MySQL 9.4 Release Notes

Abstract

This document contains release notes for the changes in MySQL 9.4. For information about changes in a different version of MySQL, see the release notes for that version.

For additional MySQL 9.4 documentation, see the MySQL 9.4 Reference Manual, which includes an overview of features added in MySQL 9.3 (What Is New in MySQL 9.4), 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: 2025-08-25 (revision: 30457)

Table of Contents

Preface and Legal Notices

This document contains release notes for the changes in MySQL 9.4.

Legal Notices

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

Documentation Accessibility

Access to Oracle Support for Accessibility

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

- 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)

Installation Notes

- **macOS:** When `mysqld` was started without `--plugin-dir`, but with `--basedir`, where the base directory did not end in a slash character (`/`), and `mysqld` was configured to load the `keyring_file` component, server startup failed with errors. This caused problems for the macOS installer for MySQL, leading it not to set the password for the root account. (Bug #36816216)

  References: See also: Bug #36398484.

- Debian packages for installing MySQL can now be run by users other than root. This helps prevent issues for Debian or Ubuntu systems that require rootless installations. (Bug #37765153)
- RPM and Yum repository installation are now supported for Red Hat Enterprise Linux and Oracle Linux 10. (Bug #37592019)

JavaScript Programs

- Under heavy memory usage, when attempting to execute a JavaScript stored procedure, the first `context.parse()` call in `SP::init()` raised an out-of-memory error. Subsequently, a second `context.parse()` call was made asynchronously; this could succeed if other sessions had cleared the memory in the meantime. This meant that an error was reported by the MLE component's error handler, but—since the second call succeeded—MLE returned a status code indicating success. Now in such cases, `CALL` is rejected with an out-of-memory error. (Bug #37952656)
- Upgraded the bundled version of Graal Maven to 24.2.1.0.1. (Bug #37938310)

- The included GraalVM and the Truffle library used by the MLE component have been upgraded to versions 23.1.7 and 24.2.1, respectively. (Bug #37833200)

- When `mysqldump` was from a MySQL 9.2.0 or later distribution, `mysqldump --routines` did not work properly with servers from previous versions, because it could not find the Information Schema `LIBRARIES` table. Now in such cases, `mysqldump` skips this table and does not try to dump it. (Bug #37498680)

- The MySQL `BIT` type is now supported for use in JavaScript stored routines. For type conversion rules and other information, see JavaScript Stored Program Data Types and Argument Handling. (WL #16885)

- JavaScript programs using the MySQL Multilingual Engine component now supports libraries written in WebAssembly (see `https://webassembly.org/`). Such a library can be created using `CREATE LIBRARY ... LANGUAGE WASM` and the hexadecimal or base64-encoded representation of a compiled WebAssembly program as shown here:

```
CREATE LIBRARY test_lib LANGUAGE WASM
  AS $$AGFzbQEAAAABFwVgAABgAX8Bf2ACf38Bf2ABfwBgAAF/AwYFAAECAwQEBQFwAQICBQYBAYICg
gIGCAF/AUGQiAQLB4sBBwZtZW1vcnkCAAxhZGRfdG9fc3RhdGUAAQpteV9hZGRfaW50AAILX2luaXRpY
WxpemUAABlfX2luZGlyZWN0X2Z1bmN0aW9uX3RhYmxlAQAZX2Vtc2NyaXB0ZW5fc3RhY2tfcmVzdG9yZ
QADHGVtc2NyaXB0ZW5fc3RhY2tfZ2V0X2N1cnJlbnQABAkHAQBBAQsBAAovBQMAAQsVAEGACEGACCgCA
CAAaiIANgIAIAALBwAgACABagsGACAAJAALBAAjAAs=$$;
```

Note

We have wrapped the lines of the WebAssembly code in this example for legibility. To reproduce the statement locally, you should remove any linefeeds appearing between the leading and trailing `$$` delimiters.

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 are updatable using `INSERT`, `UPDATE`, or `DELETE` statements, so that updates of such a view cause corresponding updates on the base table or tables to be performed. Updatable JSON duality views are supported in MySQL Enterprise Edition only.

  As part of this work, MySQL also now supports an `ETAG()` function which returns a 128-bit hash for each row; this value is shown in a `SELECT` from a JSON duality view, as shown here:

```
mysql> SELECT * FROM jdv1;
+-----------------------------------------------------------------------------------------------------------
| data
+-----------------------------------------------------------------------------------------------------------
| {"_id": 1, "key": "base64:type252:bG9uZ19ibG9iX2RhdGE=", "_metadata": {"etag": "0bb9af42e150011737
+-----------------------------------------------------------------------------------------------------------
```

  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_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.
  - `JSON_DUALITY_VIEW_TABLES`: Provides information about tables referenced by JSON duality views.

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)

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)

- 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)

- 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)