about changes in a different version of MySQL, see the release notes for that version.

For additional MySQL 9.6 documentation, see the [MySQL 9.6 Reference Manual](#), which includes an overview of features added in MySQL 9.6 ([What Is New in MySQL 9.6](#)), and discussion of upgrade issues that you may encounter while [upgrading](#).

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: 2026-02-27 (revision: 30971)

## Table of Contents

Preface and Legal Notices .....	1
Changes in MySQL 9.6.0 (2026-01-20, Innovation Release) .....	3
Changes in MySQL 9.5.0 (2025-10-21, Innovation Release) .....	10
Changes in MySQL 9.4.0 (2025-07-22, Innovation Release) .....	20
Changes in MySQL 9.3.0 (2025-04-15, Innovation Release) .....	33
Changes in MySQL 9.2.0 (2025-01-21, Innovation Release) .....	48
Changes in MySQL 9.1.0 (2024-10-15, Innovation Release) .....	62
Changes in MySQL 9.0.1 (2024-07-23, Innovation Release) .....	80
Changes in MySQL 9.0.0 (2024-07-01, Innovation Release) .....	80

## Preface and Legal Notices

This document contains release notes for the changes in MySQL 9.6.

### 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 9.6.0 (2026-01-20, Innovation Release)



### Note

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

- [Audit Log Notes](#)
- [Compilation Notes](#)
- [Configuration Notes](#)
- [Doxygen Notes](#)
- [Replication with GTIDs](#)
- [InnoDB Notes](#)
- [JavaScript Programs](#)
- [JSON Duality Views](#)
- [Logging Notes](#)
- [MySQL Enterprise Notes](#)
- [Optimizer Notes](#)

- [Packaging Notes](#)
- [Performance Schema Notes](#)
- [Pluggable Authentication](#)
- [Security Notes](#)
- [SQL Syntax Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

## Audit Log Notes

- Updated the `audit_log_rotate_on_size` system variable to require the `AUDIT_ADMIN` privilege for modification. (Bug #38620615)
- MySQL now supports a more modular and efficient auditing system with the introduction of the Audit Log component. This update splits the monolithic Audit Log software into smaller, more manageable components, encapsulated into MySQL components, making it easier to install, manage, and maintain. The new component-based architecture allows for improved flexibility and customization, enabling users to easily configure audit log settings, such as output file location, format, and buffer size. With this enhancement, MySQL provides a more streamlined and user-friendly auditing experience, allowing you to focus on monitoring and analyzing your database activities with greater ease.

See [MySQL Enterprise Audit Component](#) for more information. (WL #12716)

## Compilation Notes

- MySQL Server's `clang-tidy` configuration now consistently disables all brace checks, including `hicpp-braces-around-statements`, to prevent duplicate diagnostics. (Bug #38601902)
- A number of unused symbols were removed from the code base. (Bug #38557463)
- The cmake macro `MYSQL_CHECK_PROTOBUF` did not properly handle multiple calls when using system protobuf. Errors were returned similar to the following:

```
ADD_LIBRARY cannot create imported target
"ext::libprotobuf" because another target with the same
name already exists.
```

(Bug #38417986)

## Configuration Notes

- The `mysqlx_port` server variable was not persisted in the `my.ini` file when set through the **Type and Networking** configuration page. (Bug #38728671)
- MySQL Configurator treated single-quotes (') as part of the password.  
The configurator now correctly handles quoted passwords by removing any quotes, double quotes, or backticks from the beginning and end of the password. (Bug #38069029)
- Server file permissions were not applied as specified with `--server-file-permissions-access`. Full access was granted regardless of the value defined. (Bug #38068994)

- `--slow-query-log-file` could be used even if the slow query log was not enabled. (Bug #38068822)
- It was possible to set the `--general-log-file` CLI option when the general log was disabled. As of this release, an error is returned. (Bug #38068775)
- `--open-win-firewall` could be used when TCP/IP connections were disabled. As of this release, `--open-win-firewall` can only be used when TCP/IP connections are enabled. (Bug #38068699)
- Updating the password with MySQL Configurator took a long time. (Bug #37459271)

## Doxygen Notes

- A number of issues were corrected in the Server API documentation. Our thanks to Daniël van Eeden for the contribution. (Bug #114671, Bug #117516, Bug #37329988, Bug #37615382, Bug #36526371)

## Replication with GTIDs

- MySQL Server now supports a new GTID set data structure, providing a simpler, more modern, and efficient library for handling Global Transaction IDs. This implementation replaces the existing library, offering improved maintainability and performance. (WL #16076)

## 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)

- **InnoDB:** If an `XA PREPARED` transaction existed during a server shutdown or crash, restarting the server with a non-zero `--innodb-force-recovery` value caused InnoDB to incorrectly change the transaction state from `PREPARED` to `ACTIVE`. This could lead to issues ranging from assertion failures to rolling back the `PREPARED` transaction. (Bug #35659774)
- **InnoDB:** InnoDB now generates unique rowid values in tables without primary keys more efficiently. (Bug #13395168)
- **InnoDB:** The field `fk_max_recursive_level` in `dict_table_t struct` is no longer used and has been removed.

Our thanks to Xiacong Ding for the contribution. (Bug #119063, Bug #38492574)

- **InnoDB:** If the server closed during transaction commit, transactions in an already committed state could remain in the undo log. If the transaction is already committed, it is cleared from `trx_sys->rw_trx_list` during recovery, but is not cleared from `active_tw_trxs` of the `trx_sys->shards`. This invalid transaction stored in the `trx_sys->shards` could cause errors or assertions in the execution of the later queries.

As of this release, during cleanup on startup, processed transactions are not added to `trx_sys->shards`.

Our thanks to Yewei Xu and the team at Tencent for the contribution. (Bug #118515, Bug #38132549)

- **InnoDB:** When selecting blob data after a small partial update, incorrect results may be returned due to inconsistencies in the Multi-Version Concurrency Control (MVCC) for blob objects. (Bug #114138, Bug #36342348)
- **InnoDB:** Concurrent execution of FLUSH TABLE FOR EXPORT with DROP TABLE and DML operations could result in a deadlock. (Bug #110485, Bug #35221441)
- Fixed an issue related to bulk inserts. (Bug #38208188)
- MySQL Server now supports a new startup option, `container_aware`, which controls the server's ability to discover and adhere to CPU and memory resource limits set by a container. (WL #16937)

## JavaScript Programs

- GraalVM performance metrics are now available on Windows. (Bug #38498718)
- The `SHOW CREATE LIBRARY` command generated HEX representations of binary libraries on x86\_64 architectures with `FFFFFF` patterns before each byte with the high bit set.

Our thanks to Akshat Nehra and the team at Amazon for the contribution. (Bug #118927, Bug #38458235)

## JSON Duality Views

- The table IDs reported in the I\_S views for JSON Duality Views differed from those reported by ORACLE DB due to differences in calculation order, leading to inconsistent `REFERENCED_TABLE_ID` values when multiple `Content_tree_nodes` referred to the same base table. (Bug #38624596)

## Logging Notes

- MySQL now supports enhanced debugging capabilities with the addition of timestamps to GCS/XCOM trace file entries.

For more information, see the `group_replication_communication_debug_options` system variable documentation in [Group Replication System Variables](#). (WL #17008)

## MySQL Enterprise Notes

- Option Tracker now supports MySQL Shell and MySQL Shell for VS Code.

The following status variables were added:

- `option_tracker_usage-MySQL_Shell_VSC_Natural_Language_to_SQL`
- `option_tracker_usage-MySQL_Shell_VSC_HeatWave_Chat`
- `option_tracker_usage-MySQL_Shell_VSC_Natural_Language_to_SQL`
- `option_tracker_usage-MySQL_Shell_VSC_Lakehouse_Navigator`
- `option_tracker_usage-MySQL_Shell`
- `option_tracker_usage-MySQL_Shell_for_VS_Code`

- `option_tracker_usage-MySQL_Shell_Dump`
- `option_tracker_usage-MySQL_Shell_VSC_Dump`
- `option_tracker_usage-MySQL_Shell_Dump_Load`
- `option_tracker_usage-MySQL_Shell_VSC_Dump_Load`
- `option_tracker_usage-MySQL_Shell_MRS`
- `option_tracker_usage-MySQL_Shell_VSC_MRS`
- `option_tracker_usage-MySQL_Shell_Copy`
- `option_tracker_usage-MySQL_Shell_Upgrade_Checker`

The following system variable was added:

- `component_option_tracker.mysql_shell_support`

See [Option Tracker Component](#). (WL #16659, WL #17000)

## Optimizer Notes

- Fixed an issue relating to Common Table Expressions (CTE). (Bug #38573285)
- Using the `coalesce` or `any_value` functions with specific JSON operations, such as `json_search` and `json_remove`, could cause an assertion failure. (Bug #38549573)
- Negative year values were accepted as valid datetime values by one method, but not by another. Negative year values are now always reported as errors. (Bug #38487373)
- Fixed issues relating to executing specific SELECT queries. (Bug #38465147, Bug #38465178)
- 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)

## Performance Schema Notes

- **Important Change:** It is now possible to view accounts which have been temporarily locked, using the new Performance Schema table `TEMPORARY_ACCOUNT_LOCKS`.

The Performance Schema table, `HOST_CACHE`, was extended with the following columns:

- `COUNT_ACCOUNT_LOCKED_ERRORS`: The number of errors caused by a permanently locked account.

- [COUNT\\_TEMPORARY\\_ACCOUNT\\_LOCKED\\_ERRORS](#): The number of errors caused by a temporarily locked account.

(Bug #37378977, WL #16803)

- Fixed an issue relating to querying variables. (Bug #38100668)
- Performance Schema did not capture SQL or digests for prepared statements from non-MySQL clients. (Bug #114480, Bug #36445818)

- Negative numbers on digests in [WHERE](#) were not parsed properly.

Our thanks to Yakir Gibraltar for the contribution. (Bug #110045, Bug #35078042)

- The following logs are now instrumented for Telemetry Logging:

- Slow Query Log
- General Log

See [The setup\\_loggers Table](#) and [Configuring Log Telemetry](#). (WL #17167)

## Pluggable Authentication

- Connecting to a MySQL server with a non-existing user now consistently returns an "Access denied for user" error message, regardless of the username length or MySQL version. Previously, errors were returned similar to the following:

```
ERROR 1045 (28000): Access denied for user
'foo'@'localhost' (using password: NO), ERROR
1524 (HY000): Plugin 'mysql_native_password' is not
loaded
```

(Bug #36527984)

## Security Notes

- **Important Change:** MySQL now supports enhanced security and flexibility with the relocation of [MD5\(\)](#) and [SHA1\(\)](#) SQL functions to a separate component, allowing for greater control over deprecated hashing algorithms and improved compliance with security standards. You can install the [classic\\_hashing](#) component to continue using [MD5\(\)](#) and [SHA1\(\)](#) functions in your applications, ensuring a seamless transition and maintaining the integrity of your database operations. This update enables you to choose whether to utilize these functions, promoting a more secure environment by avoiding unacceptable hashing algorithms as defined by industry standards.

See [Legacy Hashing Component](#) for more information. (WL #16956)

- Fixed an issue relating to authentication. (Bug #118447, Bug #38077617)
- The error message returned when attempting to create a user with an existing user name was unclear. (Bug #28331, Bug #11746738)

## SQL Syntax Notes

- MySQL Server now supports table-level DML tagging for JSON duality views. You can explicitly specify allowed DML operations ([INSERT](#), [UPDATE](#), [DELETE](#)) per table when defining a JSON duality view. Restrictive tags ([NO INSERT](#), [NO UPDATE](#), [NO DELETE](#)) are also supported to explicitly disallow

specific operations. You can specify combinations of these tags. At runtime, JSON duality views validate incoming DML operations against the configured tags and accept or reject operations accordingly. This provides per-table control over write operations within a single duality view definition. The behavior is compatible with Oracle DB JSON duality views.

See [CREATE JSON DUALITY VIEW Statement](#) for more information. (WL #17053)

## Functionality Added or Changed

- **Important Change:** In previous releases, Foreign Key constraints and cascade operations were handled by the InnoDB storage engine. As of this release, they are handled by the SQL layer of the MySQL Server. This change allows for more comprehensive binary logs and improved analytics. This change is fully backward compatible and ensures that data changes are always visible, logged, and replicated.

To carry on using InnoDB to handle Foreign Key constraints and cascade operations, you must start the server with the new system variable, `innodb_native_foreign_keys`. (WL #11249)

- The bundled version of `opentelemetry-cpp` was upgraded to version 1.23.0. (Bug #38483354)
- As of this release, `GROUPING()` is permitted in queries without `ROLLUP`. (Bug #38168051)

## Bugs Fixed

- **InnoDB:** Under certain circumstances, when using the `que_eval_sql` interface, a race condition could occur. (Bug #118705, Bug #38310595)
- `mysqltest` did not print the symbolic error code (error name) for failed SQL statements. (Bug #38743161)
- Running `SET PERSIST` on a system variable after an upgrade can result in duplicate variable entries across different sections in `mysqld-auto.cnf`. (Bug #38680162)
- Binary log purged before persisted `binlog_expire_*` options were loaded. (Bug #38554467)
- Fixed an issue relating to changing users and roles. (Bug #38492547)
- Linux debug builds started with the `--basedir` option set to an existing absolute path will no longer assert that `local_mysql_home` ends with a specific character. (Bug #38483568)
- Releasing MDLs before the `Auto_releaser` has run could cause asserts in debug builds due to locks not being released. (Bug #38430539)
- 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 an issue related to user name handling. (Bug #38355483)
- `mysqldump`'s `--order-by-primary` option sorted data by every index on the table, instead of just sorting by the primary key. (Bug #38284832)
- Fixed several issues relating to Thread Pool blocking connections from closing. (Bug #38170188, Bug #36782728, Bug #38549372)
- Under certain circumstances, when used for an upgrade, `mysqlpump` altered the SQL mode and collation of stored procedures. (Bug #38154661)
- Fixed several issues in the parser. (Bug #37866044, Bug #38483735)

- Upgraded the MySQL Server to use non-deprecated OpenSSL algorithms for enhanced security and compatibility. (Bug #33951773)
- It was possible to specify a password expiry policy for externally authenticated users. (Bug #32724080)
- GTID gaps were generated because of the `replica- skip-errors` option. (Bug #28590993)
- Database names containing a backslash were not correctly quoted when using the `--routines` option with `mysqldump`. (Bug #22240504)
- 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)
- (Bug #97091, Bug #30381098)
- `Wait_timeout` and the sql\_mode `MODE_IGNORE_SPACE` were set at connect time, only. They should also have been set at `reset_connection` time.  
Our thanks to Facebook for the contribution. (Bug #97091, Bug #30381098, Bug #97666, Bug #30551153)
- A memory management issue occurred in the CSV engine when meta files became inaccessible.  
Our thanks to Yan Huang for the contribution. (Bug #93394, Bug #28987279)
- Trailing comments on the `!includedir` or `!include` directives in the MySQL configuration file could prevent the server from starting. (Bug #87657, Bug #26750972)

## Changes in MySQL 9.5.0 (2025-10-21, Innovation Release)



### Note

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

- [Account Management Notes](#)
- [Audit Log Notes](#)
- [Authentication Notes](#)
- [Binary Logging](#)
- [Compilation Notes](#)
- [Component Notes](#)
- [Configuration Notes](#)
- [Connection Management Notes](#)
- [Deprecation and Removal Notes](#)
- [SQL Function and Operator Notes](#)
- [InnoDB Notes](#)
- [JavaScript Programs](#)

- [JSON Notes](#)
- [JSON Duality Views](#)
- [Keyring Notes](#)
- [MySQL Programs](#)
- [Optimizer Notes](#)
- [Packaging Notes](#)
- [Performance Schema Notes](#)
- [Pluggable Authentication](#)
- [Replication Notes](#)
- [Spatial Data Support](#)
- [sys Schema Notes](#)
- [Thread Pool Notes](#)
- [Bugs Fixed](#)

## Account Management Notes

- MySQL now supports enhanced role management with the introduction of a new option, `activate_mandatory_roles`.

If `activate_all_roles_on_login` is enabled, `activate_mandatory_roles` is ignored, and both mandatory and granted roles are activated. If `activate_all_roles_on_login` is disabled and `activate_mandatory_roles` is enabled, mandatory roles are activated in addition to any default roles associated with the account.

If neither are activated, only the account's default roles are activated.

`activate_mandatory_roles` is enabled by default. (WL #16597)

## Audit Log Notes

- Fixed an issue related to audit log strategy. (Bug #38223011)
- 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)

## Authentication Notes

- **Security Enhancement:** The default value of `caching_sha2_password_digest_rounds` is increased to 10,000. (WL #16974)
- Under certain circumstances, the keyring component could not read the configuration file if the configured data directory path was shorter than the default path. (Bug #38078884)
- General-Purpose Keyring Functions did not handle invalid parameters correctly. (Bug #38055488)

## Binary Logging

- **Important Change:** The default value for the `binlog_transaction_dependency_history_size` server system variable has been increased from 25000 to 1000000 (one million). This variable's maximum value has also been increased, from 1000000 to 10000000 (ten million).

This change should not have any immediate effect on existing setups.

See the description of `binlog_transaction_dependency_history_size` for further information. (WL #16439)

## Compilation Notes

- **Solaris:** Clang and GCC now can be used for compiling MySQL on Solaris. (Bug #30562248)
- 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)
- MySQL Server now bundles libzip and provides CMAKE options to choose between the bundled or system library when compiling the server. (Bug #38109824)
- The minimum required version of CMAKE is now 3.17.5, for supported platforms. (Bug #38090672)
- Compiler warnings were logged when building MySQL Server with GCC 15. (Bug #37721464)

## Component Notes

- The `MYSQL_OPTION.OPTION_USAGE.OPTION_NAME` field used a different character set than `PERFORMANCE_SCHEMA.MYSQL_OPTION.OPTION_NAME`.

As of this release, the `CLUSTER_ID`, `SERVER_ID`, and `OPTION_NAME` fields of `MYSQL_OPTION.OPTION_USAGE` use `utf8mb4_general_ci` instead of `ascii`. (Bug #38230424)

- The `sql_errno()` method in the `mysql_command_error_info` service returned 0 when a command failed, instead of the appropriate error code. (Bug #38178087)
- The server plugin system variable flag `PLUGIN_VAR_NOSYSVAR` was not processed properly. Variables defined with this flag were displayed when they should be hidden. (Bug #37654916)
- Errors were not handled correctly when using the keyring UDF plugin and executing specific queries with the `keyring_key_generate` function. (Bug #37501458)
- Telemetry component introduces a secret decoding service. This enables you to create your own secret-decoding service for OpenTelemetry.

The following system and status variables were added:

- `telemetry.otel_exporter_otlp_traces_secret_headers`
- `telemetry.otel_exporter_otlp_metrics_secret_headers`
- `telemetry.otel_exporter_otlp_logs_secret_headers`
- `telemetry.resource_provider`

- `telemetry.secret_provider`
- `telemetry.run_level`

(WL #16647)

## Configuration Notes

- **Microsoft Windows:** If the MySQL Server was configured as a process, not as a service, the Configurator's **Starting the server** step could take a long time to complete due to delays in detecting when the server was ready to receive connections. (Bug #38068649)
- **Microsoft Windows:** Shared Memory connections, `--enable-shared-memory=true`, failed when they were the only enabled protocol during server configuration. (Bug #38068619)
- **Microsoft Windows:** It was not possible to create usernames with escaped characters, using single or double quotes, during new server configuration. (Bug #37459624)
- **Microsoft Windows:** When upgrading, for example from version 9.1.0 to 9.2.0, using a non-default port, such as 3307, messages could be generated indicating a connection issue to port 3306, even if the server configuration is correct and the upgrade is successful. Errors were returned similar to the following:

```
Can't connect to MySQL server on 'localhost:3306'  
(10061)
```

(Bug #37459255)

- **Microsoft Windows:** The CLI help for MySQL Configurator now correctly displays the list of supported values for configurable server options. (Bug #37458648)
- `--config-type` Configurator CLI option did not accept any values. (Bug #38068684)

## Connection Management Notes

- Connection control now supports exempting unknown users from delay criteria, introducing a new system variable `component_connection_control.exempt_unknown_users` that allows administrators to configure whether failed connection attempts without user credentials should be penalized. This enhancement improves the component's ability to handle legitimate connection attempts from load balancers, ensuring better server availability while maintaining effectiveness in thwarting brute force attacks. A new status variable `component_connection_control.exempted_unknown_users` provides insights into the number of exempted connections.

