MySQL 26.7 Release Notes

Abstract

This document contains release notes for the changes in MySQL 26.7. For information about changes in a different version of MySQL, see the release notes for that version.

For additional MySQL 26.7 documentation, see the MySQL 26.7 Reference Manual, which includes an overview of features added in MySQL 26.7 (What Is New in MySQL 26.7), and discussion of upgrade issues that you may encounter while upgrading.

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

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

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

For legal information, see the Legal Notices.

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

Document generated on: 2026-09-14 (revision: 31556)

Table of Contents

Preface and Legal Notices 1
Changes in MySQL 26.7.1 (2026-08-18) 3
Changes in MySQL 26.7.0 (2026-07-28) 3

Preface and Legal Notices

This document contains release notes for the changes in MySQL 26.7.

Legal Notices

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

License Restrictions

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

Warranty Disclaimer

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

Restricted Rights Notice

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

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

Hazardous Applications Notice

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

Trademark Notice

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

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

Third-Party Content, Products, and Services Disclaimer

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

Use of This Documentation

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

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

as any dissemination of printed copies or use of this documentation, in whole or in part, in another publication, requires the prior written consent from an authorized representative of Oracle. Oracle and/or its affiliates reserve any and all rights to this documentation not expressly granted above.

Documentation Accessibility

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

Access to Oracle Support for Accessibility

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

Changes in MySQL 26.7.1 (2026-08-18)

Note

This release is a Critical Security Patch Update for the MySQL Server docker image, only. See Oracle Critical Security Patch Update Advisory - August 2026 for more information.

Changes in MySQL 26.7.0 (2026-07-28)

- Audit Log Notes
- Compilation Notes
- Component Notes
- Data Dictionary Notes
- Deprecation and Removal Notes
- INFORMATION_SCHEMA Notes
- InnoDB Notes
- JSON Notes
- MySQL Programs Notes
- Optimizer Notes
- Packaging Notes
- Performance Schema Notes
- Replication Notes
- Security Notes
- Thread Pool Notes
- X Plugin Notes
- Functionality Added or Changed

- Bugs Fixed

Audit Log Notes

- With legacy audit-log filtering, setting `audit_log_include_accounts`, clearing it to `NULL`, and then changing `audit_log_exclude_accounts` could filter out every account and stop audit logging. (Bug #35137790)
- The server could close unexpectedly if `audit_log_encryption_password_get(keyring ID)` was called. (Bug #33296125)
- The following Audit Log status variables are introduced in this version:

  - `Audit_log_file_count` (Audit plugin)
  - `audit_log.file_count` (Audit component)

  This variable reports the number of audit log files generated by the Audit Log plugin or component when `audit_log_format=JSON`. The value is incremented for automatic rotations and decremented for pruning actions. (WL #17292)

Compilation Notes

- The version of `libfido2` used with MySQL was upgraded to 1.17.0. (Bug #39478664)
- Fixed a potential memory leak in the comp_err utility. (Bug #36532183)

Component Notes

- The command-service default DOM consumer stored SQL NULL fields as literal NULL string and recomputed field lengths with strlen() from the buffered row. This made SQL NULL indistinguishable from a real "NULL" value and truncated binary values at embedded NUL bytes. (Bug #39405458)
- The statement result metadata service exposes `field_info("type")` as a stored-program argument type. That value is not always the raw MySQL field type carried in `Column_metadata::type`, so consumers cannot retrieve the original field type value from result metadata.

  As of this release, a component-safe `mysql_field_type_t` type and `MYSQL_FIELD_TYPE_*` values are added under `services/bits` to mirror raw MySQL field type values. That type is used for a new `field_info("mysql_type")` metadata key while preserving the existing type behavior. (Bug #39359509)
- In previous releases, command-service consumers did not have a public `mysql_command_options` API to set or read extra client flags on a command-service MYSQL connection.

  The new command-services option, `MYSQL_COMMAND_CLIENT_FLAGS`, passes client flags to `mysql_real_connect()`.

