MySQL 9.0 Release Notes

Abstract

This document contains release notes for the changes in MySQL 9.0. For information about changes in a different version of MySQL, see the release notes for that version.

For additional MySQL 9.0 documentation, see the MySQL 9.0 Reference Manual, which includes an overview of features added in MySQL 9.0 (What Is New in MySQL 9.0), 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: 2024-07-03 (revision: 28669)

Table of Contents

Preface and Legal Notices

This document contains release notes for the changes in MySQL 9.0.

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 9.0.0 (2024-07-01, Innovation Release)

Audit Log Notes

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

  Our thanks to Martin Alderete for the contribution.

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.

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)
- Upgraded the bundled `googletest` and `googlemock` sources to version 1.14.0. (Bug #36562482)
- Added a missing dependency on `GenError`. (Bug #36551721)
- 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)
- 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

- 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 (the plugin was 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)

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 #36361165)

Deprecation and Removal Notes

- 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—Obtaining Session Information.

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

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

- 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)

sys Schema Notes

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

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.

  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]')                             |
+-------------------------------------------------------+
| 0x0000004000004040000040A0400000E040                  |
+-------------------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT VECTOR_TO_STRING(0x0000004000004040000040A0400000E040);
+-------------------------------------------------------+
```

```
| VECTOR_TO_STRING(0x0000004000004040000A0400000E040) |
+-----------------------------------------------------+
| [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)

Bugs Fixed

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

  References: This issue is a regression of: Bug #35183686.
- **InnoDB:** 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:** The column number returned in the error `ER_SERVER_REPLICA_CONVERSION_FAILED` was incorrect. It was one less than the actual value. (Bug #36246205)

- **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:** 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:** 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)

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

  References: This issue is a regression of: Bug #17809, Bug #11745618.
- 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)
- 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)
- 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)
- Some `CREATE USER` statements were not handled correctly. (Bug #36022885)
- For a `SELECT` with `ORDER BY` and `LIMIT`, the optimizer first chose a full table scan with a very expensive cost, then performed another check and used the `perform_order_index` type of path, but this was not reflected by the cost in the optimizer plan. (Bug #35930969)
- 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)
- It was not possible to add a functional index on `FIND_IN_SET()`. (Bug #35352161)
- 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)
- 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)
- `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)
- The `PROCESSLIST_INFO` column of `THREADS` was not updated when executing a prepared statement. (Bug #104121, Bug #33057164)