For more information, see (xref linkend="to-fill"). (WL #17037)

## Deprecation and Removal Notes

- The `group_replication_allow_local_lower_version_join` system variable, deprecated in MySQL 8.4.0, has now been removed. (WL #16019)
- The `replica_parallel_type` system variable, deprecated in MySQL 8.0.29, has now been removed. (WL #13955)
- The `SCRAM-SHA-1` authentication method for SASL LDAP authentication is now deprecated, and subject to removal in a future version of MySQL. You are encouraged to use `SCRAM-SHA-256` instead.

As part of this work, the default value for the `authentication_ldap_sasl_auth_method_name` system variable has been changed to `SCRAM-SHA-256`; the value `SCRAM-SHA-1` is now deprecated.

For further information, see [LDAP Pluggable Authentication](#). (WL #17022, WL #17034)

- The following deprecated plugins are removed in this release:
  - `semisync_master` replaced by `semisync_source`.
  - `semisync_slave` replaced by `semisync_replica`.

(WL #16801)

## 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

- **Important Change:** The default value for `innodb_log_writer_threads` is now determined in part by whether binary logging is enabled, as shown here:

```
if (log_bin = OFF)
{
  if ([number of logical CPUs] <= 4)
  {
    innodb_log_writer_threads = OFF
  }
  else
  {
    innodb_log_writer_threads = ON
  }
}
else
{
  if ([number of logical CPUs] <= 32)
  {
    innodb_log_writer_threads = OFF
  }
  else
  {
    innodb_log_writer_threads = ON
  }
}
```

When binary logging is disabled (`log_bin` is `OFF`), the default is determined as it was in MySQL 9.4 and earlier.

This change does not affect the variable's configured value, if set.

For further information, see the description of `innodb_log_writer_threads` in the MySQL documentation; see also [Optimizing InnoDB Redo Logging](#). (WL #16999)

- **InnoDB:** The `dict_stats_analyze_index()` function could discard optimal index-based access paths, resulting in suboptimal performance for large tables when empty index statistics are read concurrently or an error leads to empty index statistics.

Our thanks to Casa Zang and the Tencent team for the contribution. (Bug #33472935)

- 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)
- The `mlog read_1_bytes` comment now accurately reflects a byte size of 1 byte, rather than 2 bytes.

Our thanks to Jeong Geun Lee for the contribution. (Bug #38001828)

- 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 the purge coordinator.

Our thanks to Yin Peng and the Tencent team for the contribution. (Bug #37628911)

- Fixed an issue related to virtual indexes. (Bug #37602657)
- MySQL Server could behave unpredictably when simultaneously optimizing and updating a table. (Bug #37324137)
- 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)

- The `dict_stats_analyze_index()` function in could discard optimal index-based access paths, resulting in suboptimal performance for large tables when empty index statistics are read concurrently or an error return leads to empty index statistics.

Our thanks to Casa Zang and the Tencent team for the contribution. (Bug #33472935)

- Under certain, rare circumstances, a race condition was possible during a purge operation. (Bug #32679024)
- As of this release, the default value of `innodb_change_buffering` is `ALL`. This improves handling of secondary index updates. (WL #16967)

## JavaScript Programs

- Size limit for result sets in SQL-callouts from JavaScript stored procedures has been increased to support larger data sizes. (Bug #38413760)
- JavaScript language support provided by the MySQL Multilingual Engine component now conforms to the ECMAScript 2025 Language Specification (ECMA-262, 15th Edition) as shown at <https://262.ecma-international.org/16.0/>. (Bug #38358516)
- Virtual FileSystem paths were not handled correctly by the MLE component. As of this release, paths with redundant or relative directories, such as `./` or `../`, are accepted without errors. (Bug #38198011)
- MySQL now supports automatic configuration for the `mle.memory_max` variable setting on-premises, utilizing a formula-based approach to dynamically allocate memory. This ensures that the default value of this variable is calculated as 5% of the total physical memory, capped within a range of 0.4GB to 32GB.

For more information, see [MLE Component Memory and Thread Usage](#). (WL #17033)

## JSON Notes

- Parsing very-large, high-precision float numbers in exponential notation, such as `5.0000000000000004e36`, in a JSON column could result in a checksum verification error. (Bug #38033684)
- Large TIME values inserted into a TIME column became smaller when converted to JSON. (Bug #37458212)
- Fixed an issue related to handling JSON key values in the `mysql.user.user_attribute` column when using the `ALTER USER` statement with a JSON attribute containing a blank or `null` key-value pair. (Bug #31067575)

## JSON Duality Views

- Using 0 as an explicit request for `AUTO_INCREMENT` columns was allowed. (Bug #38319365)
- It was possible to delete a singleton child of a JSON Duality View's nested child. As of this release, it is no longer possible to delete a singleton child of a JSON Duality View's nested child. (Bug #38305543)
- Creating a JSON Duality View with `WHERE` conditions that contained sub-queries resulted in an assertion failure, and the server possibly closing unexpectedly. Errors were returned similar to the following:

```
Assertion `nullptr != dynamic_cast<Target>(arg)' failed.
```

(Bug #38187473)

- `INSERT` statements for a JSON Duality View that included an `AUTO_INCREMENT` column were rejected even if the column was not part of a join condition. (Bug #38179925)
- JSON Duality Views exhibited inconsistent behavior with DML queries and `SELECT` queries when the `derived_merge=off` optimizer hint was used. (Bug #38175557)
- Executing a JSON Duality Views `UPDATE` statement could result in the server closing unexpectedly. Errors were returned similar to the following:

```
Address not mapped to object
```

(Bug #38084877)

## Keyring Notes

- `component_keyring_kmip` or `keyring_okv` now support `SECRET` key types on Oracle Key Vault (OKV) server 21.10, or higher. (WL #17090)
- The `keyring_okv` plugin is deprecated and replaced by the `keyring_kmip` component, enabling secure communication with Key Management Interoperability Protocol (KMIP) 1.1-compatible products such as Oracle Key Vault, Gemalto SafeNet KeySecure Appliance, Townsend Alliance Key Manager, and Entrust KeyControl. (WL #16524)

## MySQL Programs

- This version includes the MySQL Diagnostic Monitor (`mysqldm`), a new client tool designed to collect server diagnostic information. To supply to Oracle Support for efficient issue resolution, for example. The `mysqldm` client tool runs a predefined set of queries on the server, storing the results in JSON format, and provides options for customizing the output directory, number of iterations, and delay between iterations.



### Note

MySQL Diagnostic Monitor is only available with MySQL Enterprise Edition.

(WL #16973)

## Optimizer Notes

- Sorting with row IDs could produce incorrect results when the result came from a hash join reading a table in the temptable engine, where the temporary table is large enough to be stored on disk. (Bug #38418831)
- 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 related to query execution. (Bug #37847144)
- Separate evaluation functions for `DATE` and `DATETIME` values have been introduced, utilizing distinct classes for these data types to optimize server performance. (Bug #37743288)
- Fixed an issue relating to SQL queries involving correlated scalar subqueries and JSON tables. (Bug #36684370)
- Queries using prepared statements to retrieve time type data did not always return expected results, regardless of whether precision was specified. (Bug #36065557)
- Using `WHERE IN (NULL, 'xx:xx:xx.xxxxxx')` on a `TIME` column returned an empty set, whereas using `WHERE IN ('xx:xx:xx.xxxxxx')` returned the expected results. (Bug #35500028)
- Using the `IN()` operator with `TIME` columns could return an empty set when a value is matched if the list contains `NULL` or empty strings. (Bug #34188599)
- The default values of the following system variables have changed:
  - `explain_format`: changed from `TRADITIONAL` to `TREE`
  - `explain_json_format_version`: changed from 1 to 2.

(WL #17039)

## 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)
- MySQL Server RPM installation on Fedora 42 could fail due to conflicts with MariaDB 11.8 packages. Errors were returned similar to the following:

```
file /usr/bin/mysqld conflicts between mysql-community-server-8.4.7
and mariadb11.8-server-3:11.8.3, file
/usr/share/man/man1/mysql.1.gz conflicts between mysql-community-
client-8.4.7 and mariadb11.8, file
/usr/lib/systemd/system/mysqld.service conflicts between mysql-
community-server-8.4.7 and mariadb11.8-server
```

(Bug #38384844)

- The RPM specification and DEB control files were missing several requirements, making it impossible to build those formats. (Bug #38310603)
- MySQL Server now supports side-by-side installations of different innovation and LTS releases. (Bug #38104701)

## Performance Schema Notes

- Invalid temporal values in the binary protocol were silently adjusted when `SQL_MODE` was set to a non-strict value. As of this release, such values are rejected with an error. (Bug #38352728)
- The following telemetry system variables were enabled by default, resulting in connection failures and log spamming:
  - `telemetry.log_enabled`
  - `telemetry.metrics_enabled`
  - `telemetry.trace_enabled`

As of this release, these variables are disabled by default. (Bug #38345446, Bug #38347061)

- The default values for the following telemetry endpoint system variables, such as `http://localhost:4318/v1/traces`, caused log spamming and were defined using http instead of https:
  - `telemetry.otel_exporter_otlp_logs_endpoint`
  - `telemetry.otel_exporter_otlp_metrics_endpoint`
  - `telemetry.otel_exporter_otlp_traces_endpoint`

As of this release, these variables have no default value. (Bug #38133218)

- Connections waiting on the connection control plugin were not displayed in the `performance_schema.processlist` table. (Bug #38043683)
- Queries with the same `SQL_DIGEST` could produce identical summaries, even if they had distinct characteristics, such as differing `ORDER BY` clauses. The digest computation has been improved to better distinguish between these queries. (Bug #31103386)

## Pluggable Authentication

- Client-server handshake errors occurred during authentication with the `authentication_oci` plugin, when using the `--default-auth` parameter, if the server and client's default authentication methods were different on first connection. Errors were returned similar to the following:

```
An error occurred during the client server handshake.
```

(Bug #37092159)

## Replication Notes

- MySQL Replication supports enhanced security features with encryption now being enabled for all replication connections by default. With this configuration change, the default value of the `SOURCE_SSL` option in the `CHANGE REPLICATION SOURCE TO` Statement has been changed to `1`, the default value of the `group_replication_ssl_mode` variable has been changed to `REQUIRED`, and the default value of the `group_replication_recovery_use_ssl` variable has been changed to `ON`.

For more information, see [Replication Security](#). (WL #16438)

- To enable enhanced replication capabilities by default, the default values of `gtid_mode` has been changed to `ON`. Additionally, the default value of `enforce_gtid_consistency` has been changed from `ON`.

For more information, see [Replication with Global Transaction Identifiers](#). (WL #8602)

## Spatial Data Support

- Geometry functions now support the inclusion of CRS URN in implicit JSON representation, ensuring that spatial reference system information is preserved when converting geometries to GeoJSON. This enhancement enables accurate representation and updating of geometry values in JSON duality views. It also affects the `ST_AsGeoJSON` function with options 2 and 4, which now always includes a CRS URN. With this update, GeoJSON resulting from an implicit conversion from a geometry type will have an extra "crs" attribute, providing a more comprehensive and accurate geometric representation.

See [Spatial GeoJSON Functions](#) for more information. (WL #16929)

## sys Schema Notes

- Errors encountered by the `sys.ps_trace_thread` stored procedure were not handled properly.

Our thanks to Artem Avetian for the contribution. (Bug #34537707)

- The sys schema stored procedure, `revoke_schema_privileges_from_all_accounts_except()` is now installed by default. This procedure enables you to revoke privileges on a schema, for all users, except for a defined list of exceptions.

See [The revoke\\_schema\\_privileges\\_from\\_all\\_accounts\\_except\(\) Procedure](#) for more information. (WL #16996)

## Thread Pool Notes

- MySQL Thread Pool now supports automatic hardware-based configuration, allowing it to detect the underlying server hardware and set optimal defaults for better performance. The plugin automatically adjusts its settings based on the number of VCPUs available on the system, and users can still make

changes when needed. Additionally, the plugin automatically validates and correct misconfigured parameters, and logs warnings when values are auto-corrected.

The following system variables are affected:

- `thread_pool_size`
- `thread_pool_max_transactions_limit`
- `thread_pool_query_threads_per_group`
- `thread_pool_algorithm`
- `thread_pool_max_unused_threads`

(WL #16621)

## 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)
- Several updates were made to the generated API documentation.  
Our thanks to Daniël van Eeden for the contribution. (Bug #38341735)
- The server sometimes processed table references incorrectly. (Bug #38001000)

## Changes in MySQL 9.4.0 (2025-07-22, Innovation Release)



### Note

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

- [Authentication Notes](#)
- [Character Set Support](#)
- [Compilation Notes](#)
- [Component Notes](#)
- [Configuration Notes](#)
- [Deprecation and Removal Notes](#)
- [Firewall Notes](#)
- [InnoDB Notes](#)
- [Installation Notes](#)
- [JavaScript Programs](#)
- [JSON Duality Views](#)
- [Logging Notes](#)
- [Packaging Notes](#)

- [Performance Schema Notes](#)
- [Vector Data Type](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

## Authentication Notes

- The LDAP SASL authentication plugin could not be installed successfully when the Option Tracker component was already installed. (Bug #37983282)

## Character Set Support

- **Important Change:** `IFNULL()` used in a `LIKE` clause raised `ER_CANT_AGGREGATE_2COLLATIONS`. This was due to the fact that `IFNULL()` returns a binary collation and the collation derivation `NONE`, but `NONE` was not allowed in this context, leading to the error.

We solve this problem by making a slight change in semantics such that we lower the strength of collation derivation `NONE` to be less than the strength of any other collation derivation. This means that, when an expression with derivation `NONE` is used, the other operand determines the comparison collation to be used by `LIKE`. This should have minimal impact on existing functionality, and has required no changes in our existing test suite.

We also change collation aggregation to consider only collations with the same (and highest) strength when determining the result collation. We also rename the previous `IGNORABLE` derivation to `NULL`, since it is used only for nulls, and since it is no longer ignorable given that `NONE` now has lesser strength.

In addition, we leave a gap in the strength values for the former `NONE` strength, in order that as few `COERCIBILITY()` calls as possible return values that are incompatible with the previous implementation.

See [Collation Coercibility in Expressions](#), for more information. (Bug #37285902)

- `CREATE TABLE` with a generated column expression, such as a `CHECK` constraint, which referenced a non-ASCII identifier caused a syntax error if the current client character set was not compatible with UTF-8 (for example, GBK). (Bug #30453221)

## Compilation Notes

- **Group Replication:** Defining `TASK_EVENT_TRACE` in `gcs_profile.in` broke the build of GCS/XCOM in MySQL Server's Group Replication GCS component, returning errors similar to `member reference type 'connection_descriptor *' is a pointer; did you mean to use '->'?`. (Bug #38042851)
- **macOS:** It is now possible to compile the server on MacOS using `-DWITH_KERBEROS`.
- **macOS:** Aligned the buffer used for reading status variables. This fixes a potential issue with MacOS/M1 platforms.
- 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)
- Use `-DWITH_PROTOBUF=system` for relevant `.proto` files, which can reduce build times. (Bug #38022751)

- Removed a warning in `gcs_xcom_networking.cc`. (Bug #38021787)
- The system `bzip2` library is now located by `cmake` before linking with `-lbz2`. (Bug #38005363)
- MySQL Server now requires GCC version 11 or later to compile.

As part of this change, support for ARM systems using RHEL7 is removed in this release, due to the lack of availability of GCC 11 on that platform. (Bug #38004285)

- Removed workaround for old versions of CMake.



#### Note

The minimum version of CMake supported for building MySQL is 3.14.6.

(Bug #37901122)

- 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)
- Warnings raised by Clang 20 for non-trivially copyable types, deprecated literal operators, and incorrect `main` function declarations are no longer generated. (Bug #37785251)
- Removed warnings observed when compiling the server with Clang 20. (Bug #37785251)
- Disabled `clang::musttail` when building with GCC 15. (Bug #37776018)
- Worked around an issue with list handling in certain older versions of CMake. (Bug #37709169)
- Some compiler features tests did not pass when building with `-fprofile-use`. (Bug #37707556)
- On RHEL 9, MySQL Server is now built using GCC 14. (Bug #37702396)
- The version of Boost needed to build MySQL has been raised from 1.85.0 to 1.87.0. (Bug #37403602)
- The linker tried to use the empty Cmake variable `${ICU_LIBRARIES}`, even though the correct library (`ext::icu`) was already linked elsewhere. (Bug #36524167)
- 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)
- 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)

## Component Notes

- When a UDF registered by a component is running, it cannot be unregistered, and so the component cannot be uninstalled (`UNINSTALL COMPONENT` is rejected with an error). When the component was uninstalled twice while the UDF was running, the first attempt failed (as expected), but the second one succeeded, causing the library to be unloaded, leading to issues with the UDF and possibly an unplanned server exit.

This occurred because the component's deinitialization function cleared the container of registered UDFs even when it did not succeed, so the second uninstallation skipped deregistration. We fix this by storing the status of previous registrations in such cases. (Bug #35772996)

- The MySQL server's dynamic loader has been improved such that, when loading a component, its services are no longer registered until after the component's `init()` method has been called. This has the following implications for component behavior:
  - For developers, it is no longer necessary to perform sanity checks to ensure that needed data structures are present before calling the component's service methods.
  - For users, components should be more reliable especially in situations where components are loaded and unloaded with great frequency.

See [MySQL Components](#), for more information. (WL #16918)

## Configuration Notes

- **Microsoft Windows:** An in-place upgrade of MySQL Server using MySQL Configurator failed when a Windows service name other than the default had been used. (Bug #37917039)
- **Microsoft Windows:** When upgrading a server from 8.0 to a higher series, MySQL Configurator did not persist custom server settings in the `my.ini` file. (Bug #37481548)
- **Microsoft Windows:** When upgrading a MySQL Server using MySQL Configurator, the process hung in the "Starting the server and upgrading system tables" step if a custom error log name was used. (Bug #37463478)
- **Microsoft Windows:** MySQL Configurator allowed additions of duplicate users, such as users defined with the same user name and host name, and then created them as different users on the server. With this fix, in GUI mode, a duplicate user is rejected with an error message, and in CLI mode, a duplicate user specified with `--add-user` is ignored. (Bug #37460190)
- **Microsoft Windows:** When used in CLI mode, MySQL Configurator now accepts a file path for the environment variable `MYSQL_PWD`, allowing the password to be specified in a file. (Bug #37460173)
- **Microsoft Windows:** When used in CLI mode, MySQL Configurator always rejected a root password shorter than 4 characters in length. With this fix, the restriction applies only for new configurations. In addition, the password option is no longer mandatory except for new configurations. (Bug #37460061)
- **Microsoft Windows:** As MySQL Enterprise Edition 9.4.0 includes a new [The MySQL Enterprise Firewall Component](#), intended to be a replacement for the now deprecated MySQL Enterprise Firewall plugin, the MySQL Configurator now supports enabling the firewall component and upgrading to it from the firewall plugin:
  - For new server configurations: Users can choose to enable the new firewall component.
  - For server reconfigurations: Users can choose to disable the firewall, enable the new firewall component (if the firewall was not enabled before), keep the old firewall plugin, or upgrade the firewall plugin to the firewall component.
  - For server upgrades: If the firewall was enabled on the server before, users can choose to keep the old firewall plugin, or upgrade it to the firewall component.

See [MySQL Server Configuration with MySQL Configurator](#) and [MySQL Server Configuration with MySQL Configurator](#) for details. (WL #16760)

- The default value of `back_log` is increased to 10000.

See [back\\_log](#) for information on Linux configuration parameters which must be set to make MySQL Server resistant to connection bursts. (WL #16888)

- It is now possible to limit the maximum amount of physical memory used by MySQL Server when determining the default values of configuration variables, using `server_memory`. (WL #16938)

## Deprecation and Removal Notes

- As part of ongoing work to transition from MySQL plugins to MySQL components, the API used to write MySQL plugins is now deprecated and subject to removal in a future version of MySQL. This change has the following effects:
  - The MySQL server's `--early-plugin-load` option is deprecated. Starting the server with this option now raises a deprecation warning.
  - A deprecation warning is now issued whenever a keyring plugin is loaded.

[Keyring Components Versus Keyring Plugins](#), provides a summary of the differences between keyring plugins and keyring components. See also [Key Migration Using a Migration Server](#), for information about migrating from a keyring plugin to a keyring component. (WL #16574)

- The system variable `temptable_use_mmap`, deprecated in version 8.0.26, is removed in this version. (WL #16745)
- The SQL functions `MD5()` and `SHA1()` are deprecated as of this release and scheduled for removal in a future release. (WL #16955)

## Firewall Notes

- **Packaging:** This release includes a MySQL Enterprise Firewall component (see [The MySQL Enterprise Firewall Component](#)) intended to replace the firewall plugin, which is now deprecated and thus subject to removal in a future version of MySQL. The component implements most of the plugin's functionality, with the exception of account profiles, which are deprecated in the plugin and not supported by the component.

