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

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/utr11-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_prefers_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` 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:

  a. All such comparisons are now supported (`>ANY`, `>=ANY`, `<ANY`, `<=ANY`; `>ALL`, `>=ALL`, `<ALL`, `<=ALL`; `=ANY`, `<>ALL`).

  b. 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 handlerton `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]
  ```

```
[ERROR] [MY-013183] [InnoDB] Assertion failure: buf0buf.cc:5889:ib::fatal trig
```

(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 behaviour 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 retrived, 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)

- ***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, 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 updateable, 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 #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.

  Out 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` handlerton `get_tablespace()` method. (Bug #109443, Bug #34916556)