  Custom consumers can continue to override command capabilities with `MYSQL_TEXT_CONSUMER_CLIENT_CAPABILITIES`. The callback delegate remains unaware of `MYSQL_COMMAND_CLIENT_FLAGS`. (Bug #39349732)
- Command-service consumers could not generically retrieve session-tracker information—such as schema, system-variable, transaction, and GTID changes—from a MYSQL_H handle.

  A new mysql_command_session_state service now provides iterator-based access to cached session-tracker entries. The cache is refreshed for each command, cleared on reset or errors, and retains the latest completion state for multi-result commands, giving consumers consistent access to current session state. (Bug #39330990)

- If default service-name insertion failed during service registration, a subsequent component unload could access a mapping to a destroyed service implementation. The failed registration path now removes the mapping. (Bug #120342, Bug #39303621)

Data Dictionary Notes

- Data dictionary validation now checks invalid SET definitions with no elements in addition to ENUM definitions.

  Our thanks to Yongfei Ji for the contribution. (Bug #36357937)

Deprecation and Removal Notes

- The following replication system variables are deprecated in this release and scheduled for removal in a later release:

  - `group_replication_communication_stack`
  - `group_replication_ip_allowlist`

  (WL #17300)

INFORMATION_SCHEMA Notes

- Large `AUTO_INCREMENT` values (>INT64_MAX) were truncated in `information_schema.tables.auto_increment`. (Bug #116149, Bug #37077567)

InnoDB Notes

- **InnoDB:** Fixed a potential race condition in processing of `SET GLOBAL innodb_redo_log_encrypt = ON;`. (Bug #39484886)
- **InnoDB:** Fixed an issue relating to Clone validation. (Bug #39252970)
- **InnoDB:** Fixed a debug assertion in `row_upd_rec_in_place()` for non-versioned `ROW_FORMAT=REDUNDANT` rows. (Bug #39165747)
- **InnoDB:** During a buffer-pool resize, Adaptive Hash Index could be disabled in mysqld-auto.cnf even though innodb_adaptive_hash_index=ON. (Bug #39157211)
- **InnoDB:** Fixed an issue relating to table maintenance and clean-up. (Bug #39091376)
- **InnoDB:** Fixed an issue relating to unique indexes. (Bug #38501299)
- **InnoDB:** InnoDB buffer-pool index statistics could report implausibly high `INFORMATION_SCHEMA.INNODB_CACHED_INDEXES.N_CACHED_PAGES` values, including values above the number of table pages or the buffer pool capacity. The metric now counts cached leaf index pages for the tracked index in the buffer pool structures and handles page reuse, split and merge operations, compressed pages, spatial indexes, and SDI correctly. (Bug #30164033)
- **InnoDB:** In workloads that frequently create, truncate, or drop tables, InnoDB's cached-index statistics map could grow very large because obsolete entries were retained. Adding a new index entry could then trigger expensive internal map maintenance while page latches were held, causing other transactions to wait for extended periods.

  InnoDB now pre-creates the statistics entry before btr_create() begins making page changes. This moves potentially costly map maintenance outside the latch-sensitive work, reducing prolonged contention during index creation and table truncation. (Bug #120441, Bug #39509101)
- **InnoDB:** ReadView objects are shared through an intrusive list that is frequently accessed by both transaction threads and the purge thread. Their padding was intended to keep the list-link node on a separate CPU cache line, avoiding cache contention between these concurrent operations. However,

the implementation assumed a 64-byte cache line and calculated padding based on the list node's size, which did not reliably provide the intended isolation.

The padding now reserves a full, platform-specific InnoDB cache line. This removes the hardcoded assumption and ensures the list-link node is separated from other ReadView fields regardless of its position within the object, reducing the risk of false sharing on different hardware platforms.

Our thanks to Kaiwang Chen and the team at Tencent for the contribution. (Bug #116878, Bug #37569394)

- **InnoDB:** Under certain circumstances, when adding an AUTO_INCREMENT column to an existing table, some records in that table could be skipped, resulting in inaccurate values in the AUTO_INCREMENT column.