If you are not already using MySQL Enterprise Firewall and wish to perform a clean installation of the firewall component, see [Installing the MySQL Enterprise Firewall Component](#). For help with upgrading a current installation of the firewall plugin, see [Upgrading to the MySQL Enterprise Firewall Component](#). Prior to upgrading the firewall plugin to the firewall component, you must convert any account profiles you may be using with the plugin to group profiles; [Migrating Account Profiles to Group Profiles](#); provides assistance with this task.

MySQL Enterprise Firewall is a commercial feature available as part of MySQL Enterprise Edition; see [MySQL Enterprise Firewall](#), for more information. (WL #16570)

## InnoDB Notes

- Memory management has been improved to prevent potential memory leaks, which could occur in certain error handling scenarios. (Bug #37826893)
- The `information_schema.innodb_cmp_per_index` table sometimes returned unknown values for database name and table name under certain conditions, such as when tables and indexes were evicted from the cache. (Bug #37820227)
- Fixed an issue relating to importing tables. (Bug #37621360)
- The default value of `innodb_change_buffer_max_size` has been changed to 5. This update aims to strike a balance between the benefits of change buffering for IO-bound workloads and the potential negative impact on in-memory workloads when a larger portion of the buffer pool is dedicated to change buffering. (WL #16968)



You can compile WebAssembly from code written in C, C++, or any other language that can be built with LLVM (see <https://llvm.org/>), using Emscripten (see <https://emscripten.org/>) or another compiler toolchain which supports WebAssembly as a target.

Once you have created the WebAssembly library, you can import it into a MySQL JavaScript program like this:

```
mysql> CREATE PROCEDURE test_wasm_lib() LANGUAGE JAVASCRIPT
-> USING (test_lib)
-> AS $$
->   console.clear()
->   console.log("test_lib keys: ", Object.keys(test_lib))
-> $$;
Query OK, 1 row affected (0.01 sec)
```

Calling `test_wasm_lib()` and then `mle_session_state()` produces a list of functions provided by the library, as shown here:

```
mysql> CALL test_wasm_lib();
Query OK, 1 row affected (0.01 sec)

mysql> SELECT mle_session_state("stdout");
+-----+
| mle_session_state("stdout") |
+-----+
| test_lib keys: __indirect_function_table,_emscripten_stack_restore,
_initialize,add_to_state,emscripten_stack_get_current,memory,
my_add_int |
+-----+
1 row in set (0.00 sec)
```

WebAssembly code is parsed during execution of `CREATE LIBRARY`. In the output of `SHOW CREATE LIBRARY`, WebAssembly code is shown as was used in the original `CREATE LIBRARY` statement, as a `utf8mb4` (base64 encoding) or hexadecimal (binary encoding) string. WebAssembly code is *not* displayed in the `LIBRARY_DEFINITION` column of the Information Schema `LIBRARIES` table.

As part of this work, the following objects are implemented in MySQL JavaScript programs:

- `WebAssembly`
- `TextEncoder`
- `TextDecoder`

APIs specific to MySQL (such as using `session` or `ml`) are not supported in WebAssembly libraries. The WASI API (see <https://wasi.dev/>) is not currently supported; this means that MySQL WebAssembly libraries cannot make use of system, clock, or I/O calls.

WebAssembly libraries are supported as a feature by the MySQL Option Tracker component (MySQL Enterprise Edition—see [Option Tracker Supported Components](#)).

For further information and examples, see [Using WebAssembly Libraries](#). (WL #16794, WL #16824, WL #16834, WL #16951)

- JavaScript language support provided by the MySQL Multilingual Engine component now conforms to the ECMAScript 2024 Language Specification (ECMA-262, 15th Edition) as shown at <https://262.ecma-international.org/15.0/>. (WL #16887)

## JSON Duality Views

- MySQL now supports JSON duality views, which provide a way to expose data stored in relational tables as JSON documents. Such views can be created, altered, dropped, and viewed using [CREATE JSON DUALITY VIEW](#) and [ALTER JSON DUALITY VIEW](#) (both implemented in this release); [DROP VIEW](#) and [SHOW CREATE VIEW](#) now work with JSON duality views as well as SQL views.

For users of MySQL Enterprise Edition, JSON duality views can be modified using [INSERT](#), [UPDATE](#), or [DELETE](#) statements, so that updates of such a view cause corresponding updates on the base table. DML operations on JSON duality views are supported in MySQL Enterprise Edition only.

The [SELECT](#) from a JSON duality view includes a 128-bit hash value generated for the document as part of an [ETAG\(\)](#) field in the metadata section (see [JSON Duality Views—Concurrency](#)).

See the following example:

```
mysql> SELECT * FROM jdvl;
+-----+
| data |
+-----+
| {"_id": 1, "key": "base64:type252:bG9uZl9ibG9iX2RhdGE=", "_metadata": {"etag": "0bb9af42e15001173759fb" |
+-----+
```

See the description of this function for more information.

You can obtain information about existing JSON duality views from the following Information Schema tables which have been implemented in this release:

- [JSON\\_DUALITY\\_VIEWS](#): Provides per-view information about JSON duality views.
- [JSON\\_DUALITY\\_VIEW\\_TABLES](#): Provides information about tables referenced by JSON duality views.
- [JSON\\_DUALITY\\_VIEW\\_COLUMNS](#): Provides information about columns defined in JSON duality views.
- [JSON\\_DUALITY\\_VIEW\\_LINKS](#): Describes parent-child relationships between JSON duality views and their base tables.

In addition, for users of MySQL Enterprise Edition, the JSON duality views feature is supported by the Option Tracker component (see [Option Tracker Supported Components](#)).

See [JSON Duality Views](#), for more information and examples. (WL #16616, WL #16617, WL #16618, WL #16619, WL #16623)

## Logging Notes

- MySQL Server now logs the total number of logical CPUs and physical memory accessible to the server in the error log. This information is always logged, regardless of the log verbosity. (WL #16940)

## 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)

## Performance Schema Notes

- The internal `pfs_get_thread_statement_locker_vc()` function contained debugging code that was inadvertently included in release builds. (Bug #37743667)

## Vector Data Type

- The `VECTOR_TO_STRING()` function did not set its output collation correctly. (Bug #37815490)

## Functionality Added or Changed

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

This option is disabled by default. To enable it, start the client with `--commands` or `--commands=ON`.

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.

- **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.

- **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)

- **NDB Replication:** It is now possible to divide binary logging for a MySQL Cluster or for individual `NDBCLUSTER` tables into equal portions or “slices” between multiple MySQL servers.

For dividing binary logging for an entire cluster into slices, this `NDB` release implements two `mysqld` startup options. The `--ndb-log-row-slice-count` option determines the number of slices, and thus the number of servers sharing binary logging. `--ndb-log-row-slice-id` identifies the slice for which this MySQL server is responsible. See the descriptions of these options for more information.

Division of binary logging into slices can be performed for a specific `NDB` table by adding rows to the `ndb_replication` table with appropriate values for the `binlog_row_slice_count` and `binlog_row_slice_id` columns which have been added to this table. If you are upgrading an existing setup, you may need to perform an `ALTER TABLE` statement, or drop and re-create the table to obtain this functionality. For further information and examples, see [Per-table binary log slicing](#). (WL #15413)

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

- **Performance:** Redundant conditions in some queries optimized away in MySQL 8.0 were no longer removed in later versions, leading to a significant drop in the performance of such queries. (Bug #117907, Bug #37808260)

References: This issue is a regression of: Bug #30112096.

- **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:** A check whether the table is referenced by a foreign key was performed for every row update, even when it was not required.

The check is no longer performed unless it is required. (Bug #37867653)

- **InnoDB:** The return value from `pthread_setaffinity_np` was not properly interpreted when setting processor affinity for threads during buffer pool creation. As a result, no error was returned even if setting affinity failed. (Bug #37825544)
- **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:** MySQL Community Edition binaries included unnecessary OpenTelemetry symbols, due to unprotected static arrays. (Bug #37689163)
- **InnoDB:** The double write buffer was unnecessarily large. When calculating the number of segments per file, one extra segment was always added, whereas an extra segment should be added only if the number of `Double_write` instances is not divisible by the number of `dblwr` files. (Bug #37684656)
- **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:** The `temptable` handler did not terminate cleanly during server shutdown, potentially causing the server to terminate unexpectedly. This issue has been addressed by implementing a mechanism to track and properly clean up temptable objects associated with each thread. (Bug #36538081)
- **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:** Fixed an internal memory management issue in `libs/mysql/binlog/event/event_reader.cpp`. (Bug #37371443)
- **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:** The `Gr_empty_consensus_proposals_count` system status variable was not updated as expected. (Bug #37937927)
- **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.

- RPM installations on Fedora 24 could not be completed because conflicting packages were pulled from upstream. This fix adds the proper obsoletes to block the conflicting packages. (Bug #37976913)
- In certain cases, `expr BETWEEN expr AND expr` (where `expr` is a complex expression) led to an assert in `sql/sql_base.cc`. (Bug #37952274)

References: This issue is a regression of: Bug #113506, Bug #36137690.

- Some triggers which called stored routines did not always execute correctly. (Bug #37915445)
- The internal function `transform_table_subquery_to_join_with_derived()` did not restore the current query block properly following invocation, leading to an assert. (Bug #37884336)

References: See also: Bug #37832605.

- Removed a potential memory leak in `item_cmpfunc.cc`. (Bug #37883669)
- Using the `HLL()` function (MySQL MySQL HeatWave only) with a `CAST()` operation to `UNSIGNED` on a column, along with a `STRCMP()` operation and a `NULL` in the `WHERE` clause, now behaves as expected. Previously, this returned errors similar to `Assertion '!null_value || is_nullable()' failed`. (Bug #37839325)
- A view using a CTE which contained a subquery was not always handled correctly. (Bug #37832605)
- The server failed to start in a Docker container if `component_keyring_encrypted_file` or `component_keyring_file` was enabled and `binlog_encryption` was set to `ON`, due to issues with accessing the keyring data file. (Bug #37821740)
- Removed a warning caused by an assertion that set rather than compared a value in `sql/opt_costconstantcache.cc`. (Bug #37814484)
- Installing Oracle packages for MySQL on an Ubuntu 25.04 system was not possible where MySQL was already installed using the system's APT repositories. (Bug #37804480)
- 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.

- Hit an assert in `INTERSECT` when the computed number of chunks exceeded 65535, which can happen if the number of estimated rows is very high, or because the `set_operations_buffer_size` system variable is set too low.

This fix adds a missing check for this situation. If we need more than 2 chunk file sets, we revert to index temp table based deduplication. To alert the user, this information is presented in a note for the query, and in the optimizer trace.

As part of this fix, the block size used for `set_operations_buffer_size` is increased from 128 to 1024. (Bug #37742092)

- 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)
- Removed an assertion from `sql/sql_resolver.cc`. (Bug #37601389)
- A subquery which was part of a condition that had more than one subquery and whose strategy was already finalized as `IN-to-EXISTS` was checked for materialization, leading to problems. The check for materialization was made because one of the other subqueries which was part of the condition had its strategy finalized as `MATERIALIZATION`. We address this by adding checks so that only those subqueries which should be materialized are looked into. (Bug #37587388)
- 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.

- `mysql_secure_installation` did not check whether the root passwords had expired, as expected. (Bug #37563088)
- 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.

- Removed a performance regression introduced by work done in MySQL 9.2.0 on the caching SHA-2 authentication plugin. (Bug #37523469)
- `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.

- On some Windows systems, after installing MySQL Server 9.1 with the MSI installer, the server failed to start. This was because those systems did not have Visual C++ Redistributable v.14.42 or later installed. The MySQL Server MSI installer now checks for that requirement and refuses to start installation unless it is satisfied. (Bug #37365476)

- 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.

- Removed a potential memory leak from the `keyring_aws` plugin. (Bug #36684413)
- 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.

- Setting `max_join_size` led to improper processing of some nested queries. (Bug #35625769)
- 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)
- `EXPLAIN ANALYZE FORMAT=JSON` did not handle queries with subqueries correctly when `explain_json_format_version` was equal to 1.

Our thanks to Peiyuan Liu and the Tencent team for the contribution. (Bug #117995, Bug #37285902)

- It was possible to use `PERSIST` or `PERSIST_ONLY` with `SET TRANSACTION ISOLATION LEVEL`, even though this should not be allowed, and later caused errors on server restart. Now attempting to do so causes the statement to be rejected with `ER_GRP_RPL_UNSUPPORTED_TRANS_ISOLATION`. (Bug #115619, Bug #36854635)
- 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)
- `MIN()`, when used as a window function, did not ignore nulls as expected. (Bug #113631, Bug #36182490)
- It was possible in a window frame specification using `RANGE` frame units to have an `ORDER BY` expression containing `RAND()`. This broke an invariant in the `RANGE` frame specification; that the values be monotonically ascending or descending. We solve this issue by disallowing a non-deterministic `ORDER BY` expression when combined with `RANGE` units; this is now rejected with an error.

This also fixes a related issue in which `RANGE` unit comparison failed when computing the frame limits for `BETWEEN CURRENT ROW AND after_value FOLLOWING` where the row value being compared was unsigned and could be less than `after_value`, leading to underflow and a possible incorrect result. In such cases, we now reject the operation with an error.

See [Window Function Frame Specification](#), for more information. (Bug #111510, Bug #35521787)

## Changes in MySQL 9.3.0 (2025-04-15, Innovation Release)

- [Account Management Notes](#)
- [Audit Log Notes](#)
- [Authentication Notes](#)
- [Compilation Notes](#)
- [Component Notes](#)
- [Configuration Notes](#)
- [Deprecation and Removal Notes](#)
- [Doxygen Notes](#)
- [SQL Function and Operator Notes](#)
- [INFORMATION\\_SCHEMA Notes](#)
- [InnoDB Notes](#)
- [JavaScript Programs](#)
- [JSON Notes](#)
- [MySQL Enterprise Notes](#)
- [Optimizer Notes](#)
- [Performance Schema Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

## Account Management Notes

- It was possible in some cases to grant a user the `EXECUTE` privilege, but subsequently to be unable to revoke it from the same user. (Bug #37570206)

## Audit Log Notes

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

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

## Authentication Notes

- This release introduces a MySQL component, `component_keyring_hashicorp`, which takes the place of the HashiCorp Vault keyring plugin; that plugin and its associated system variables are now deprecated (and subject to removal in a future version of MySQL). Like the plugin, the component allows communication with HashiCorp Vault for backend storage, and supports HashiCorp Vault AppRole authentication, but has been reimplemented using the MySQL component architecture.

For more information, including information about installation and configuration of the HashiCorp Vault keyring component, see [Using the HashiCorp Vault Keyring Component](#). For additional information about MySQL keyring components, see [Keyring Component Installation](#), as well as [Security Components and Plugins](#). For general information about MySQL components, see [MySQL Components](#).

`component_keyring_hashicorp` plugin is an extension included in MySQL Enterprise Edition, a commercial product. To learn more about commercial products, see <https://www.mysql.com/products/>. (WL #16505)

## 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)
- **Linux:** Use `--experimental_allow_proto3_optional` when building with `protoc` version 3.14 or earlier. (Bug #37579947)
- **Microsoft Windows:** The server could not be built using Visual Studio 17.13.1 with `MSVC_CPPCHECK` enabled. (Bug #36925076)
- Upgraded the bundled Curl library to version 8.12.1. (Bug #37633587)
- Disabled the `-ftls-model=initial-exec` option when compiling MySQL on FreeBSD. (Bug #37613105)
- Abseil could not be built on FreeBSD. (Bug #37611924)
- Do not read `share/charsets/Index.xml` when running `comp_err`. (Bug #37569683)
- The file `mysql_version.cmake` was included multiple times. (Bug #37559512)
- Removed the unused file `strings/utrl1-dump.cc`. (Bug #37549844)
- The bundled version of `opentelemetry-cpp` was upgraded to version 1.19.0. (Bug #37506554)
- Fixed a large number of warnings generated by clang-tidy. (Bug #37471922)
- `include/my_systime.h` included `std::chrono`, which was unneeded, and which has now been removed. (Bug #37458343)
- 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)
- The bundled version of `opentelemetry-cpp` was upgraded to version 1.18.0. (Bug #36708755)
- Added documentation for unused bytes at the end of `Protocol::ColumnDefinition41`.  
Our thanks to Daniyaal Khan for the contribution. (Bug #117346, Bug #37541403)

## Component Notes

- **Important Change:** As of this release, MySQL Enterprise Data Masking and De-Identification, part of MySQL Enterprise Edition, is now known as *MySQL Enterprise Data Masking*.  
Our documentation, beginning with the 9.2 edition, has been updated to reflect this change.

For more information, see [MySQL Enterprise Data Masking](#). (WL #16721)

- **Group Replication:** Some of the entries in package specification files for normal and debug files for the Flow Control Statistics component were misplaced.

The Flow Control Statistics component is available as part of MySQL Enterprise Edition. For more information, see [Group Replication Flow Control Statistics Component](#). (Bug #37486491)

- **Group Replication:** Added the Group Replication Primary Election component, which makes it possible to specify the most-up-to-date selection method for choosing a new primary in the event of failover. The Group Replication plugin is a prerequisite for this component, which must be installed on each group member. In addition, the `group_replication_elect_prefer_most_updated.enabled` system variable must be set to `ON`, on each group member, in order for the component to function.

This component also provides two status variables for monitoring purposes.

`Gr_latest_primary_election_by_most_uptodate_members_trx_delta` was the difference in the number of transactions between the new primary and the secondary most up to date with it, when the most-up-to-date primary selection method was last used.

`Gr_latest_primary_election_by_most_uptodate_member_timestamp` provides a timestamp for the most recent election of a new primary using the most-up-to-date selection method.

When a new primary is elected, the component records the event in the log. This information in the log entry includes a timestamp, the UUID of the promoted secondary, and the method used to select the new primary: either the most-up-to-date method (and how many transactions this secondary was behind the primary), or weighting (with the weight that was assigned to the secondary chosen).

For more information about this component, see [Group Replication Primary Election Component](#).

This component is part of MySQL Enterprise Edition, a commercial offering. See [MySQL Enterprise Edition](#), for more information. (WL #16432)

- The following enhancements have been made in this release to the MySQL Option Tracker component, part of MySQL Enterprise Edition:
  - Each feature that supports the Option Tracker now provides a global status variable named `option_tracker_usage:feature_name`, which provides a count of the number of times a feature has been used. This variable is provided whether or not the Option Tracker component is installed.

You can view these status variables using `SHOW GLOBAL STATUS LIKE 'option_tracker_usage%'` or by selecting from the Performance Schema `global_status` table.

- The Boolean `used` key in the JSON format employed for usage data has been replaced with a counter, `usedCounter`. After upgrading to this release, no `used` members are added or updated in this data by the Option Tracker.

See [Option Tracker Status Variables](#), as well as [Option Tracker Supported Components](#), for more information. (WL #16721)

## Configuration Notes

- **Microsoft Windows:** The `--install-sample-database` option was ignored by the *MySQL Configurator CLI*. (Bug #37701034)
- **Microsoft Windows:** *MySQL Configurator CLI* did not execute any actions specified with the `--action` option, except for `configure`. (Bug #37473745)

- **Microsoft Windows:** During an upgrade from MySQL 9.1.0 to 9.2.0, the *MySQL Configurator* failed to find the `my.ini` configuration file and required manual selection of its file path. (Bug #37468826)
- **Microsoft Windows:** Upgrading an existing server with a non-default instance port by the *MySQL Configurator CLI* failed, unless the `--old-instance-protocol` option was used in the command. (Bug #37459238)

## Deprecation and Removal Notes

- **NDB Cluster:** The `--restore-privilege-tables` option for `ndb_restore`, which was deprecated in NDB 8.0.16, has now been removed. (Bug #36298807)
- **Replication:** The `replica_parallel_workers` server system variable can no longer be set to 0; the minimum permitted value is now 1. (WL #13957)
- The system variable `innodb_undo_tablespaces`, deprecated in MySQL 8.0.14, has been removed. (WL #16746)
- The system variables `innodb_log_file_size` and `innodb_log_files_in_group`, deprecated in MySQL 8.0.30, have been removed. (WL #16743)
- The Version Tokens plugin, which was deprecated in MySQL 9.2.0, has been removed in this release. (WL #16614)

## Doxygen Notes

- Addressed issues in the server source code documentation as noted here:
  - The index page linked to the MySQL 8.0 Manual. This fix avoids the versioning issue here by using an unversioned link instead.
  - In `protocol_classic.cc`, the parameter type and flag descriptions were unclear, and have been improved.