  Our thanks to Huaxiong Song and the team at Alibaba for the contribution. (Bug #115136, Bug #37105825)

- **InnoDB:** Source builds with WITH_INNODB_EXTRA_DEBUG enabled could fail.

  Our thanks to Sam Bird for the contribution. (Bug #114455, Bug #37706575)

- Under high I/O load, particularly with many threads and the open-file limit exhausted, concurrent temporary-tablespace truncation could cause a deadlock and server failures while a stale buffer page was prepared for I/O. (Bug #39244016)

- Valid `CREATE TABLE` or `INSERT` operations involving an InnoDB `SPATIAL KEY` could be rejected with `ER_TOO_BIG_ROWSIZE`. Errors were returned similar to the following:

  ```
  ER_TOO_BIG_ROWSIZE(1118) (42000): Row size too large (> 8126).
  Changing some columns to TEXT or BLOB may help. In current row format,
  BLOB prefix of 0 bytes is stored inline.
  ```

  Our thanks to Pawel Olchawa and the team at Percona for the contribution. (Bug #120504, Bug #39433811)

- MySQL Clone plugin supports the updated CalVer model.

  - Cloning is allowed when the version strings match.
  - Cloning is allowed across patch versions of a given version.
  - Cloning is allowed from donor in one LTS to recipient in next LTS.
  - Cloning is not allowed from donor in one LTS to recipient in an older LTS.
  - Cloning is not allowed from donor in one LTS to recipient in a later LTS; an LTS which is not the next in the sequence.
  - Cloning is not allowed if either donor or recipient is not LTS and the major and/or minor versions mismatch.

  (WL #17317)

- InnoDB undo tablespace creation and truncation no longer rely on a local undo truncate log files. Undo truncation progress information is now stored in the Undo Tablespace header.

  As of this release, undo truncation logs are no longer generated.

  Note

  For backward compatibility, existing truncate log files are still supported, but no new truncate logs are created.

(WL #16989, WL #17187)

- `innodb_autoinc_preallocate` is added in this release. This system variable enables InnoDB to reserve additional `AUTO_INCREMENT` values in persistent metadata. When enabled, unused reserved values can be skipped after an unexpected shutdown or crash. A graceful shutdown persists the current value and does not introduce gaps.

  Preallocation does not apply to `AUTO_INCREMENT` integer types of two bytes or less, during `CLONE` operations, or when the variable is set to `0`.

  Applications should not rely on gap-free `AUTO_INCREMENT` sequences.

  See AUTO_INCREMENT Handling in InnoDB. (WL #15550)

JSON Notes

- Updating a JSON relational duality view that deleted rows from a nested child base table could violate foreign-key constraints. JSON Duality View update processing now performs deletes first, in reverse binding order, before inserts and updates are executed. (Bug #39566804)
- Fixed an issue relating to JSON Schema validation. (Bug #39489024)
- Fixed an issue relating to processing JSON date values. (Bug #39361343)
- Fixed an issue relating to processing of nested arrays. (Bug #39255759)
- Fixed an issue relating to metadata validation. (Bug #39255734)
- `JSON_VALUE()` with `RETURNING DATETIME` and a `DEFAULT` value could return an invalid constant date when the expression was used in a `DATE` context, silently affecting predicates and joins. The datetime value is now extracted before narrowing to a date. (Bug #120788, Bug #39649893)

MySQL Programs Notes

- When dumping the `mysql.innodb_table_stats`, and `mysql.innodb_index_stats` tables with `--replace` option, `mysqldump` added an `IGNORE` clause for the `REPLACE` command.

  Our thanks to Jinyou Ma for the contribution. (Bug #119810, Bug #38939358)

- `mysql_tzinfo_to_sql` generated unnecessary warnings. For example:

```
Warning: Unable to load '/usr/share/zoneinfo//iso3166.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo//leap-seconds.list' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo//leapseconds' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo//tzdata.zi' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo//zone.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo//zone1970.tab' as time zone. Skipping it.
```

  As of this release, those warnings are suppressed.