Our thanks to Daniël van Eeden for the contributions. (Bug #117391, Bug #117503, Bug #37559971, Bug #37607749)

- Addressed the following issues in the server source code documentation for `COM_STMT_PREPARE` response packets:
  - The payload specification showed the wrong condition for checking the warning count.
  - Removed extraneous pipe characters (|) from the first example.

Our thanks to Kanno Satoshi for the contribution. (Bug #117373, Bug #37552681)

- Addressed the following issues in the server source code documentation:
  - Corrected misspelling in the MySQL Client/Server Communication protocol documentation for `AuthSwitchRequest`.
  - `protocol_classic.cc`: Table for distinguishing `OK` packets from `EOF` packets used the wrong operator; this has now been corrected.
  - Documented previously undocumented behavior regarding the sending of default values in `COM_FIELD_LIST`.

Our thanks to Daniyaal Khan for the contributions. (Bug #117325, Bug #117374, Bug #117596, Bug #37534532, Bug #37552684, Bug #37645678)

## 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)
- **InnoDB** now supports container-aware resource allocation, allowing it to adhere to the restrictions imposed by the container. The default values of **InnoDB** configurations are now calculated based on the logical CPUs and physical memory allocated by the container, rather than relying on system-wide resources.

The values for the following system variables are calculated and set based on those resource limits:

- The values of the following are calculated based on the number of logical CPUs:
  - `innodb_buffer_pool_instances`
  - `innodb_page_cleaners`
  - `innodb_purge_threads`
  - `innodb_read_io_threads`
  - `innodb_parallel_read_threads`
  - `innodb_redo_log_capacity` (value set only if `--innodb-dedicated-server` is enabled.)

- `innodb_log_writer_threads`
- The values of the following are calculated based on the available memory:
  - `temptable_max_ram`
  - `innodb_buffer_pool_size` (value set only if `--innodb-dedicated-server` is enabled.)

See [Container Detection and Configuration](#). (WL #16484)

## JavaScript Programs

- **Important Change:** JavaScript stored programs now fully support the `DECIMAL` type, including its alias `NUMERIC`; can now be used with JavaScript programs as input arguments, output arguments, prepared statement `bind()` parameters, and return values.

In order to maintain precision, MySQL `DECIMAL` is converted to JavaScript `String` by default, but it is possible to override this behavior, causing it to be converted to `Number` instead by setting the value of the `decimalType` option to `NUMBER` (or `mysql.DecimalType.NUMBER`).

It is possible to convert JavaScript `Boolean`, `Number`, `String`, and `BigInt` values to `DECIMAL` (or `NUMERIC`). Trying to convert any other JavaScript type to a MySQL decimal type is not supported, and is rejected with an error.

See [Conversion to and from MySQL DECIMAL and NUMERIC](#), for additional information and examples.

JavaScript stored program support requires the Multilingual Engine Component (MLE), available with MySQL Enterprise Edition. See [Multilingual Engine Component \(MLE\)](#). (WL #16747)

- **Important Change:** Dynamic import of JavaScript libraries is now supported using the `await` operator. This means you can use constructs such as that shown here to insure that libraries are loaded before using them:

```
let module = await import(`/db1/lib_${object_type}`)
return module.default.print() // assume this method is defined for each lib
```

See [Using JavaScript Libraries](#), for more information and examples. (WL #16733)

- Upgraded the MLE component to use GraalVM Truffle version 24.2.0. (Bug #37668857)
- Importing a library with a global wait led to an internal error. (Bug #37425528)
- An issue with error handling led to an assert in `sql/sql_class.cc`. (Bug #36777428)
- The `mle_session_reset()` function has been enhanced with an optional string argument which takes one of the values `"stderr"`, `"stdout"`, or `"output"` to clear `stderr`, `stdout`, or both, respectively.

When called without an argument, `mle_session_reset()` behaves exactly as it did in previous versions of MySQL: it clears both `stderr` and `stdout`, resetting the session time zone and clearing the stack trace as it does so; this removes any observable output from `mle_session_state()`.

For more information, see [JavaScript Stored Programs—Session Information and Options](#).

`mle_session_reset()` is provided by the MLE component, part of MySQL Enterprise Edition. See [Multilingual Engine Component \(MLE\)](#), for more information about this component. (WL #16660)

- MySQL JavaScript programs now provide localization and internationalization of numbers, dates, and other values by supporting the `Intl` global object. MySQL locales map to JavaScript locales by substituting a dash character for the underscore; for example, setting `lc_time_names = "ja_JP"` means that JavaScript returns `"ja-JP"` for the locale.

It is also possible to override the session or default locale within a stored program by calling an object's `toLocaleString()` method or making use of one of the `Intl` formatting objects.

The locale in effect the first time a JavaScript stored program is invoked in a given session remains the default locale for that stored program (unless it explicitly overrides the locale) until the session is reset, even after setting `lc_time_names` to a new value.

For more information and examples, see [JavaScript Localization and Internationalization](#). (WL #16709)

- This release includes a number of changes and additions relating to JavaScript library SQL:
  - The statements `ALTER PROCEDURE` and `ALTER FUNCTION` now accept a `USING` clause and so can add, replace, or remove a list of libraries imported by the named JavaScript stored procedure or stored function. See the descriptions of these statements for more information and examples.
  - `CREATE LIBRARY` now supports a `COMMENT` clause. This comment is shown in the output of `SHOW CREATE LIBRARY` and `SHOW LIBRARY STATUS` (see next item). It is also shown in the `ROUTINE_COMMENT` column of the Information Schema `ROUTINES` table; the `LIBRARIES` table also adds a `LIBRARY_COMMENT` column for displaying this value.
  - This release also implements a `SHOW LIBRARY STATUS` statement which provides basic information about one or more JavaScript libraries. Like `SHOW PROCEDURE STATUS`, this statement supports `LIKE` and `WHERE` clauses for filtering the output.
  - The `ALTER LIBRARY` statement implemented in this release makes it possible to update the comment for a JavaScript library.
  - The status variables `Com_alter_library` and `Com_show_library_status`, added in this release, provide counts of `ALTER LIBRARY` and `SHOW LIBRARY STATUS` statements, respectively.

For general information about JavaScript stored programs, see [JavaScript Stored Programs](#). See also [Multilingual Engine Component \(MLE\)](#). (WL #16737)

## JSON Notes

- It was possible to create a table including a `JSON` column with `DEFAULT ''` under the default `sql_mode`, but the output from `SHOW CREATE TABLE` for this table, when run on another `mysqld`, resulted in the error `BLOB, TEXT, GEOMETRY or JSON column ... can't have a default value`, even when the `sql_mode` of the second `mysqld` was also non-strict.

This issue did not occur with `BLOB` or `TEXT` columns. (Bug #116479, Bug #37219226)

- A query using `WHERE EXISTS( SELECT ... FROM JSON_TABLE(...) )` did not return the expected result. (Bug #114897, Bug #3666073)

## MySQL Enterprise Notes

- The Option Tracker component has added support for two MySQL features—the traditional MySQL Optimizer, and the MySQL Hypergraph Optimizer (available in MySQL HeatWave only).

See [Option Tracker Supported Components](#), for more information. (WL #16548)

## Optimizer Notes

- Previous versions of MySQL, when the `subquery_to_derived` optimization was enabled, supported transformation into an inner or outer join with a derived table corresponding to a subquery using one of the quantified comparison operations `=ANY` (equivalent to `IN`) or `<>ALL` (equivalent to `NOT IN`) in the `WHERE` clause of a query. This release extends the functionality of this optimizer switch in two ways:
  1. All such comparisons are now supported (`>ANY`, `>=ANY`, `<ANY`, `<=ANY`; `>ALL`, `>=ALL`, `<ALL`, `<=ALL`; `=ANY`, `<>ALL`).
  2. The transformation of such comparisons is now supported in the `SELECT` clause and in the `WHERE` clause.

For more information and examples, see [Optimizing ANY and ALL Subqueries](#). (WL #13052)

References: See also: Bug #37616992.

## Performance Schema Notes

- The `PERFORMANCE_SCHEMA` service thread v7 was not exposed, preventing its use by components. (Bug #37579218)
- You can now configure a network namespace for your telemetry endpoints on Linux platforms. The following system variables are added:
  - `telemetry.otel_exporter_otlp_traces_network_namespace`
  - `telemetry.otel_exporter_otlp_metrics_network_namespace`
  - `telemetry.otel_exporter_otlp_logs_network_namespace`

(WL #16735)

## Functionality Added or Changed

- **Important Change:** Beginning with this release, it is no longer possible to downgrade between individual MySQL Innovation series releases, even within the same series. For example, were a version 9.3.1 to be released, it would not be possible after upgrading to it to downgrade back from MySQL 9.3.1 to 9.3.0. (Bug #37387488)
- **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:** The output process of the `mysql` client for binary values (printed as hexadecimal strings) in query results has been optimized, speeding up slightly the output of large result sets containing binary values. (Bug #37334107)
- Added Enterprise Linux 10 (EL10) support. (Bug #37592019)
- The signature of the handler on `drop_database_t` API has been changed: instead of the database path, it now accepts the database name as a parameter. The change makes the API more efficient. (Bug #37191149)
- The `mysqldump` utility can now provide logical dumps of information about user accounts, writing the appropriate `CREATE USER` and `GRANT` SQL statements to the dump, when run with the `--users` command line option introduced in this release.

You can cause the `CREATE USER` statements generated by `mysqldump` to be preceded by `DROP USER` by including the `--add-drop-user` option as well.

It is also possible to include or exclude specific user accounts from the dump using either of the options `--include-user` or `--exclude-user`.

For more information and examples, see the descriptions of the options cited, as well as [mysqldump — A Database Backup Program](#). (Bug #28038954, WL #15658)

- The `mysql` client now displays query execution times with three decimal places of precision to show milliseconds.

Our thanks to Marcelo Altmann for the contribution. (Bug #117270, Bug #37510263)

## Bugs Fixed

- **InnoDB:** Under certain circumstances, `Trx_by_id_with_min::insert()` during `get_better_lower_bound_for_already_active_id()` could set an incorrect `s_lower_bound` value. (Bug #37548045)
- **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:** Under certain circumstances, an assertion failure occurred if the InnoDB engine performed unnecessary conversions for end range checks.

This resulted in an error similar to the following:

```
Assertion failure: lob0lob.cc:897:trx == nullptr || trx->is_read_uncommitted()
```

(Bug #35006212)

- **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:** Removed a potential race condition in [rpl\\_opt\\_tracker.cc](#). (Bug #37644518)
- **Replication:** When the log sanitizer analyzes relay log files, it first searches for a starting point (such as a rotation event or transaction end), but in some cases, it was possible for a binary log file containing a needed GTID to be deleted as having no relevant data; this resulted in the point-in-time recovery process hanging indefinitely while waiting for the missing GTID to be applied. Now in such cases, the analysis skips parsing of transaction boundaries until the start point is established. (Bug #37635908)
- **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:** It was found during testing that it was possible to force the process responsible for termination of replica threads to access a deleted object. (Bug #37375269)
- **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:** Removed redundant GCS code which tested the same conditional variable twice in succession. (Bug #37538338)
- **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)
- Cleaned up code in `overflow_bitset.h`. (Bug #37591520)
- Use `std::string_view` rather than `std::string` when looking up character sets and collations, which saves on memory allocation and deallocation. (Bug #37586193)
- Removed code left unused after a previous fix. (Bug #37574896)

References: This issue is a regression of: Bug #28956360.

- Use `std::string::starts_with()` instead of the `starts_with()` function defined in `strings/ctype.cc`, and remove the latter function as no longer needed. (Bug #37568373)
- `EXPLAIN` did not always handle subqueries correctly. (Bug #37560280)
- Collation name aliases were sometimes handled in case-sensitive fashion. (Bug #37554688)

References: This issue is a regression of: Bug #36878077.

- 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)
- If a server was installed on an Enterprise Linux platform using RPM packages, after installing `component_log_sink_json`, trying to set `log_error_services` resulted in an error. This was due

to a permission issue with the file path of the JSON log file, which has been fixed by this patch. (Bug #37508168)

- Attempting to install an unknown component using `INSTALL COMPONENT` was not always handled correctly. (Bug #37437317)
- For user input such as `COLLATE utf8_bin` we perform an alias lookup to find the actual collation (in this case, `utf8mb3_bin`). Now we use this name, rather than the input string, when reporting SQL errors. (Bug #37412963)
- Removed the internal `binary_keyword` variable, which was not actually used. (Bug #37408338)
- In `libmysqld`, errors were not correctly handled in `udf_handler::add()` for aggregates. (Bug #37398919)
- Removed the potential for undefined behavior in certain cases from the internal function `check_if_server_ddse_readonly()`. (Bug #37394933)
- The internal function `recover_innodb_upon_upgrade()` was no longer used, and has been removed. (Bug #37394850)
- The Audit Log plugin did not handle errors correctly when writing JSON output.  
See [MySQL Enterprise Audit Plugin \(Deprecated\)](#), for more information. (Bug #37370439)
- `ER_SERVER_OFFLINE_MODE` was not always handled correctly. (Bug #37355755)
- 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)
- Removed an error found in the preparation of stored functions. (Bug #36684438)
- 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)
- Errors relating to `SET` subqueries were not handled correctly. (Bug #36335695)
- 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)
- `PARTITION BY ... (DEFAULT (column))` was not always handled correctly. (Bug #35044654)

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

- 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)
- Corrected an uninitialized variable in `sql/statement/protocol_local_v2.cc`. (Bug #117541, Bug #37622633)
- The `LPAD()` function did not return the correct value when given an empty string enclosed in double quotes unless the string's length exceeded the specified length variable. (Bug #117227, Bug #37498117)
- The null-safe equality operator (`<=>`) showed unexpected behavior when comparing multiple columns (`ROW` values) containing `NULL`. Fixed by simplifying the implementation of the operator. (Bug #117168, Bug #37462769)
- 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 using `MAX()` as a window function, it returned `NULL` for the first row within the window frame, despite data existing in the first row. This happened when the start of the window frame was defined using `N FOLLOWING`, and the frame was ordered by the same expression as the argument of `MAX()`, possibly differing only in syntax such as aliases or table references, in descending order. A sequence of statements demonstrating the issue is shown here:

```
CREATE TABLE t0 (c0 INT);

INSERT INTO t0 VALUES (1), (2);

SELECT
c0, MAX(c0) OVER (ORDER BY c0 DESC ROWS BETWEEN 1 FOLLOWING AND 1 FOLLOWING)
FROM t0;
```

We fix this by making sure that the first row number in the frame is set in the appropriate place in the program logic. (Bug #117013, Bug #37466984)

- Removed a double space within the `INSERT IGNORE` statements generated by `mysqldump`.  
Our thanks to Pieter Oliver for the contribution. (Bug #116845, Bug #37353658)
- The types of all ACL variables used internally have been changed to `Access_bitmask`.  
Our thanks to Mike Wang for the contribution. (Bug #116737, Bug #37318159)
- An error in `include/assert_grep.inc` could lead to erroneous results from any file that included it.  
Our thanks to Ke Yu for the contribution. (Bug #116239, Bug #37105430, Bug #37675340)
- If one client session had a uncommitted transaction that caused a `DROP TABLE` statement in another client session to be blocked, a third client session hung when trying to issue a `USE DATABASE` statement. (Bug #115706, Bug #36892499)
- The maximum for `ssl_session_cache_timeout` was defined as 84600 rather than 86400, and 84600 was stated erroneously to be the length of the day in seconds, in `sql/ssl_init_callback.cc`.  
Our thanks to Pika Mander for the contribution. (Bug #115165, Bug #37354555)
- Removed a memory leak from the `mysqldump` client. (Bug #111793, Bug #35621833)
- Removed the unused `InnoDB` and `NDB` handlers on `get_tablespace()` method. (Bug #109443, Bug #34916556)

## Changes in MySQL 9.2.0 (2025-01-21, Innovation Release)

- [Account Management Notes](#)
- [C API Notes](#)
- [Character Set Support](#)
- [Compilation Notes](#)
- [Component Notes](#)
- [Configuration Notes](#)
- [Deprecation and Removal Notes](#)
- [Doxygen Notes](#)
- [Firewall Notes](#)
- [Installation Notes](#)
- [JavaScript Programs](#)
- [MySQL Enterprise Notes](#)
- [Optimizer Notes](#)
- [Performance Schema Notes](#)
- [Pluggable Authentication](#)
- [SQL Syntax Notes](#)

- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

## Account Management Notes

- The database cache was not flushed properly following execution of `DROP USER`. (Bug #37132323)
- This release adds the `CREATE_SPATIAL_REFERENCE_SYSTEM` privilege, which allows the user to execute any of the statements `CREATE SPATIAL REFERENCE SYSTEM`, `CREATE OR REPLACE SPATIAL REFERENCE SYSTEM`, and `DROP SPATIAL REFERENCE SYSTEM`. Use of the `SUPER` privilege for this purpose should now be considered deprecated.

For more information, see [Privileges Provided by MySQL](#). (Bug #37046126, WL #16547)

- Failed password validation was not always handled correctly. (Bug #37041439)

## C API Notes

- Process memory usage grew when the `libmysqlclient` API user tried to cache and reuse a prepared statement, preparing it once and then calling either of `mysql_stmt_bind_param()` or `mysql_stmt_bind_named_param()` followed by `mysql_stmt_execute()`, repeatedly without calling the matching `mysql_stmt_close()`, or calling it in the distant future (on application exit, for example).

We solve this by introducing a separate `MEM_ROOT` object for storing the bind parameters array, which object can be cleared (deallocating the memory) on each call to `mysql_stmt_bind_param()` or `mysql_stmt_bind_named_param()`. (Bug #37202066)

- The Doxygen comments for the `OK_Packet` did not cover all possible cases; this has been updated so that it now accurately reflects the data actually sent by the server. (Bug #35630063)

References: See also: Bug #35358417, Bug #36374232.

## 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:** Use `OBJECT` library in place of `routing_guidelines` when compiling with Xcode. (Bug #37350937)

- **macOS:** Removed obsolete CMake code from MacOS builds. (Bug #37258036)
- **macOS:** It is now possible to build MySQL using the Homebrew version of Clang. (Bug #37256912)
- **macOS:** Removed warnings of the form `ld: warning: ignoring duplicate libraries` and warnings specific to `xcodebuild`. (Bug #37065301)
- **Microsoft Windows:** Windows builds are now configured with `-DWITH_ASAN`. (Bug #37309813)
- **Microsoft Windows:** Disabled windows compiler warnings C26445 and C26821 in `cmake/msvc_cppcheck.cmake`. Both of these relate to MSVC substitution of `gsl::span` for `std::span`, which is not used for MySQL. (Bug #37158156)
- **Microsoft Windows:** Removed compiler warnings issued when building MySQL with Visual Studio 2022. (Bug #113870, Bug #36256477)
- **Microsoft Windows:** The CMake option `-DWITH_SASL` is not supported for building MySQL on Windows. Attempting to use this option when doing so now causes the build to terminate with an error. (Bug #113558, Bug #36155972)
- **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)
- 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 files `stream_cipher.h` and `stream_cipher.cc`, used by `mysys/`, were located in `sql/` but did not depend on any other server code in this directory; these files have now been moved into `mysys/`. (Bug #37257736)
- Removed or fixed issues in a number of copyright header files. (Bug #37238155)
- Removed a `maybe-uninitialized` error found in `sql/item.cc` when building MySQL with GCC 14. (Bug #37157201)
- Disabled several `clang-tidy` checks which generated unnecessary warnings when compiling MySQL. (Bug #37129808)
- The CMake code for building `component_mle` assumed that GraalVM binaries were available, either under `/usr/global/share`, or in a location provided on the command line. Now, when neither of these conditions hold, it is possible to download GraalVM from a remote server. (Bug #37121798)
- The version of `libedit` used to compile MySQL was upgraded to 20240808-3.1. (Bug #37101293)
- The server could not be built on Ubuntu 22.04 using Clang 13. (Bug #37075154)
- Removed an error in `mysql_prepare_create_table()` (in the file `sql/sql_table.cc`) found when compiling MySQL with XCode 16. (Bug #37068527)
- Some plugins failed to load when MySQL was built with `-FTLS-MODEL=initial-exec`. Fixed by removing this compiler flag, and falling back to the default (`global-dynamic`) instead. (Bug #37017999)

- MySQL could not be compiled using Clang 19. (Bug #37014761)
- `#include <chrono>` was missing from `plugin/group_replication/libmysqlgcs/src/bindings/xcom/xcom/task.cc` even though `std::chrono::duration_cast()` was referenced in this file. (Bug #116779, Bug #37329617)
- 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

- **Group Replication:** This release includes the Group Replication resource Manager component for MySQL Enterprise Edition, which monitors applier channel lag, recovery channel lag, and system resource usage on each secondary, and ejects any member whose lag or memory usage exceeds a user-configurable limit. Expelled servers may attempt to rejoin the group, provided that `group_replication_autorejoin_tries` is not set to 0.