  Our thanks to Daniël van Eeden for the contribution. (Bug #115021, Bug #36624161)

- `--extended-insert-multiline` option is added for `mysqldump`. This option can be used with `--extended-insert` to write table data using a separate line for each row.

  `--extended-insert-multiline` is disabled by default.

  Our thanks to Michiel Beijen for the contribution. (Bug #112719, Bug #35912859)

Optimizer Notes

- Fixed an issue relating to `subquery_to_derived`. (Bug #39525738)

- A query using the hypergraph optimizer with GROUP BY ... WITH ROLLUP on a constant expression and an empty USE INDEX FOR GROUP BY() hint could cause an assertion failure during optimization.

  As of this release, the optimizer recognizes that constant grouping requires no index ordering, allowing a valid streaming aggregate plan to be selected. (Bug #39505077)

- Fixed an issue relating to `ROLLUP` queries. (Bug #39501528)

- Fixed an issue related to error handling in aggregation of decimal data. (Bug #39474810)

- Fixed an issue related to executing EXPLAIN on a LEFT JOIN. (Bug #39461426)

- Some temporal and geospatial expressions could trigger assertion failures. (Bug #39279682, Bug #39280411)

- Fixed an issue relating to error-handling. (Bug #39272322)

- Fixed an issue relating to user-variable handling. (Bug #39243027)

- Under certain circumstances, predicate-heavy queries using the hypergraph optimizer could trigger an assertion failure. (Bug #39181175)

- Fixed an issue relating to Common Table Expressions with `subquery_to_derived` enabled. (Bug #39179197)

- Fixed an issue in `ExtractValue()`. (Bug #39138426)

- Fixed an issue relating to string-to-temporal SQL expression evaluation. (Bug #39135197)

- Complex queries using the hypergraph optimizer could spend unnecessary planning time repeatedly checking for a secondary-engine row-count hook, even when no secondary engine was installed.

  The optimizer now caches this hook lookup once per query, reducing planning overhead. (Bug #39015463)

- Opening a table after updating its histogram could trigger an assertion failure because metadata locks for histograms were released incorrectly.

  Histogram metadata locks are now released only for the table currently being opened. (Bug #38940484)

- In debug builds, an `UPDATE` assigning the result of an `IN` subquery could abort when the subquery used a derived table with `UNION`. The failure reported an assertion similar to the following:

  ```
  Assertion `(common_table_expr() !=
    nullptr && common_table_expr()->references.size() > 1) || tab == nullptr ||
    tab->type() != JT_CONST || table->has_null_row()' failed.
  ```

  (Bug #38911659)

- Fixed an issue relating to dynamic scan execution.

  Our thanks to Mengchu Shi and the team from Alibaba for the contribution. (Bug #38657550)

- Queries which used `BETWEEN` with constant bounds could show an unnecessary filter operation above an otherwise exact index range scan when using the hypergraph optimizer.

  As of this release, the optimizer recognizes these range conditions as exact and removes the redundant filter step. (Bug #37152269)

- Queries involving materialized derived tables could choose an index-lookup plan even when the cost of building the temporary index outweighed its benefit.

As of this release, the hypergraph optimizer includes temporary-index build cost when evaluating these plans, improving plan selection for derived-table queries. (Bug #36957877)

- `SELECT DISTINCT ... LIMIT` queries, which used materialization-based duplicate removal, could process all distinct rows before applying the limit, causing performance degradation when only a small number of unique rows was requested.

  The hypergraph optimizer now passes the query limit to duplicate removal, allowing it to stop once it has found the required number of unique rows. Sort-based `DISTINCT` plans forced by `SQL_BIG_RESULT` are unchanged. (Bug #36720017)

- Hash joins could perform unnecessary disk spill I/O for queries with a low LIMIT, even when only a small number of result rows was needed.

  As of this release, the hypergraph optimizer considers both spill-to-disk and in-memory hash join plans, allowing cost-based optimization to favor the faster in-memory option when appropriate. (Bug #36684053)

- ORDER BY expressions using stored functions with side effects or non-deterministic functions such as RAND() could be evaluated more than once per row by the hypergraph optimizer, resulting in unexpected repeated side effects.

  As of this release, rows are materialized before sorting in affected queries so these expressions are evaluated only once per row. Top-level UPDATE and DELETE statements, and queries optimized for secondary engines, are unaffected. (Bug #36578540)

- Queries that referenced the same common table expression (CTE) from multiple query blocks could generate duplicate internal temporary-index names. When the CTE was materialized on disk, InnoDB could select the wrong internal index, resulting in a crash, assertion failure, “table does not exist” error, or incorrect query result.

  Internal temporary-index names are now regenerated when indexes are moved between slots, ensuring each generated index remains uniquely and consistently named.

  Our thanks to Wang Zhengmao for the contribution. (Bug #36492739)

- Under certain circumstances, filters using `IFNULL` returned incorrect results. (Bug #36448705)

- UNION ALL evaluated children in textual order. With EXISTS or LIMIT, this could cause expensive branches to be evaluated first, when a cheaper branch could produce a row immediately. (Bug #35767452)

- The execution cost of an independent scalar subquery could be counted once per outer row instead of once per query, producing inaccurate `EXPLAIN` costs and potentially affecting plan selection. One-time initialization costs are now separated from per-row filter costs, including during nested-loop rescans. (Bug #34513189)

- Planning a `NATURAL JOIN` involving many shared columns could be substantially slower with the hypergraph optimizer. (Bug #33910388)

- An `IN` predicate containing `DATE` or `TIME` constants could evaluate as true for a non-temporal string that matched no list element, causing unrelated rows to be included. (Bug #120787, Bug #39649755)

- `POW()` with a `NULL` exponent incorrectly returned 1 rather than `NULL`, which could alter computed values or query filtering. (Bug #120779, Bug #39641087)

- A negated IS TRUE condition involving a BETWEEN predicate could return incorrect results for DECIMAL values because NULL-state handling was not initialized correctly. (Bug #120540, Bug #39444505)

- AVG() of functions such as ASCII() could return a truncated value when used with SQL_BUFFER_RESULT, because the calculated decimal precision did not reserve space for a sign.

As of this release, decimal precision calculations correctly account for signed values, preventing truncation of valid results such as 255.0000. (Bug #120465, Bug #39392429)

- Due to a fix in a previous version, IFNULL() did not handle NULL correctly. (Bug #120429, Bug #39347654)

  References: This issue is a regression of: Bug #37285902.

- A fix in a previous version caused an issue where a WHERE equality condition could fail to match an existing row when a scalar subquery and its outer query referenced the same CTE.

  This fix prevents cleanup of derived-table temporary keys from a query block that does not own them, preserving correct key metadata during outer-query optimization and ensuring correct result matching. (Bug #120403, Bug #39321676)

  References: This issue is a regression of: Bug #35912840.

- Comparing a multi-column row with an empty scalar subquery using the `<=>` operator could return unexpected rows instead of an empty result because residual cached values were used.

  Our thanks to Xingyu Yang, Ximin Liang and the team from Tencent for their contribution. (Bug #119861, Bug #38969673)

- UNION and UNION ALL queries with pushed-down WHERE conditions involving AND and IS NULL could return incorrect rows.

  Condition pushdown now preserves WHERE predicate semantics in set-operation branches. (Bug #119780, Bug #38888307)

- Queries involving nested merged views and outer joins could return incorrect results when a view column containing a constant expression was evaluated before outer-join null completion.

  The optimizer retains the correct inner-table dependency for merged view columns, ensuring outer-join expressions and related WHERE conditions are evaluated at the proper stage. (Bug #119499, Bug #38729126)

  References: This issue is a regression of: Bug #35854686.