The operator can determine upper bounds for applier channel lag, recovery channel lag, and memory usage by setting the system variables `group_replication_resource_manager.applier_channel_lag`, `group_replication_resource_manager.recovery_channel_lag`, and `group_replication_resource_manager.memory_used_limit`, respectively.

The Group Replication Resource Manager component checks the status of each secondary every 5 seconds, and updates a number of status variables which it provides, showing current channel lag and memory usage, as well as how many times a given threshold has been hit, and if the secondary has ever been expelled from the group, when this last occurred.

See [Group Replication Resource Manager Component](#), for more information. (WL #14940)

- For the MLE component, the `mysql_option.option_usage` table's `USAGE_DATA` column showed the `used` value as a string rather than as a boolean value.

For more information, see [Option Tracker Tables](#). (Bug #37122749)

- The entries present in the `mysql_option.option_usage` table depended on whether the Option Tracker was installed before or after the Audit Log.

See [Option Tracker Component](#), and [MySQL Enterprise Audit Plugin \(Deprecated\)](#), for more information. (Bug #37037438)

- Installing the Option Tracker component in one session while executing `UNINSTALL COMPONENT` in a different session caused the server to hang.

See [Option Tracker Component](#), for more information. (Bug #36991399)

- Errors in configuration files for keyring components were not properly logged. (Bug #36982002)
- `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.

- This release adds a Connection Control component (`component_connection_control`) intended to replace the Connection Control plugins (see [Connection Control Plugins](#)). A single component takes the place of both plugins, which are now deprecated and subject to removal in a future version of MySQL.

One component replaces both of the deprecated Connection Control plugins; installation of the component requires a single `INSTALL COMPONENT` statement. See [Connection Control Component Installation](#).

The system and status variables associated with the plugin are also deprecated; see [Connection Control Component Configuration](#) for information about the variables supported by the Connection Control component in place of the plugin variables.

In addition, the Information Schema `CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS` table used by the plugins is now deprecated; the component keeps counts of failed connection attempts in a Performance Schema table `connection_control_failed_login_attempts`.

See [The Connection Control Component](#), for more information. (WL #16573)

## Configuration Notes

- **Microsoft Windows:** On Windows, MySQL Configurator no longer executes the deprecated `FLUSH PRIVILEGES` statement. (Bug #37170330)

## Deprecation and Removal Notes

- The `FLUSH PRIVILEGES` statement is now deprecated, and causes a warning when issued. You should expect this statement to be removed in a future MySQL release.

The following constructs are also deprecated, and now cause a warning when used:

- The `FLUSH_PRIVILEGES` privilege, and granting of this privilege
- `mysqladmin flush-privileges`
- `mysqladmin reload`.

The features listed here do not cause any warnings but should be considered deprecated:

- Flushing of privileges by `SIGHUP`
- Flushing of the `caching_sha2` cache by `FLUSH PRIVILEGES`
- Flushing of privileges by `mysqladmin refresh`

For more information, see [FLUSH Statement](#), [mysqladmin — A MySQL Server Administration Program](#), and [Privileges Provided by MySQL](#). (WL #16567)

- The Version Tokens plugin is now deprecated, and subject to removal in a future MySQL release. The following related features are also now deprecated, and raise deprecation warnings whenever they are invoked:
  - The functions `version_tokens_delete()`, `version_tokens_edit()`, `version_tokens_lock_exclusive()`, `version_tokens_lock_shared()`, `version_tokens_set()`, `version_tokens_show()`, and `version_tokens_unlock()`
  - The `VERSION_TOKEN_ADMIN` privilege
  - The `version_tokens_session` and `version_tokens_session_number` server system variables

Attempting to install the `version_tokens` plugin—or to start the server when this plugin is installed—also causes a deprecation warning to be issued. (WL #16571)

## Doxygen Notes

- The payload length for `COM_QUERY` packets was calculated incorrectly in one of the MySQL packet examples. The payload length is now computed accurately, based on the actual length of the query.

Our thanks to Konno Satoshi for the contribution. (Bug #116339, Bug #37161043)

## Firewall Notes

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

## Installation Notes

- When upgrading from MySQL 5.7 to a later MySQL release series, the system-created `mysql.sys` and `mysql.session` accounts are now modified to use the `caching_sha2_password` authentication plugin instead of the `mysql_native_password` plugin, which is deprecated in MySQL 8.0, and removed in MySQL 9.0. (Bug #36608636)

## JavaScript Programs

- The MySQL `ENUM` and `SET` types are now supported for arguments of JavaScript stored routines. For more detailed information, including rules for conversion between these MySQL types and JavaScript types, see [Conversion to and from MySQL ENUM and SET](#). (WL #16599)
- JavaScript programs supported by the MLE Component, available as part of MySQL Enterprise Edition, now supports access from JavaScript routines to user-defined functions, procedures, and variables. MySQL stored functions and procedures can be accessed using the `Schema` methods `getFunction()` and `getProcedure()` which return `Function` objects which can be used with arguments to invoke the routines. `OUT` and `INOUT` parameters of a stored procedure use placeholders created with `mysql.arg()`. For additional information and examples, see [Stored Routine API](#).

The MLE component also now provides a JavaScript MySQL transaction API which performs the actions of most MySQL transactional SQL statements, such as `START TRANSACTION`, `COMMIT`, `ROLLBACK`, and `SET AUTOCOMMIT`. Support for savepoints is also included. Support for an `SqlError` object is included. For more information, see [JavaScript Transaction API](#), and [SqlError Object](#).

MySQL user variables can be accessed directly as properties of the JavaScript global `Session` object. For example, a user variable named `myvar` can be read or set as `session.myvar`. See [Accessing Session Variables from JavaScript](#), for more information and examples.

This release also adds support for direct access to the MySQL built-in functions `RAND()`, `SLEEP()`, `UUID()`, and `IS_UUID()`, as, respectively, `rand()`, `sleep()`, `uuid()`, and `isUUID()`. See [MySQL Functions](#).

For further information about JavaScript programs and the MLE Component, see [JavaScript Stored Programs](#), and [Multilingual Engine Component \(MLE\)](#). (WL #16585)

- The MLE component, available as part of MySQL Enterprise Edition, now supports reusable JavaScript libraries containing functions which can be called from other JavaScript stored programs. JavaScript libraries can be managed using the `CREATE LIBRARY` and `DROP LIBRARY` SQL statements added in

this release; they can be included in other stored JavaScript programs with the `USING` clause added to `CREATE FUNCTION` and `CREATE PROCEDURE`; `USING` supports a list of one or more library names.

`CREATE LIBRARY` creates a new JavaScript library in a given database. Library code is parsed and checked for validity at creation time, and is rejected if it contains errors. `DROP LIBRARY` drops a JavaScript library. Library functions can be referred to in other JavaScript stored programs using `library_name.function_name` notation. Libraries can be aliased with `USING` when including them in JavaScript programs.

You can obtain the code from a library using the `SHOW CREATE LIBRARY` statement, also new in this release. Two new related Information Schema tables have also been added: The `LIBRARIES` table provides information about JavaScript libraries, and the `ROUTINE_LIBRARIES` provides information about stored routines which include JavaScript libraries.

Counts of library DDL and `SHOW CREATE LIBRARY` statements which have been issued on the server are available as the status variables `Com_create_library`, `Com_drop_library`, and `Com_show_create_library`.

For more information and examples, see [Using JavaScript Libraries](#), as well as [Multilingual Engine Component \(MLE\)](#). (WL #16359, WL #16360, WL #16362, WL #16555)

## MySQL Enterprise Notes

- **Replication:** The `rnd_pos()` function of the handler interface for replication applier metrics tables was implemented wrongly; in cases where this function was used, the tables appeared to be missing one row.

For more information, see [Replication Applier Metrics Component](#). (Bug #37076428, Bug #37132660)

- **Replication:** For most wait operations, we note the time the wait started and the time it ends, and add their difference to the total waiting time, but this was not the case when tracking time spent waiting for the commit order, where we used a function that woke up every second and caused the wait time to be incremented by 1. This 1-second precision was too low for most practical purposes and caused unnecessary complication of the code.

To fix these issues, we now track waiting time in the same way for commit order waits as we do for other waits.

For more information, see [Replication Applier Metrics Component](#). (Bug #37053708)

- **Replication:** The MySQL Option Tracker component can now provide usage information about the binary log, group replication, and use of the server as a replica whenever these features are enabled.

For more information, see [Option Tracker Supported Components](#). (WL #16529)

## Optimizer Notes

- **JSON:** Added lookup references to iterator-based `EXPLAIN FORMAT=JSON` for index lookups. The `lookup_references` field in the JSON v2 `EXPLAIN` format corresponds to the `ref` field in the JSON v1 `EXPLAIN` format. (Bug #37126176)
- Some hash joins spent an unreasonably high time in `pack_rows::RequestRowId()`, even though they did not use row IDs.

Fixed by skipping the loop over `TableCollection::tables()` in `RequestRowId()` in the case where there are no tables from which to request row IDs. (Bug #37243461)

- [EXPLAIN FORMAT=TREE](#) now shows the clustered primary key scan for RowID-Ordered Retrieval (ROR) intersection plans. (Bug #37199800)
- 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)

- [PERFORMANCE\\_SCHEMA.PROCESSLIST](#) filtered foreground threads which did not have a user name. As of this release, [system\\_user](#) is assigned to foreground threads without a user name. (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)

## Pluggable Authentication

- The following issues relating to the [AUTHENTICATION\\_POLICY\\_ADMIN](#) privilege have now been resolved:
  - For a user having the privilege, it was not possible to create a user for whom a default authentication plugin was omitted, because the defaults specified in the global value of [authentication\\_policy](#) were included at creation time whether any were specified or not.

We fix this by ignoring the global authentication policy when the user performing account creation has the [AUTHENTICATION\\_POLICY\\_ADMIN](#) privilege.

- For a user not having the privilege, an attempt to create a user without specifying any authentication factors was rejected with an error, because mandatory factors in the default global authentication policy were not included when they were not specified.

We fix this by including any mandatory factors from the global authentication policy when the user performing the account creation does not have the [AUTHENTICATION\\_POLICY\\_ADMIN](#) privilege

For more information, see the description of the [authentication\\_policy](#) system variable, as well as [Privileges Provided by MySQL](#). (Bug #37027739)

- The [authentication\\_openid\\_connect](#) plugin is now registered with the Option Tracker component.

For more information, see [Option Tracker Component](#), and [OpenID Connect Pluggable Authentication](#). (Bug #116045, Bug #37041216)

## SQL Syntax Notes

- **Important Change:** The [BINLOG](#) keyword can no longer be used as an unquoted label name in MySQL stored programs. Prior to upgrading to this release, you should update any affected applications accordingly.

For more information, see [Keywords and Reserved Words](#). (Bug #22574003)

## Functionality Added or Changed

- **InnoDB:** During recovery, IBUF merges were disabled for all recovery batches except for the last batch, but is now disabled for all batches. This change prevents the addition of new redo log entries during recovery, as they led to deadlock and recovery performance degradation with reads and writes from the additional IBUF pages. (Bug #33134355, WL #15372)
- **Microsoft Windows:** On Windows, MySQL Configurator has added a command-line interface for configuring a MySQL Server installation. See [MySQL Configurator CLI](#), for more information.

A known limitation of this release: the new CLI only supports the `configure` action. (WL #16564)

- Binary packages that include `curl` rather than linking to the system `curl` library have been upgraded to use `curl` 8.11.1. (Bug #37389565)
- Refactored code to take advantage of OpenSSL 3.x functionality to improve performance for cryptographic operations, such as those performed by the MySQL `MD5()` and `SHA2()` functions. (Bug #116939, Bug #37185170, WL #16504)
- Added a source code static analysis tool. For usage details, see [scripts/static\\_analysis.md](#) in the source code or execute `python3 ./scripts/static_analysis.py --help` for additional help. Use this to check single commits or the entire source code repository. (WL #16329)
- Converted the caching SHA-2 authentication plugin to use the event component API instead of the audit log plugin API. This also removes the `sha2_cache_cleaner` audit plugin. (WL #16572)

## 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:** Improved the simulated asynchronous I/O (AIO) handler performance for high volume situations. (Bug #37366607)
- **InnoDB:** Improved asynchronous I/O (AIO) page cleaner thread management performance. (Bug #37359213)
- **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:** Improved `can_older_trx_be_still_active()` so that it no longer uses `trx_sys->rw_trx_list`. (Bug #36729529)
- **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:** Removed a memory leak from the `mysqladmin` client. (Bug #36537941)
- **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:** 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 `DELETE` operation on a table with a self referential foreign key and full-text index could have triggered an assertion. (Bug #36234681)
- **InnoDB:** Removed code that handled obsolete redo log formats. (Bug #35020216)
- **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:** If binary logging was disabled, it was possible for full-text search queries to be incorrect after a server crash.

Our thanks to Yin Peng and the Tencent team for the contribution. (Bug #116212, Bug #37095383)

- **InnoDB:** Removed a duplicate declaration of the `dd_set_autoinc()` method. (Bug #116175, Bug #37089340)
- **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 REPLICATION`, 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 REPLICATION` followed by `START REPLICATION`) `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.
- Improved an existing fix made in MySQL 9.1.0 for a condition which allowed queries with unknown columns in scope. (Bug #37341055)

References: This issue is a regression of: Bug #35058815.

- Calling `mle_set_session_state()` with a type argument that was not a JSON string caused the component to exit. (Bug #37334566)
- A subquery which was marked for materialization during resolution was not optimized when an impossible condition was detected else where in the query. This created problems when optimizer was trying to detect subqueries that needed to be materialized. Since the subquery was not optimized, trying to access the query block's join pointer resulted in a server exit.

We solve this by checking to see whether the subquery was optimized before finalizing the materialization transform for it. (Bug #37321762, Bug #37847157)

- Certain nested `SELECT` statements produced `ER_DUP_KEY` on a temporary table. (Bug #37309915)

References: This issue is a regression of: Bug #115597, Bug #36846564.

- A query having a subquery which did not use any tables and which had a nonzero `OFFSET` clause did not return the correct result. For example, `SELECT (SELECT 1 LIMIT 1 OFFSET 10)` returned `1` instead of the expected `NULL`. (Bug #37293822)
- 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)

- Added missing checks for `NULL` input arguments in `mle::validate()` and `validate_session_options()`. (Bug #37267887)
- For hash joins, time and resources were spent in unnecessary construction and destruction of `HashJoinChunk` objects that were never used, even in cases where a hash join could not be performed completely in memory and subsequently spilled to disk. (Bug #37235166)
- Adding an event to the `sys` schema during an upgrade led to an unplanned shutdown of the server. (Bug #37162611)
- 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)
- When evaluating an expression such as `value BETWEEN 'a' AND 'b'`, the server now checks the string literals and ensures that they use the same character set and collation as the `value` to be compared with them. (Bug #37086818)
- Removed an unnecessary assertion in `sql/item_func.cc`. (Bug #37083848)

References: See also: Bug #29467577.

- Removed a memory leak found in `option_tracker/udfs.cc`. (Bug #37075241)
- 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.

- Added a missing check for an empty pointer to `walkAndReplace()`, in `sql/sql_resolver.cc`. (Bug #36987582)

References: This issue is a regression of: Bug #112557, Bug #35855294.

- MySQL allowed outer references in window `PARTITION BY` and `ORDER BY` expressions, which was not in accordance with the SQL standard. (Bug #36921175, Bug #37847161)
- 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)
- Transformations of some scalar subqueries to derived tables were not always performed correctly. (Bug #36902116)
- 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)
- `MyISAM` block length calculations were not always performed correctly. (Bug #36347992)
- Removed a potential race condition. (Bug #35981769)

References: See also: Bug #36608160.

- The `mysqldump --column-statistics` option is now disabled by default. (Bug #35209008)
- 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)
- Setting `explain_json_format_version` to 2 now has the following effects on the output of `EXPLAIN FORMAT=JSON`:
  - The output includes the JSON schema version in `major.minor` format; this is always displayed as `2.0` in MySQL 9.2.
  - Only query attributes are stored directly in the top-level object; in version 1 output, this object contains both query attributes and iterator attributes.

The output of `EXPLAIN FORMAT=JSON` when `explain_json_format_version = 1` remains completely unchanged in this release. (Bug #116915, Bug #37372130, Bug #35239659)

- The `mysqlslap` utility did not disable SSL when the `--ssl-mode=disable` option was passed in. (Bug #116844, Bug #37353662)
- When taking the `UNION` of a `YEAR` column and a `BOOL (TINYINT)` column, the data type of the result was `TINYINT`, due to a flaw in the internal `field_types_merge_rules` array. We fix this by adjusting the result data type in such cases to `SMALLINT`. (Bug #116415, Bug #37192491)
- 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)

- A query containing a common table expression produced different results on the second and subsequent executions in some cases when used in a prepared statement or stored procedure. (Bug #116140, Bug #37074489)
- When the server `sql_mode` included `ANSI_QUOTES`, `SHOW GRANTS` quoted stored function and stored procedure names in backticks (```), while using double quotes (`"`) with the names of other objects. Now double quotes are always used in such cases. (Bug #115953, Bug #37003502)
- Window functions having `BIT` values as arguments did not return `BIT`. (Bug #115597, Bug #36846564)
- The `mysqlslap` utility became unresponsive if `--auto-generate-sql-execute-number` and `--concurrency` were specified without `--auto-generate-sql` or `--query`. (Bug #113215, Bug #36048754)
- 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)
- The `mysqldump --compact` option now disables `--tz-utc`.  
  
Previously, with `--tz-utc` enabled (the default), `--compact` executed `SET TIME_ZONE='+00:00'` before reading data but did not write this statement to the dump file. This caused an inconsistency, since data was extracted using UTC but readers of the dump did not know that the data used this time zone. (Bug #58491, Bug #11765514)
- When comparing internal representations of `VECTOR` columns, their length properties were not considered.

## Changes in MySQL 9.1.0 (2024-10-15, Innovation Release)

- [Atomic DDL Notes](#)
- [Audit Log Notes](#)
- [Authentication Notes](#)
- [C API Notes](#)
- [Compilation Notes](#)
- [Component Notes](#)
- [Configuration Notes](#)
- [Deprecation and Removal Notes](#)

- [SQL Function and Operator Notes](#)
- [JavaScript Programs](#)
- [Keyring Notes](#)
- [MySQL Enterprise Notes](#)
- [Performance Schema Notes](#)
- [Pluggable Authentication](#)
- [SQL Syntax Notes](#)
- [Thread Pool Notes](#)
- [Trigger Notes](#)
- [Vector Data Type](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

## Atomic DDL Notes

- Previously, `CREATE DATABASE` and `DROP DATABASE` were atomic but were not fully crash-safe. This could lead to the following issues:
  - When `CREATE DATABASE` failed after the database directory was created but before the operation was actually committed, the data dictionary had no record of the database's existence, but the database directory was still present in the file system, necessitating manual cleanup.
  - When removal of the database directory as part of `DROP DATABASE` was unsuccessful due to a file system error or an unexpected shutdown, the transaction containing `DROP DATABASE` was not always rolled back; manual intervention was required in such cases to rectify the problem.

These statements are now fully atomic and crash-safe, as long as all tables in the affected database use a storage engine (such as `InnoDB`) that supports atomic data definition statements.

For more information, see [Atomic Data Definition Statement Support](#). (WL #16232)

## Audit Log Notes

- The audit log treated a user whose name did not begin with an alphanumeric character (such as `'$foo'`) as invalid, despite the fact that the MySQL server allows such users to be created.

The same issue also affected MySQL Enterprise Firewall. (Bug #36778917)

## Authentication Notes

- **Microsoft Windows:** MySQL `webauthn` plugins now support Windows Hello authentication with Windows 11 and later using the MySQL `authentication_webauthn` plugin; the Windows Hello passkey store is now accepted as a software “device” (in addition to supported hardware devices). The server plugin is available with MySQL Enterprise Edition only; the client plugin is available with MySQL Enterprise Edition and MySQL Community Edition.

It is possible both to generate and to authenticate against passkeys stored in the Windows OS passkey store using Windows Hello, and is otherwise compatible with previous MySQL releases.

A new `mysql` client option `--plugin-authentication-webauthn-device=#` can be used to set the device number plugin option for the `webauthn` client plugin when multiple devices are available. If no such option is specified the first one (0) is used by default. The client raises an error if the device specified does not exist.

Only keys stored in the Windows passkey store are supported by this feature. Other possible backends are not supported with Windows Hello.

Stored passkey deletion from the windows passkey store must be accomplished using the tools provided by the system; it is not possible using the `mysql` client to delete a passkey.

For more information, see [WebAuthn Pluggable Authentication](#). (WL #16293)

- Added OpenID Connect support for MySQL Enterprise Edition using the `authentication_openid_connect` server-side authentication plugin.

For additional details, see [OpenID Connect Pluggable Authentication](#). (WL #16269)

## 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)
- Pulling in the most recent CMake packages for gRPC and Protobuf caused the builds for these libraries to fail. Fixed by rewriting the package lookup to work for both older and newer CMake packages. (Bug #36905657)
- 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)
- Added CMake code for building and linking with the system Tesseract OCR library. (Bug #36872432)
- Replaced the custom CMake function `MY_TARGET_LINK_OPTIONS` with the builtin function `TARGET_LINK_OPTIONS`. (Bug #36850490)
- The `-DWITHOUT_SERVER` option for CMake did not work on Enterprise Linux 7. (Bug #36824515)

- When compiling MySQL, we no longer check for Fedora versions 28 or 34, since these have reached end of life and are no longer supported. (Bug #36819263)
- Linking with `gold` did not work on Enterprise Linux 9. (Bug #36818902)

References: See also: Bug #34099162.

- Added the `DISABLE_PERFSHEMA` build option. When enabled, this has the effect of setting all `DISABLE_PSI_*` options to `ON`. The default for `DISABLE_PERFSHEMA` is `OFF`. (Bug #36778698)
- The version of `libfido` used with MySQL was upgraded to 1.15.0. (Bug #36752604)
- The RE2 regular expression library is now bundled with the MySQL sources. (Bug #36729026)
- The bundled version of `opentelemetry-cpp` was upgraded to version 1.15.0. (Bug #36708755)
- The version of `clang-format` used for the MySQL codebase was upgraded from 10 to 15. (Bug #36500268)
- `SQL_I_list` uses a member field `next` to track the pointer field to the next object, which is initialized as the address of the member field `first`; this is valid only with the list object constructed with regular constructors. The default assignment operator used memberwise assignment, which is not correct for an empty list.

This is fixed by supplying a customized move constructor and operator `=` function. In addition, we also change the implementation of `save_and_clear()` to use `std::move()` rather than assignment.

Our thanks to Xingyu Yang and the Tencent team for the contribution. (Bug #115712, Bug #36891161)

- 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)

## Component Notes

- **Group Replication:** The component `component_group_replication_flow_control_stats`, which provides statistics about transactions throttled by Group Replication flow control, is implemented in this release as part of MySQL Enterprise Edition. Installation of this component requires that the Group Replication plugin already be installed (see [Configuring an Instance for Group Replication](#)).

Flow control throttling statistics are made available as the global status variables `Gr_flow_control_throttle_active_count`, `Gr_flow_control_throttle_count`, `Gr_flow_control_throttle_last_throttle_timestamp`, and `Gr_flow_control_throttle_time_sum`.

These variables can be observed in `performance_schema.global_status` and the output of `SHOW GLOBAL STATUS`; their values are specific to each group member.

See [Group Replication Flow Control Statistics Component](#), for more information. (WL #16239)

- The `SHA3-224` algorithm was used when `SHA3-256` was specified as the encryption method. (Bug #36980306)
- The header file `mysql/plugin.h` was included with some logging components, although it is not actually needed by any component. (Bug #34772742)

- The Option Tracker component ([component\\_option\\_tracker](#)) added in this release implements two tables that supply information about options (that is, named features) present in the MySQL server as well as those in components and plugins.
- The read-only [performance\\_schema.mysql\\_option](#) table contains information about all options installed on the server, including name, component name (or [MySQL Server](#)), and whether the option is currently enabled.
- The read/write [mysql\\_option.option\\_usage](#) table shows usage information for each option on the system in [JSON](#) format. If the server is part of a Group Replication cluster, its server ID and cluster ID are also shown in this table.

For a given option, the values of the [OPTION\\_NAME](#) column of the [mysql\\_option](#) table and the column having the same name in the [option\\_usage](#) table are the same; you can use these as the join columns in queries against both tables together, like this:

```
mysql> SELECT * FROM performance_schema.mysql_option o
-> JOIN mysql_option.option_usage u
-> ON o.OPTION_NAME=u.OPTION_NAME\G
***** 1. row *****
OPTION_NAME: Enterprise AUDIT
OPTION_ENABLED: TRUE
OPTION_CONTAINER: audit_log plugin
CLUSTER_ID:
SERVER_ID:
OPTION_NAME: Enterprise AUDIT
USAGE_DATA: {"used": true, "usedDate": "2024-10-16T09:14:41Z"}
***** 2. row *****
OPTION_NAME: JavaScript Stored Program
OPTION_ENABLED: TRUE
OPTION_CONTAINER: component:mle
CLUSTER_ID:
SERVER_ID:
OPTION_NAME: JavaScript Stored Program
USAGE_DATA: {"used": false, "usedDate": "2024-10-15T13:40:03Z"}
```

Several status variables are added by the Option Tracker component; see [Option Tracker Status Variables](#). The component also provides functions for working with Option Tracker usage data, which are described in [Option Tracker Functions](#); you should be aware that the use of these functions requires the [OPTION\\_TRACKER\\_UPDATER](#) privilege, which is also implemented by the component.

The Option Tracker component is available as part of MySQL Enterprise Edition, a commercial offering.

For additional general information about the Option Tracker and the tables described previously, see [Option Tracker Component](#). For a list of components and plugins which support the Option Tracker, see [Option Tracker Supported Components](#). See also [The mysql\\_option Table](#). (WL #16217, WL #16403, WL #16503)

- This release introduces an Amazon Web Services keyring component ([component\\_keyring\\_aws](#)); this component supersedes the AWS keyring plugin ([keyring\\_aws](#)), which is now deprecated and thus subject to removal in a future version of MySQL.

For more information about the AWS keyring component, see [Using the component\\_keyring\\_aws AWS Keyring Component](#). For help with migration, see [Migration from AWS keyring plugin](#). For an overview comparing MySQL keyring components with keyring plugins, see [Keyring Components Versus Keyring Plugins](#). (WL #16348)

## Configuration Notes

- **Microsoft Windows:** On Windows, *MySQL Configurator* revert button functionality did not always revert to the original value. (Bug #36702176)
- **Microsoft Windows:** On Windows, the *MySQL Configurator* data directory file browsing mechanism required the `ProgramData` folder but can now function outside of that folder. (Bug #36702072)
- **Microsoft Windows:** On Windows, *MySQL Configurator* no longer performs unnecessary background operations after clicking **Next** if the selected options did not change. (Bug #36395605)
- **Microsoft Windows:** On Windows, *MySQL Configurator* described an empty password as strong rather than weak. (Bug #35533726)

## Deprecation and Removal Notes

- **NDB Cluster APIs:** Support for `Node.js`, which was deprecated in NDB Cluster 9.0, is removed in this release. (WL #16246)
- The `keyring_aws` plugin is now deprecated, and subject to removal in a future version of MySQL. Users should migrate to the AWS keyring component, which is introduced in this release.

For more information about the AWS keyring component, see [Using the component\\_keyring\\_aws AWS Keyring Component](#). For help with migration, see [Migration from AWS keyring plugin](#). (WL #16348)

## SQL Function and Operator Notes

- The output from `DATABASE()` was truncated when this function was used as part of a `UNION` query. (Bug #36871927)
- `SUM()` yielded a different result in some cases for the same value when `DISTINCT` was also used. For example:

```
mysql> SELECT SUM(b'1100'), SUM(DISTINCT b'1100');
+-----+-----+
| sum(b'1100') | sum(DISTINCT b'1100') |
+-----+-----+
|          12 |                9      |
+-----+-----+
```

This happened because, when using `DISTINCT`, a temporary table is used to hold the values so that in the end only the unique values are returned to the user. When creating such table while setting up `DISTINCT`, we did not take into consideration that the data type and length had been determined, and use these, recalculating them instead. Now we use the data type and length already determined. (Bug #115476, Bug #36796197)

## JavaScript Programs

- A JavaScript stored program always uses the `utf8mb4` character set, but the value shown in the `COLLATION_CONNECTION` column of `INFORMATION_SCHEMA.ROUTINES` for such a stored program was shown as `latin1`; now it is always set to `utf8mb4_0900_ai_ci` (the default collation for `utfmb4`) instead. This overrides any collation set by the user.

In most cases, this change does not have any visible effect on the result since the collation used for comparisons is determined based on the columns involved. An exception to this occurs when comparing literal values; `COLLATION_CONNECTION` is used in such cases. For example, if the user has previously set `collation_connection` to `latin1_danish_ci`, the result of `SELECT 'å' < 'ø'` when executed from within a JavaScript procedure may not match the user's expectations. A workaround is to

provide the collation explicitly, like this: `SELECT 'å' < 'ø' COLLATE utf8mb4_da_0900_ai_ci`. (Bug #36925906)

- When a prepared statement returned a blob value inside a stored routine, the blob value became invalid after the prepared statement was deallocated. (Bug #36902680)
- Employing a user variable in a DML statement within a stored procedure that used the Statement handle interface sometimes led to an unplanned server exit when any statement was executed following execution of such a stored procedure. To address this issue, we now reset the memory used to store the user variable and its value to the statement execution memory root after executing a sub-statement from the Statement handle interface. (Bug #36892945)
- `SqlResult` did not support iterators.

Now it is possible to iterate through a result set in the manner shown here:

```
let result = session.runSql("SELECT * FROM t")
for (let row of result) {
  console.log(row.c1+row.c3);
}
```

(Bug #36835486)

- The Multilingual Engine Component (MLE) now emits Telemetry metrics.

See [mysql.mle Metrics](#).

Additionally, SQL queries executed in stored JavaScript procedures are now observable in Performance Schema query digests and SQL statements executed in stored JavaScript procedures are now observable in `events_statements_history_long`. (Bug #36216014, WL #16354)

- The `VECTOR` type is now fully supported by JavaScript stored programs. Vectors can now be used as input arguments, output arguments, prepared statement `bind()` parameters, and return values.

For more information, see [JavaScript Stored Program Data Types and Argument Handling](#). (WL #16405)

## Keyring Notes

- Support for AES ECB, which is insecure, has been removed from all MySQL keyring-related components. (Bug #36749277)

## MySQL Enterprise Notes

- **Replication:** This release adds the MySQL Replication Applier Metrics component, which provides users with statistical information about replication formerly logged in the error log. The component adds two tables containing this information to the MySQL Performance Schema: `replication_applier_metrics` provides replication applier metrics for a given replication channel, and `replication_applier_progress_by_worker`: This table provides similar metrics for a specific worker.

This enhances observability of replication by gathering statistics from the entire replication pipeline, and unifying their presentation. As part of this work, some metrics which were not especially helpful have been replaced with more useful ones.

For more information about this component, see [Replication Applier Metrics Component](#). (WL #15620)

References: See also: Bug #32587480.

## Performance Schema Notes

- Fixed issues relating to the OTLP exporter. (Bug #36792180, Bug #36783070)
- It is now possible to enable or disable Telemetry meters from the command line or configuration file using the `performance-schema-meter` parameter, on server startup.

See [Server Meters](#). (Bug #36698082)

- Fixed an erroneous comment in `storage/perfschema/table_host.h`.

Our thanks to Sho Nakazono for the contribution. (Bug #115844, Bug #36954266)

- 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)

- MySQL's OpenTelemetry Logging enables you to export telemetry logs from your MySQL Server to OpenTelemetry backends for analysis. This feature is implemented in the following ways:
  - Telemetry Logging Component: (MySQL Enterprise Edition and MySQL HeatWave, only) collects instrumented log events from the server, formats it in OpenTelemetry's OTLP format, and exports the logs to the defined endpoint using the OpenTelemetry OTLP network protocol. The process listening at the endpoint can be an OpenTelemetry collector or any other OpenTelemetry-compatible backend. If you want to export to multiple backends, you must use an OpenTelemetry collector.

See [Installing OpenTelemetry Support](#).

- Telemetry Logging Interface: (MySQL Community Server, Enterprise Edition, and MySQL HeatWave) an API which enables you to define and integrate your own OpenTelemetry Logging components. This interface makes it possible to discover the available logging instrumentation, enable loggers, generate records, and extract the associated trace contexts.

The interface does not provide logging. You must use MySQL Enterprise Edition, MySQL HeatWave, or develop your own component to provide logging.

For information on the interface and sample component code, see the *Server telemetry logs service* sections of the [MySQL Server Doxygen Documentation](#).

(WL #15201, WL #15202)

## Pluggable Authentication

- `SET PERSIST authentication_ldap_simple_bind_root_pwd=password` did not actually save the password. (Bug #35478833)

## SQL Syntax Notes

- A query such as `SELECT test.dt.a FROM (SELECT 1 AS a) AS dt` uses an identifier chain of the form `db_name.tbl_name.col_name` to reference a column in a derived table, although this behavior contradicts the SQL standard. Such references were also accepted (also wrongly) in table expressions. We now disallow such column references in these cases. (Bug #36704815)

References: See also: Bug #36542023.

- Support is added in this release for an `IF NOT EXISTS` clause for use with the `CREATE VIEW` statement. `IF NOT EXISTS` has the following effects:
  - If the view named in the statement does not exist, it is created as usual, exactly as if the `IF NOT EXISTS` clause had not been used.
  - If the view already exists, the statement appears to succeed but does not change the view definition, and yields a warning, as shown here:

```
mysql> CREATE VIEW v1 AS SELECT c1, c3 FROM t1;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE VIEW v1 AS SELECT c2 FROM t1;
ERROR 1050 (42S01): Table 'v1' already exists
mysql> CREATE VIEW IF NOT EXISTS v1 AS SELECT c2 FROM t1;
Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> SHOW WARNINGS;
+-----+-----+-----+
| Level | Code | Message                               |
+-----+-----+-----+
| Note  | 1050 | Table 'v1' already exists           |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> SHOW CREATE VIEW v1\G
***** 1. row *****
          View: v1
          Create View: CREATE ALGORITHM=UNDEFINED DEFINER=`vuser`@`localhost` SQL
SECURITY DEFINER VIEW `v1` AS select `t1`.`c1` AS `c1`,`t1`.`c3` AS `c3` from `t1`
character_set_client: utf8mb4
collation_connection: utf8mb4_0900_ai_ci
1 row in set (0.00 sec)
```

`IF NOT EXISTS` is not compatible with `OR REPLACE`, and these two clauses cannot be used together in the same `CREATE VIEW` statement. Attempting to do so causes the statement to be rejected with a syntax error.

For more information, see [CREATE VIEW Statement](#). (WL #14774)

## Thread Pool Notes

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

## Trigger Notes

- **Performance:** Previously, for a table having one or more triggers, the triggers were fully parsed and loaded into memory every time the table was accessed. This often resulted in unexpectedly high resource usage and unreasonably long execution times in scenarios involving many tables, complex triggers, or both, in which no data was changed. It also led to unnecessary resource usage on read-only replicas.

We solve this problem by dividing the parsing and loading of triggers into two separate phases: In the first phase, we read only trigger metadata, since this can be stored once and shared between different instances of the same trigger. Actual parsing and execution of the trigger is done in the second phase, which is performed only for operations that modify table data; this saves resources because we no longer parse and cache triggers which are not used when executing, for example, a `SELECT` statement.

We implement this enhancement by tracking tables having triggers in the table cache separately from tables which do not have triggers. The maximum number of open tables which have fully loaded triggers and which are held in the table cache at any given time is determined by the value of the server system variable `table_open_cache_triggers`, which is added in this release. This variable is similar to the existing `table_open_cache` system variable, but controls an additional eviction mechanism, introduced in this release, specific to tables with fully loaded triggers. The default value for `table_open_cache_triggers` is the same as its maximum value (524288). Setting `table_open_cache_triggers` to a lower value activates the eviction logic specific to tables with fully loaded triggers. Leaving `table_open_cache_triggers` at its default values means that tables are not evicted from the cache based on whether they have any fully loaded triggers; this is the same behavior as in previous versions of MySQL.

A number of server status variables added in this release track cache usage for tables having triggers: `Table_open_cache_triggers_hits` provides the number of cache hits for open tables with triggers; `Table_open_cache_triggers_misses` shows the number of cache misses for open tables with triggers; `Table_open_cache_triggers_overflows` provides the number of cache overflows for open tables with triggers.

In addition, memory usage is also improved by replacing static buffers for trigger errors with variably-sized buffers which are allocated on demand.

Also as part of this work, `SHOW CREATE TRIGGER` (which reads but does not modify data) now displays the complete definition of all triggers, even in cases where trigger bodies cannot be parsed by the server.

For more information, see the descriptions of the server variables mentioned previously. For general information about the table cache, see [How MySQL Opens and Closes Tables](#).

Our thanks to Dmitry Lenev for the contribution. (Bug #44625, Bug #86821, Bug #11753220, Bug #26366333, WL #16455)

## Vector Data Type

- The `STRING_TO_VECTOR()` function now allows trailing whitespaces in string representations of `VECTOR` arguments. The parser now trims whitespace characters in such values after a number, before a square bracket, after a square bracket, or any combination of these. Some examples of previously unsupported values which are now permitted are shown here:

- `"[1 ,2]"`
- `"[1,2 ]"`
- `" [1,2]"`
- `"[1,2] "`

(Bug #37009633)

## Functionality Added or Changed

- **Important Change; InnoDB:** The default value of `innodb_log_writer_threads` is now `OFF` on systems with fewer than 32 logical processors. (WL #16396)
- **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)

- **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)
- **Group Replication:** Added new `INFO` level logging messages in the following cases:
  - When an action begins on all nodes
  - When sending of messages is blocked while waiting for the action to complete
  - When messages are received and the internal message counter is decremented

(Bug #32929873)

References: See also: Bug #32762677.

- For consistency of output, spaces have been added before and after the `=` sign in the condition shown for `lookup_condition` by `EXPLAIN FORMAT=JSON` when `explain_json_format_version=2`, as well as those shown for `operation` when using `EXPLAIN FORMAT=TREE`. (Bug #36825515)
- Output from `EXPLAIN FORMAT=TREE` now includes information about the semijoin strategy used, if any. (Bug #36773414)
- Added `"multi_range_read": true` to `EXPLAIN FORMAT=JSON` output, when `explain_json_format_version` is set to 2, whenever MRR is used for MRR and index range scan access paths. In addition, `(Multi-Range Read)` now appears in the `operation` field in the `EXPLAIN FORMAT=TREE` output for index range scans using MRR. (Bug #36614948)
- Added the `--system-command` option for the `mysql` client, which enables or disables the `system` client command.

This option is disabled by default, which means that the `system` command is rejected with an error. To enable it, use `--system-command=ON`. (Bug #36377685, WL #16482)

References: See also: Bug #36248967.

- The MySQL server now supports an alternative syntax for the `GROUP BY` clause `ROLLUP` modifier. Consider the following query:

```
SELECT
  IF(GROUPING(year), 'All years', year) AS year,
  IF(GROUPING(country), 'All countries', country) AS country,
  IF(GROUPING(product), 'All products', product) AS product,
  SUM(profit) AS profit
FROM sales
GROUP BY year, country, product WITH ROLLUP;
```

Using the alternative syntax, the same query can be written like this:

```
SELECT
  IF(GROUPING(year), 'All years', year) AS year,
  IF(GROUPING(country), 'All countries', country) AS country,
  IF(GROUPING(product), 'All products', product) AS product,
  SUM(profit) AS profit
FROM sales
```

```
GROUP BY ROLLUP (year, country, product);
```

Either version of the query produces the same result.

For more information, see [GROUP BY Modifiers](#), as well, as [SELECT Statement](#). (WL #15843)

- This release implements the following two status variables for tracking internal temporary table conversions from in-memory to disk:
  - `TempTable_count_hit_max_ram` counts the number of in-memory to disk table conversions due to reaching the `temptable_max_ram` limit with the `TempTable` storage engine. This is only a global status variable.
  - `Count_hit_tmp_table_size` counts the number of in-memory to disk conversions due to reaching the `tmp_table_size` limit with the `TempTable` storage engine, or from reaching the smaller value of `tmp_table_size` and `max_heap_table_size` with the `MEMORY` storage engine. This is both a global and session status variable.

(WL #16377)

- The maximum `TOTAL_ROW_VERSIONS` value changed from 64 to 255. This value is incremented by executing `ADD COLUMN` and `DROP COLUMN` operations using the `INSTANT ALGORITHM`. (WL #16369)
- Added the following server system and status variables for use in tracking memory usage by queries when `global_connection_memory_tracking` is enabled; the first two variables in the list shown here apply globally, and the second two to individual user connections:
  - `global_connection_memory_status_limit`: This system variable determines the maximum total amount of memory that can be consumed by all user connections to the server. When the total amount of memory used by all queries exceeds this amount, this causes `Count_hit_query_past_global_connection_memory_status_limit` to be incremented.
  - `Count_hit_query_past_global_connection_memory_status_limit`: This status variable tracks the number of times that queries using any connection have caused total memory consumption to exceed `global_connection_memory_limit`. This is incremented only when a query brings the total memory consumption from less than `global_connection_memory_limit` to greater than this amount.
  - `connection_memory_status_limit`: This system variable determines the maximum amount of memory that can be consumed by a given user connection before incrementing `Count_hit_query_past_connection_memory_status_limit`.
  - `Count_hit_query_past_connection_memory_status_limit`: This status variable tracks the number of times that queries using the current connection have caused total memory consumption to exceed `connection_memory_limit`. This is incremented, for a query executed within the connection, only when such a query brings the total memory consumption from less than `connection_memory_limit` to greater than this amount.

These variables are intended for testing configurations prior to putting them into production, and do not cause queries to be rejected as when exceeding `connection_memory_limit`. (WL #16097)

## 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:** Removed the `DEBUG_SYNC_C_IF_THD` macro in favor of `DEBUG_SYNC`. (Bug #36870544)
- **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:** Improved error handling for bulk load operations with tables containing a mix of `BLOB` and other data types. Previously, the error type was reported as an empty string. (Bug #36764710)
- **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:** The log index size calculation now accounts for column order changes. (Bug #36526369)

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

- **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:** 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 #36302624)

- **InnoDB:** Fixed a memory leak in the bulk loader. (Bug #35988311)
- **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:** 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 an 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)
- For a subquery which uses window functions in its `SELECT` list, an `EXISTS()` transformation cannot be used; this means that, for such a subquery, the transformation used is always materialization, but this is true only when the subquery uses one of `IN`, `ALL`, `ANY`, or `SOME`. The check for whether materialization is forced (`IsForcedMaterialization()`) for a subquery returned true when the query had an `IN` subquery whose `WHERE` clause used an `EXISTS` subquery containing window functions, due to the fact that the innermost subquery having window functions always returned its strategy as materialization, even though the `EXISTS` strategy had already been chosen for the `IN` subquery. As a result, some checks failed when this subquery was considered for materialization.

We fix this by adding a check for a quantified comparison predicate in `IsForcedMaterialization()`, so that the subquery with a window function is not looked into. (Bug #37008930)

- Removed a heap-use-after-free warning in `regex::Regex_facade::~~Regex_facade()`. (Bug #36867806)
- Some combinations of `EXCEPT` and `INTERSECT` led to an error in `Item::convert_charset()`. (Bug #36838373)
- 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)
- Eliminated unnecessary copying in `StoreLinkedImmutableStringFromTableBuffers()`, improving the performance of some hash joins by 5 to 10 percent. (Bug #36805260)
- 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 calculating `Last_query_cost`, the optimizer did not always take the costs of all subqueries into account. (Bug #36790906)
- 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)
- When the argument to `COUNT()` used as a window function was a `GROUP BY` expression and `WITH ROLLUP` was used, `COUNT()` returned `NULL` instead of `0`. (Bug #36750571)

- 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)
- Preparation of table value constructors did not track the number of hidden columns added for expressions in an `ORDER BY` clause. This could cause problems later in the resolution process, possibly leading to an unplanned server exit.

We fix this by counting the number of hidden items in table value constructors in the same way as this is done in other query blocks. (Bug #36560156)

References: This issue is a regression of: Bug #35785452.

- `DROP INDEX` with the addition of a `FULLTEXT` index in the same transaction sometimes led to an unplanned server exit. (Bug #36559642)
- Fixed an issue related to the handling of FTS indexes during an upgrade. (Bug #36526587)
- 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.

- If an expression involved in condition pushdown had one view reference which was also an outer reference and one local reference, then dependency consistency checks failed while cloning the condition.

We solve this by traversing the item tree prefix and postfix. While traversing the prefix, we set the `Item_ref` object to be used for setting dependency and context information for the underlying columns of the reference object; while traversing the postfix, we clear this object. (Bug #36368181)

- Some prepared statements were not reprepared correctly. (Bug #36267792, Bug #35712413)

References: This issue is a regression of: Bug #34929930.

- 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)
- `RelWithDeblInfo` 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)
- When converting a string to a `MYSQL_TIME` struct, we did not always verify that the day number was less than or equal to 31. (Bug #35884337, Bug #36633426)
- 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)
- Removed a memory leak from the `FEDERATED` storage engine that was discovered during testing of queries using index merge plans. (Bug #35362984)
- In some cases, table DDL in prepared statements was not handled correctly. (Bug #35221658)
- A statement which called a stored routine containing an aggregate function, whose result was stored in a variable local to the routine, raised an assert in `sql/sql_lex.h`. (Bug #35102083)
- In a `UNION`, it was sometimes possible for one of the `SELECT` statements involved to reference a column in an `ORDER BY` clause when this column was not among those accessed by that `SELECT`. (Bug #35058815)
- 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)
- 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)
- The XML function `ExtractValue()` did not handle arguments containing accented characters correctly. (Bug #117778, Bug #36236440)
- `NOT` worked correctly with a quantified comparison such as `< ALL`, `> ANY`, and so on, but `NOT NOT` did not. (Bug #115962, Bug #37004689)
- A `GROUP BY` query containing a `LATERAL` subquery raised a spurious `Duplicate column` error. (Bug #115696, Bug #36885503)
- 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)

- Added `gcc-toolset-12` to the EL8 and EL9 build specification file to allow building the MySQL 8.0 `mysql-community-libs-compat` RPM.

Our thanks to Simon Mudd for the contribution. (Bug #115484, Bug #36796229)

- Fixed a typo in the Debian package description.

Our thanks to Henning Pöttker for the contribution. (Bug #115363, Bug #36749142)

- `MYSQL_TYPE_JSON` was missing from the description of binary resultsets in the MySQL source code documentation.

Our thanks to Daniël van Eeden for the contribution. (Bug #115360, Bug #36740656)

- Certain complex `UNION` queries triggered an assertion in debug builds, and caused release builds to hang. (Bug #115346, Bug #36739383)

- A query such as `SELECT CASE WHEN 1 THEN NOW() ELSE x.x END FROM (SELECT NULL) x(x)` led an assert. This occurred because a generated column description used a temporal type with a fractional seconds precision of 31, which is an invalid precision value. The invalid precision came from type aggregation, where one of the underlying items to the `CASE` expression was a `NULL` expression, which has a decimal precision equal 31, meaning an undefined precision.

The problem is fixed by not aggregating types from expressions that yield `NULL`.

This issue did not occur with numeric and string arguments, since they cannot be aggregated directly into a temporal type.

This issue did not occur with MySQL 8.0 or with MySQL 8.4 release binaries. (Bug #115233, Bug #36705061)

- A query, which normally used an index range scan, used a less efficient index lookup when run as a prepared statement. (Bug #115227, Bug #36705030)
- A singly-nested derived table was wrongly merged rather than materialized. (Bug #115215, Bug #36695371)
- A `ROLLUP` query did not take account a summary `NULL`. Consider the following statements:

```
CREATE TABLE t(a INT);  
  
SELECT a, COUNT(*) FROM t GROUP BY a WITH ROLLUP;
```

According to the SQL standard, the `SELECT` statement should be equivalent to a `UNION` query which produces the result `(0, NULL)`. This is because, even with an empty result set, `ROLLUP` should give us a single grouping row with `NULL` entries for all `GROUP BY` expressions as well as a grand total for any aggregates, in this case 0 for `COUNT(*)`, but MySQL returned an empty set instead.

Now we return the grouping row in such cases. (Bug #114638, Bug #36514339)

- Incorrect results were sometimes obtained when using `DISTINCT` and `ORDER BY` with a derived table. (Bug #114589, Bug #36496160)
- A `CREATE TABLE` statement with a `CHECK` constraint that involved a `CAST` with an `AT TIME ZONE` clause failed with the confusing error `Missing time zone function`. This was because the text of the `CHECK` constraint was regenerated internally and subsequently misinterpreted.

We fix this by ensuring that the correct `CHECK` constraint text is stored in the data dictionary along with the rest of the `CREATE TABLE` definition. (Bug #114404, Bug #36423829)

- 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 join on subqueries which themselves used subqueries in their `HAVING` and `FROM` clauses led to an assert in `sql/item.h`.

The problem occurred when removing a scalar subquery with a `HAVING` clause after having decided that the subquery was part of a predicate that could be eliminated. The `HAVING` clause pointed to an aggregation object (`Item_sum_min`), indirectly accessed through an `Item_aggregate_ref` that was added by `Item::split_sum_func2()`. When removing the subquery, the reference count for the `Item_aggregate_ref` was decremented without ever having been incremented.

We fix this by incrementing the object's reference count when it is first created. (Bug #112615, Bug #35877063)

References: See also: Bug #111492, Bug #35517962.

- Queries using `CASE (WHEN TRUE AND EXISTS (subquery))` did not always return the same result as with `CASE (EXISTS (subquery))`. (Bug #112557, Bug #35855294)

References: This issue is a regression of: Bug #32644631, Bug #32802301.

- A character string literal value selected through a derived table and matched with a column with a different character set through a `UNION` operation sometimes raised `ER_CANT_AGGREGATE_2COLLATIONS (Illegal mix of collations...)`. (Bug #108627, Bug #34646522)

References: See also: Bug #36812010, Bug #36844420.

- 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 9.0.1 (2024-07-23, Innovation Release)

### 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 9.0.0 (2024-07-01, Innovation Release)



### 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 9.0.1 instead.

- [Audit Log Notes](#)
- [Authentication Notes](#)
- [C API Notes](#)
- [Character Set Support](#)
- [Compilation Notes](#)
- [Component Notes](#)
- [Configuration Notes](#)
- [Connection Management Notes](#)
- [Data Dictionary Notes](#)
- [Data Type Notes](#)
- [Deprecation and Removal Notes](#)
- [Event Scheduler Notes](#)
- [JavaScript Programs](#)
- [Keyring Notes](#)
- [Optimizer Notes](#)
- [Performance Schema Notes](#)
- [SQL Syntax Notes](#)
- [sys Schema Notes](#)
- [Thread Pool Notes](#)
- [Vector Data Type](#)
- [X Plugin Notes](#)
- [Functionality Added or Changed](#)
- [Bugs Fixed](#)

## Audit Log Notes

- The audit log plugin printed the warning `Plugin audit_log reported: 'Cannot process audit log file. File name timestamp value is missing or invalid: 'dbname''` when no audit log file was being processed. (Bug #36400810)
- With large files (200 MB or larger), executing `audit_log_read()` took an excessive amount of time to find the bookmark matching the one specified when invoking the function. (Bug #36281295)
- Fixed an issue related to `audit_log_read` performance. (Bug #36281295, Bug #38275868)
- 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)

- Improved the `audit_null` example plugin. (Bug #35820314)
- 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.

## Authentication Notes

- Made improvements to the authentication code. (Bug #35325870)
- Improved log messages to provide clear reasons for `Access denied` errors when using the `authentication_ldap_sasl` plugin without proxying. (Bug #35317691)
- MySQL LDAP SASL authentication, when used with the GSSAPI method to access an OpenLDAP server, was rejected with the MySQL server error `Plugin authentication_ldap_sasl reported: 'LDAP authentication failed or group retrieval failed: LDAP error: Invalid DN syntax'`, because OpenLDAP did not recognize the root DN used. (Bug #32631511)

## C API Notes

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

## Character Set Support

- When resolving a call to the `REPLACE()` function, the character set and collation of the function result are copied from the first argument. The remaining two arguments, if they are literal values, should be converted to this character set, but only the second argument was converted.

This fix ensures that the third argument is also converted to the first argument's character set and collation. (Bug #114769, Bug #36562972)

- The internal function `my_instr_mb()` assumed incorrectly assumption in several places that byte lengths for input strings could be used to short-cut certain decisions. In the case of multibyte character sets and collations, this cannot be done since, under some collation rules, characters with differing byte lengths can be considered equal. In addition, `Item_func_locate()` used an incorrect byte length.

Our thanks to Dirkjan Bussink for the contribution. (Bug #113933, Bug #36277823)

- The internal function `Item_func::eq()` erroneously treated the two expressions `CONVERT(a USING latin1)` and `CONVERT(a USING utf8mb4)` as being equal. (Bug #113506, Bug #116762, Bug #36137690, Bug #37323921)

## Compilation Notes

- **macOS:** CMake no longer tries to use the native `ctags` on MacOS, and now requires the Homebrew version of it to be installed on the system when building MySQL. (Bug #36590594)
- **macOS:** Enabled use of gRPC when building MySQL on MacOS. (Bug #36537726)
- **macOS:** The Xcode version of `zlib` was removed from the default list of system libraries to use when configuring with `-DWITH_SYSTEM_LIBS=ON`. (Bug #36537593)
- **Microsoft Windows:** The `BUILD_ALL` target did not work when compiling on Windows. (Bug #36424619)
- **Microsoft Windows:** Excessive RAM usage led to disk swapping when compiling MySQL on Windows using Ninja. (Bug #36399256)

- Interface libraries for `librpdserver_shared.so` were missing from the bundled `protobuf/grpc` on Ubuntu 24.04. (Bug #36678790)
- Maintainer mode is now disabled when building the debug version of the server for `.deb` packages. (Bug #36619757)
- Upgraded the bundled `googletest` and `googlemock` sources to version 1.14.0. (Bug #36562482)
- Added a missing dependency on `GenError`. (Bug #36551721)
- When compiling on Fedora 38, `grep -E` is now used in place of `egrep`. (Bug #36507549)
- The version of Boost used for compiling MySQL was upgraded from 1.84.0 to 1.85.0. For more information see the [Boost 1.85.0 Release Notes](#). (Bug #36495694)
- Binaries for Enterprise Linux 8 and 9 are now built using GCC 13. (Bug #36331855)
- Removed linker warnings raised when compiling code that used RapidJSON. (Bug #36322583)
- 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)
- Removed warnings raised in `sql/statement/ed_connection.cc` when building on Ubuntu 23.04. (Bug #114436, Bug #36428465)
- 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)

## Component Notes

- When the server was started with an incorrect `--basedir` option and no `--plugin-dir` option on UNIX platforms with a manifest and configuration file in place, the keyring component loaded but failed to unload.  
  
Now, when loading keyring components from the manifest, if `--basedir` is specified but `--plugin-dir` is not specified, we derive the plugin directory from the base directory; if neither of these is specified, we derive the plugin directory from the MySQL installation directory. The same handling also now applies to `--datadir`. (Bug #36398484)
- The values for component options set using the `--loose` prefix were not read when the component was installed. (Bug #28341329)

## Configuration Notes

- **Microsoft Windows:** On Windows, *MySQL Configurator* was updated to support in-place upgrades as per [Upgrade Paths](#). (Bug #36685422)
- **Microsoft Windows:** On Windows, clicking the **[X]** close button on a *MySQL Configurator* wizard's page now yields a confirmation popup if the wizard is busy executing an operation. (Bug #36671317)
- **Microsoft Windows:** On Windows, *MySQL Configurator* no longer defines a custom `server_type` variable in the generated MySQL Server configuration file. This information is now stored in the `configurator_settings.xml` file. (Bug #36670309)

- **Microsoft Windows:** On Windows, the [Removing Windows Firewall](#) step in *MySQL Configurator* would fail if the `my.ini` file was missing a `mysqlx_port` definition. (Bug #36666260)
- **Microsoft Windows:** On Windows, if *MySQL Configurator* failed to find a valid `my.ini` or `my.cnf` file from the *MySQL Server Installations* page, then clicking the **Browse** button disabled the **Next** button even when the selected file was valid. (Bug #36395569)
- **Microsoft Windows:** On Windows, *MySQL Configurator* now only shows the removal steps if the associated MySQL Server was previously configured. (Bug #36395417)
- **Microsoft Windows:** On Windows, *MySQL Configurator* now detects if the existing root user is using the `mysql_native_password` authentication plugin (removed in MySQL 9.0.0) and prompts to convert root to use the `caching_sha2_password` authentication plugin before performing a MySQL Server upgrade. (WL #16139)
- **Microsoft Windows:** For MSI installations on Windows, *MySQL Configurator* now automatically upgrades MySQL 8.4 LTS installations without user intervention. (WL #16274)

## 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)

## Data Type Notes

- When a string is converted to a numeric value, any non-numeric data trailing the numeric value should cause an error with strict mode and a warning with any other SQL mode, but in some cases, depending on the length and character set of the string, an invalid string did not raise any of the expected errors or warnings. (Bug #36457756)
- In some cases, casting a double to an integer value used rounding, and in others, with truncation, which led to inconsistent results. Now rounding up is used in all such cases. (Bug #114549, Bug #36481397)
- In some cases, `DECIMAL 0` was treated as less than a `FLOAT` value between 0 and -1. (Bug #114196, Bug #117093, Bug #36361165, Bug #37438582)

## Deprecation and Removal Notes

- **Replication:** MySQL 9.0.0 deprecates transactions which update both transactional tables and nontransactional or noncomposable tables. Such a transaction now causes a deprecation warning to be written to both the client and the error log. Only the `InnoDB` and `BLACKHOLE` storage engines are transactional and composable. This means that only the combinations of storage engines shown here do *not* raise the deprecation warning:
  - `InnoDB` and `BLACKHOLE`
  - `MyISAM` and `MERGE`
  - `performance_schema` and any other storage engine
  - `TempTable` and any other storage engine

**Note**

`NDBCLUSTER` is transactional but not composable.

For more information, see [Replication and Transactions](#). (WL #10495)

- The `mysql_native_password` authentication plugin, deprecated in MySQL 8.0, has been removed, and the server now rejects `mysql_native` authentication requests from older client programs which do not have `CLIENT_PLUGIN_AUTH` capability. For backward compatibility, `mysql_native_password` remains available on the client; the client-side built-in authentication plugin has been converted into a dynamically loadable plugin.

**Note**

In MySQL 8.0, the default MySQL authentication plugin was changed to `caching_sha2_password` (see [Caching SHA-2 Pluggable Authentication](#)).

These changes also entail the removal of the following server options and variables:

- The `--mysql-native-password` server option
- The `--mysql-native-password-proxy-users` server option
- The `default_authentication_plugin` server system variable

For more information, see [Authentication Plugins](#). (WL #15930)

- The `MIN_VALUE` and `MAX_VALUE` columns of the Performance Schema `variables_info` table are now deprecated, and subject to removal in a future version of MySQL. Instead, you should use the `MIN_VALUE` and `MAX_VALUE` columns of the `variables_metadata` table, which provide the same information. (WL #15585)

## Event Scheduler Notes

- **Important Change:** The following SQL statements relating to events may now be prepared:
  - `CREATE EVENT`
  - `ALTER EVENT`
  - `DROP EVENT`

Positional parameters (`?` placeholders) are not supported for these statements; you must assemble the text of the statement to be prepared from some combination of string literals, system variables, and user variables. See [PREPARE, EXECUTE, and DEALLOCATE PREPARE Statements](#), and [SQL Syntax Permitted in Prepared Statements](#), for more information. [CREATE EVENT Statement](#), provides a basic example. (Bug #109309, Bug #34875573, WL #16298)

## JavaScript Programs

- MySQL 9.0 Enterprise Edition now includes support for stored programs written in JavaScript, such as this simple example created using the `CREATE FUNCTION` statement and JavaScript code shown here:

```
CREATE FUNCTION gcd(a INT, b INT)
RETURNS INT
NO SQL
```

```

LANGUAGE JAVASCRIPT AS
$mle$
  let x = Math.abs(a)
  let y = Math.abs(b)
  while(y) {
    var t = y
    y = x % y
    x = t
  }
  return x
$mle$
;

```

[JavaScript Stored Program Creation and Management](#), describes creation and execution of JavaScript stored programs.

JavaScript language support includes both stored procedures and stored functions, and is provided by the Multilingual Engine Component (MLE). For more information about determining whether your distribution includes this component, and enabling it, see [Multilingual Engine Component \(MLE\)](#).

JavaScript language support in MySQL conforms to the [ECMAScript 2023 Specification](#), and uses strict mode by default. Strict mode cannot be disabled. This implementation includes all of the standard ECMAScript library objects such as [Object](#), [Function](#), [Math](#), [Date](#), and [String](#). `console.log()` and `console.error()` are also supported.

Most MySQL data types are supported for JavaScript stored program input and output arguments, as well as for return data types. Strings must use the `utf8mb4` character set. MySQL [BLOB](#) and [TEXT](#) types are supported, as are many MySQL temporal types. [JSON](#) is also supported. The [VECTOR](#) type is not supported by the MLE component or by JavaScript stored programs. For more information, see [JavaScript Stored Program Data Types and Argument Handling](#), and [JavaScript Stored Program Limitations and Restrictions](#).

Stored programs written in JavaScript support an SQL and result set API provided by the MLE component. See [JavaScript SQL API](#), and [Using the JavaScript SQL API](#), for more information.

The MLE component provides a number of session information and management functions including `mle_session_state()` and `mle_session_reset()`. You can also view a number of MLE status variables in the output of a statement similar to `SHOW STATUS LIKE 'mle%'`. See also [JavaScript Stored Programs—Session Information and Options](#).

For general information about JavaScript stored programs, see [JavaScript Stored Programs](#). (WL #15605, WL #16129, WL #16172, WL #16226, WL #16272, WL #16276)

## Keyring Notes

- Made a number of improvements in the keyring code. (Bug #36401550)

## Optimizer Notes

- **Important Change:** `ER_SUBQUERY_NO_1_ROW` has been removed from the list of errors which are ignored by statements which include the `IGNORE` keyword. This has been done for the following reasons:
  - Ignoring such errors sometimes led to insertion of NULL into non-nullable columns (for untransformed subqueries), or of no row at all (subqueries using `subquery_to_derived`).
  - When subqueries were transformed to join with derived tables, the behavior differed from that of untransformed queries.

Following an upgrade to this release, this change can make an `UPDATE`, `DELETE`, or `INSERT` statement which includes the `IGNORE` keyword raise errors if it contains a `SELECT` statement with a scalar subquery that produces more than one row.

For more information, see [The Effect of IGNORE on Statement Execution](#). (Bug #110961, Bug #35373406)

## Performance Schema Notes

- **Group Replication:** The following tables did not contain data on replication channels which did not have a configured hostname, such as Group Replication recovery channels:
  - `REPLICATION_CONNECTION_STATUS`
  - `REPLICATION_CONNECTION_CONFIGURATION`
  - `REPLICATION_APPLIER_CONFIGURATION`
  - `REPLICATION_APPLIER_STATUS`
  - `REPLICATION_APPLIER_STATUS_BY_COORDINATOR`
  - `REPLICATION_APPLIER_STATUS_BY_WORKER`

As of this release, these tables contain data for partially configured Group Replication channels. (Bug #36018242)

- 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)

- This release adds two tables to the MySQL Performance Schema, listed here:
  - The `variables_metadata` table provides general information about system variables. This information includes the name, scope, type, range (where applicable), and description of each system variable recognized by the MySQL server.

The `MIN_VALUE` and `MAX_VALUE` columns of this table are intended to replace the deprecated `MIN_VALUE` and `MAX_VALUE` columns of the `variables_info` table.

- The `global_variable_attributes` table provides information about attribute-value pairs assigned by the server to global system variables.

For more information, see [Performance Schema System Variable Tables](#). (WL #15855)

## SQL Syntax Notes

- **JSON:** You can now save the `JSON` output from `EXPLAIN ANALYZE` into a user variable using the syntax shown here:

```
EXPLAIN ANALYZE FORMAT=JSON INTO @variable select_stmt
```

The variable can be used subsequently as a `JSON` argument to any of MySQL's `JSON` functions (see [JSON Functions](#)). The `INTO` clause is supported only with `FORMAT=JSON`, which must be included

explicitly. This form of `EXPLAIN ANALYZE` also supports an optional `FOR SCHEMA` or `FOR DATABASE` clause preceding the `SELECT` statement being analyzed. Statements other than `SELECT` are not supported.



### Note

This feature is available only if the `explain_json_format_version` server system variable is set to 2; otherwise, attempting to make use of it raises `ER_EXPLAIN_ANALYZE_JSON_FORMAT_VERSION_NOT_SUPPORTED`.

For more information and examples, see [Obtaining Execution Plan Information](#). (WL #16216)

- MySQL now accepts and enforces inline foreign key specifications (these were previously accepted by the parser, but ignored). MySQL now also accepts implicit references to parent table primary key columns.

Consider the parent table `person` created by the following statement:

```
CREATE TABLE person (
  id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  name CHAR(60) NOT NULL
);
```

To create a table `shirt` having a foreign key `owner` on `person`, MySQL now accepts and handles correctly any of the `CREATE TABLE` statements shown here, according to the standard:

```
CREATE TABLE shirt (
  id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  style ENUM('tee', 'polo', 'dress') NOT NULL,
  color ENUM('red', 'blue', 'yellow', 'white', 'black') NOT NULL,
  owner SMALLINT UNSIGNED NOT NULL,

  FOREIGN KEY (owner) REFERENCES person (id)
);
```

```
CREATE TABLE shirt (
  id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  style ENUM('tee', 'polo', 'dress') NOT NULL,
  color ENUM('red', 'blue', 'yellow', 'white', 'black') NOT NULL,
  owner SMALLINT UNSIGNED NOT NULL REFERENCES person (id)
);
```

```
CREATE TABLE shirt (
  id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  style ENUM('tee', 'polo', 'dress') NOT NULL,
  color ENUM('red', 'blue', 'yellow', 'white', 'black') NOT NULL,
  owner SMALLINT UNSIGNED NOT NULL,

  FOREIGN KEY (owner) REFERENCES person
);
```

```
CREATE TABLE shirt (
  id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  style ENUM('tee', 'polo', 'dress') NOT NULL,
  color ENUM('red', 'blue', 'yellow', 'white', 'black') NOT NULL,
  owner SMALLINT UNSIGNED NOT NULL REFERENCES person
);
```

);

In previous versions of MySQL, only the first of the four statements just shown had the effect of creating a foreign key; the second was parsed, but the [REFERENCES](#) clause was ignored. The remaining two statements caused syntax errors.

For more information, see [FOREIGN KEY Constraints](#), as well as [FOREIGN KEY Constraint Differences](#). (Bug #4919, Bug #17943, Bug #102904, Bug #11744902, Bug #11745637, Bug #32613571, WL #16130, WL #16131)

## sys Schema Notes

- The performance of the [innodb\\_lock\\_waits](#) view is improved in this release. (Bug #36337708)

## 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)
- Previous refactoring incorrectly removed the connection locking performed when processing incoming connection requests, which led to a race condition between the thread adding new connections and the connection handler thread processing them. This appeared to cause a situation in which connection requests might be ignored and not processed, so that the connection attempt appeared to hang.

We fix this by taking the connection before processing the queue, and releasing it before waking or creating new threads. (Bug #36548687)

- The Information Schema tables [TP\\_THREAD\\_GROUP\\_STATE](#), [TP\\_THREAD\\_GROUP\\_STATS](#), and [TP\\_THREAD\\_STATE](#) were deprecated in MySQL 8.0.14. Accessing any of these tables now produces a warning; you should use the equivalent Performance Schema tables instead.

For more information, see [Performance Schema Thread Pool Tables](#). (Bug #36359860)

- It was possible to set the [thread\\_pool\\_longrun\\_trx\\_limit](#) system variable to values outside its stated range.

In addition, settings for this variable were not reflected in the output of SHOW VARIABLES or SELECT. (Bug #36347102, Bug #36371145)

- [SET PERSIST\\_ONLY](#) did not work correctly with [thread\\_pool\\_max\\_transactions\\_limit](#). (Bug #35019884)
- [KILL CONNECTION](#) did not work correctly with [thread\\_pool\\_max\\_transactions\\_limit](#). (Bug #34019954)
- The thread ID was not displayed for client connections in the [performance\\_schema.socket\\_instances](#) table when using the Thread Pool plugin. (Bug #24796018)

## Vector Data Type

- Support is added in this release for a [VECTOR](#) column type. A vector is a data structure which consists of a list of entries (4-byte floating-point values) which can be expressed either as a binary string value or a list-formatted string. A [VECTOR](#) column is declared with a maximum length or number of entries (in parentheses); the default is 2048, and the maximum is 16383.

You can create [InnoDB](#) tables with [VECTOR](#) columns using [CREATE TABLE](#) as shown here:

```
mysql> CREATE TABLE v1 (c1 VECTOR(5000));
Query OK, 0 rows affected (0.03 sec)
```

Other storage engines do not support tables with [VECTOR](#) columns.

Vector columns in this release are subject to restrictions, some of which are listed here:

- A [VECTOR](#) column cannot be used as any type of key. This includes primary keys, foreign keys, unique keys, and partitioning keys.
- Some types of MySQL functions and operators do not accept vectors as arguments. These include but are not limited to numeric functions and operators, temporal functions, full-text search functions, XML functions, bit functions, and JSON functions.

Some (but not all) string and encryption functions support [VECTOR](#) values. For more complete information, see [VECTOR Supported and Unsupported Functions](#).

- A [VECTOR](#) cannot be compared with any other type, and can be compared with another [VECTOR](#) only for equality.
- MLE JavaScript programs do not support [VECTOR](#) columns or values.
- NDB Cluster does not support [VECTOR](#) columns or values.

Use the [VECTOR\\_DIM\(\)](#) function (also added in MySQL 9.0) to obtain the length of a vector. Functions to convert between representations are available. [STRING\\_TO\\_VECTOR\(\)](#) (alias: [TO\\_VECTOR\(\)](#)) takes a list-formatted representation of a vector and returns the binary string representation; [VECTOR\\_TO\\_STRING\(\)](#) (alias: [FROM\\_VECTOR\(\)](#)) performs the inverse, as shown here:

```
mysql> SELECT STRING_TO_VECTOR('[2, 3, 5, 7]');
+-----+
| TO_VECTOR('[2, 3, 5, 7]') |
+-----+
| 0x0000000400000040400000A0400000E040 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT VECTOR_TO_STRING(0x0000000400000040400000A0400000E040);
+-----+
| VECTOR_TO_STRING(0x0000000400000040400000A0400000E040) |
+-----+
| [2.00000e+00,3.00000e+00,5.00000e+00,7.00000e+00] |
+-----+
1 row in set (0.00 sec)
```

For more information and examples, see [The VECTOR Type](#), and [Vector Functions](#). (WL #16081)

## 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)
- An outdated link to the MySQL documentation in the [mysql\\_function\\_names](#) unit test source file has been updated.

Our thanks to Minha Jeong for the contribution. (Bug #113500, Bug #36137217)

## Functionality Added or Changed

- A correlated subquery can now be optimized as an outer left join on a derived table when the subquery contains a `LIMIT 1` clause (previously, such a subquery could not be transformed in this way if it contained any `LIMIT` clause). The `LIMIT` clause must use a literal `1`; if it uses any other value, or a placeholder (`?`) or variable, the subquery is not eligible for this transformation.

For more information, see [Correlated Subqueries](#). (Bug #36475554, WL #16124)

- Join columns are now included in the output of `EXPLAIN FORMAT=JSON`. (Bug #36230046)

## Bugs Fixed

- **Important Change; Replication:** When `replica_parallel_workers` was equal to `1`, incorrect logical timestamps for transactions caused the replica to stop with an error, despite the fact that logical timestamps are not relevant when there is only a single applier thread.

Now, the replica logs a warning when `replica_parallel_workers = 1`, and raises an error only when the value of this system variable is greater than `1`. In addition, the format of and information contained in the error message has been improved, and the same message is now used in the output of `SHOW REPLICA STATUS` as well as in the Performance Schema `replication_applier_status` table. (Bug #36058442)

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

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

- **InnoDB:** Improved the `InnoDB` recovery logic to reduce pauses between recovery actions. (Bug #36332645)
  - **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:** Improved `os_innodb_umask` handling, and made it read-only. (Bug #35932118)
- References: This issue is a regression of: Bug #29472125.
- **InnoDB:** Removed functionality specific to the Fusion IO atomic write feature, a product that was last available in 2014. (Bug #35072139)
  - **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:** Improved the `notify_about_advanced_write_lsn()` logic to prevent potential log notification delays.

Our thanks to Zongzhi Chen for the contribution. (Bug #114660, Bug #36528707)

- **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:** Removed an unnecessary conditional check from `get_next_redo_rseg_from_undo_spaces()`.

Our thanks to Alex Xing for the contribution. (Bug #113640, Bug #36185805)

- **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:** `SELECT ... GROUP BY` queries were at least twice as slow with the TempTable engine than the Memory engine. (Bug #107700, Bug #34338001)
- **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:** The column number returned in the error `ER_SERVER_REPLICA_CONVERSION_FAILED` was incorrect. It was one less than the actual value. (Bug #36246205)
- **Replication:** Handling an incident while transactions were being committed to the binary log caused MySQL to wait indefinitely. (Bug #35671897)
- **Replication:** When using row-based replication with `binlog_rows_query_log_events` enabled, SQL statements are logged in a `Rows_query_log_event` which is written before the `Table_map_event`. Since SQL may contain any binary data including embedded nulls (`\0`), when using `strlen()`, `snprintf()`, and other similar C functions which rely on processing up to the null byte, it was possible for the query string to be truncated at the first null byte, which resulted in processing incomplete data.

We fix this by specifying the length of each `Rows_query_log_event` such that it is no longer necessary to rely on C-style string methods for length calculations; in addition, while processing the query, we now use functions which do not rely on null termination. We also mark the first byte of a `Rows_query_log_event` data body as unused. (Bug #109401, Bug #35336260)

- **Group Replication:** Removed a memory leak from `/xcom/gcs_xcom_networking.cc`. (Bug #36532199)
- **Group Replication:** Under certain circumstances, after successfully setting a new primary, `group_replication_set_as_primary()` waited indefinitely for the operation to complete.  
  
As of this release, a periodic check is performed to ensure the function does not wait unnecessarily. (Bug #36348650)
- **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:** The `MEMBER_ID`, `MEMBER_HOST`, and `MEMBER_PORT` columns of the `REPLICATION_GROUP_MEMBERS` table were not always populated for offline members. (Bug #36290046)
- **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)
- Added a missing error check needed when evaluating the `<=>` operator. (Bug #36570474)
- Added a missing error check needed for evaluating `CASE` operators. (Bug #36570439)
- Averages of certain numbers were not always computed correctly. (Bug #36563773)
- Removed redundant assignments to `Item::m_table_ref` in `find_field_in_tables()` which led to invalid `GROUP BY` results and other errors. (Bug #36556725, Bug #36557029)

References: This issue is a regression of: Bug #36533080.

- 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)
- With `subquery_to_derived=ON`, an outer reference was not replaced in some cases following transformation to a derived table. (Bug #36475633)

References: See also: Bug #36314993.

- A missing check for errors relating to `TIME` values sometimes led to an assert in `sql/item.cc`. (Bug #36421511)
- Explaining a query which used `FORCE INDEX` on a spatial index containing a column with `SRID` attributes led to an unplanned exit. (Bug #36418426)
- The `InnoDB` OpenTelemetry metrics (`mysql.inno`) were not automatically updated. (Bug #36399090)
- This fix addresses two issues:
  - An item that was not yet fixed when searching for an item placed in the `GROUP BY` list led to an assert in `include/sql_string.h`.
  - The `TIME_FORMAT()` function did not handle `NULL` arguments correctly in all cases.

(Bug #36367313, Bug #36367776)

- Updated `BuildRequire` rules to align with versions now required for CMake and Bison. (Bug #36343254)
- Removed an unused argument from the internal function `MY_COLLATION_HANDLER::strstr()`. (Bug #36342997)
- An `IN` predicate containing `EXCEPT ALL` set operations yielded the wrong result. (Bug #36332697)
- With `optimizer_switch` set to `subquery_to_derived=on`, some queries with `ROLLUP` were not handled properly. (Bug #36314993)
- A query using `WHERE primary_key IN(SELECT constant1 EXCEPT SELECT constant2)` returned a differing number of rows depending on the presence or absence of an `ORDER BY` clause. (Bug #36307622)
- 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)
- In transforming subqueries to derived tables, replacement of a subquery in a `HAVING` condition failed to use an item reference, which led to an assert in production builds and an unplanned exit in debug builds. This occurred because the optimizer did not correctly detect that the subquery was part of a `HAVING` condition. (Bug #36079456)

References: This issue is a regression of: Bug #35060385.

- It was possible for `MaterializeIterator<Profiler>::load_HF_row_into_hash_map()` to exhaust resources while re-reading rows. (Bug #36075756)
- Some `CREATE USER` statements were not handled correctly. (Bug #36022885)
- 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.

- 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 always terminated correctly during shutdown. (Bug #35854919)
- Executing `mysqldump` on a replica would insert the `FLUSH TABLES` operation, an operation that writes to the binary log. Now `FLUSH LOCAL TABLES` is inserted instead to prevent GTID related issues during replication due to these binary log changes.

The workaround was to set the `--source-data` option to 1 or 2. (Bug #35665076)

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

- All internal ACL bitmask variables are now explicitly 32 bits (`uint32_t`). (Bug #35507223)
- 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.

- It was not possible to add a functional index on `FIND_IN_SET()`. (Bug #35352161)
- 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)

- In some cases, a `SELECT constant` from an empty table with `ORDER BY COUNT(*)`, when used in a view, did not return any rows. (Bug #115035, Bug #36625752)
- In some circumstances, such when DDL operations were performed on a very large number of tables, the error log was flooded with warnings from background histogram updates; the offending warning was concerning a failure to acquire metadata locks on a table.

To remedy this problem we now throttle messages written to the error log from background histogram update operations, the rate being capped at one message per minute, which should suffice for the user to identify potential problems with background histogram updates. In addition, we downgrade all error events that occur during background histogram updates from errors to warnings. (Bug #114845, Bug #36574298)

- Fixed an erroneous comment in `include/my_command.h`.

Our thanks to Sho Nakazono for the contribution. (Bug #114507, Bug #36455468)

- The range of error numbers for new errors in MySQL 9 has been designated to begin with 6400. (Bug #114414, Bug #36421351)
- When the character set for arguments to a UDF was specified using component services and the argument values passed did not convert cleanly to the desired character set, the UDF ceased executing and returned SQL `NULL`. (Bug #114409, Bug #36420251)
- 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)
- The server rejected a query containing a subquery which referred to a column of the parent table. (Bug #113887, Bug #36262779)
- A query such as the following:

```
SELECT
(
  SELECT COUNT(1) AS cnt
  FROM t2
  WHERE t2.a = t1.a
  HAVING cnt > 0
)
FROM t1;
```

was transformed to this:

```
SELECT COALESCE(derived_1_2.cnt,0) AS cnt
```

```
FROM t1
LEFT JOIN
(
  SELECT COUNT(1) AS cnt,
  t2.a AS a from t2
  GROUP BY t2.a
  HAVING (cnt > 0)
) AS derived_1_2
ON derived_1_2.a = t1.a;
```

The presence of a false `HAVING` condition in the subquery should semantically change the correct result of the scalar subquery from zero to `NULL`, which happened as expected for the original query, but not for the transformed case. (Bug #113319, Bug #36070647)

- `SUM(SUBSTRING())` returned a warning as expected, but `SUM(DISTINCT SUBSTRING())` did not. (Bug #113171, Bug #36035064)
- Added the missing `mysql-community-libs-compat` package for the EL8 and EL9 platforms. (Bug #112949, Bug #35975348)
- `SHOW PARSE_TREE CREATE SCHEMA` caused a server exit in debug builds.



#### Note

The `SHOW PARSE_TREE` statement is available in debug builds only.

(Bug #112883, Bug #35964157)

- A different result was obtained when a column reference argument to the `CHAR()` function was replaced with a `CASE()` expression that was essentially the same as the column reference. This took place when the `CHAR()` function was placed in the `WHERE` clause of an outer join, and the column reference was from the inner table of the outer join. An example of such a query is shown here:

```
SELECT 1 AS c_0
FROM t0
LEFT JOIN t1 ON t0.vkey = t1.vkey
WHERE CHAR(CASE WHEN FALSE THEN t1.vkey ELSE t1.vkey END) NOT LIKE 'X';
```

A wrong value was obtained when the column reference was used directly; the `CHAR()` function in the `WHERE` clause was used to convert the outer join to an inner join, although this is correct only when `NULL` as the argument implies a `NULL` result, which is not true of `CHAR()`. The implementation detail that enforces this conversion is that the function's `not_null_tables()` property function returns the map bit of the table, but when the column reference was replaced with the `CASE()` expression, the `CASE()` did not propagate the `not_null_tables()` value to keep the outer join from being optimized improperly to an inner join.

We fix this problem by setting the `CHAR()` function's `null_on_null` property to false instead of true, which ensures that `not_null_tables()` returns 0 rather than the table's map bit, so that that the outer join is not converted to an inner join when it should not. (Bug #112397, Bug #36118590)