- Queries that filtered an indexed column with IS NULL could perform an unnecessary filesort when the same column appeared in ORDER BY.

  As of this release, the optimizer recognizes IS NULL as a constant condition for order-by simplification, allowing index order to be used when applicable.

  Our thanks to Kishore Kumar for the contribution. (Bug #119442, Bug #38700622)

- Queries with ORDER BY, a low LIMIT, and a reverse range scan on a composite index could return incorrect results if the optimizer removed the sort operation prematurely.

  As of this release, the optimizer accounts for all key parts required for ordering before selecting a reverse index range scan.

  Our thanks to Somrak Monpengpinij for the contribution. (Bug #119198, Bug #38564020)

- When an index hint forced a secondary index, the optimizer could add an unnecessary `Sort` operator for `ORDER BY` or `GROUP BY` because it did not account for constant primary-key parts in the index ordering.

  Our thanks to Xingyu Yang and the team at Tencent for the contribution. (Bug #118960, Bug #38406661)

- CAST(NULLIF(...) AS DATETIME) could return an incorrect result because temporal values produced by NULLIF() were not evaluated with the appropriate temporal type handling.

  Temporal evaluation is now handled correctly for IF() and NULLIF() expressions. As a result, NULLIF() applied to a YEAR expression now preserves the YEAR data type; related aggregate expressions may therefore return DECIMAL rather than DOUBLE.

  Our thanks to Ximin Liang for the contribution. (Bug #118053, Bug #37875484)

- FORCE INDEX on a unique index could still perform an index dive for IS NULL predicates, contrary to the expected behavior of forcing a single index.

  The optimizer now skips the index dive in this case while continuing to use index statistics only when they are valid for the selected range.

  Our thanks to Kaiwang Chen and the team from Tencent for the contribution. (Bug #117791, Bug #37751691)

- Queries using `FORCE INDEX` with an `IS NOT NULL` predicate could perform an unnecessary index dive instead of applying the intended single-forced-index optimization. The index dive is now skipped as intended.

  Our thanks to Kaiwang Chen for the contribution. (Bug #117750, Bug #37727982)

- Temporal results returned by GREATEST(), LEAST(), or COALESCE() could report an incorrect collation through the client protocol and the COLLATION() function. Temporal values use the binary collation, but these multi-purpose functions did not apply the required translation. Temporal results from these functions now correctly report the binary collation.

  Our thanks to Dirkjan Bussink for the contribution. (Bug #117543, Bug #37626671)

- Queries using CONST access on a materialized temporary table could return incorrect results after an in-memory temporary table was converted to an on-disk table.

  As of this release, the temporary table's read state is reset after conversion, ensuring the expected rows remain accessible.

  Our thanks to Jingqi Tian for the contribution. (Bug #116741, Bug #37308710)

- Some `LEFT JOIN` queries with `OR` conditions displayed dynamic range access in `EXPLAIN`, but still performed full scans of the inner table at execution time. This occurred because the optimizer did not recognize eligible outer-join predicates when evaluating range access.

  The optimizer now considers these predicates for dynamic range analysis, allowing eligible `LEFT JOIN` queries to use an index range scan re-planned for each outer row instead of repeatedly scanning the full inner table.

  Our thanks to Hope Lee for the contribution. (Bug #113288, Bug #36061036)

- For some `LEFT JOIN` queries, an optimizer rewrite could prevent a non-equality condition from being applied while building a hash join. As a result, more rows than necessary could be added to the hash table before filtering.

  The optimizer now remaps eligible predicates to the build-side table when possible, allowing earlier filtering and more efficient hash join execution without changing inner-join filtering behavior.

  Our thanks to Wen He and Yicheng Wei for their contribution. (Bug #112198, Bug #35749693)

- Queries that assigned names to derived-table columns produced by GROUP BY ... WITH ROLLUP could report an “Unknown column” error when a window function ordered by one of those names.

  Materialized rollup fields now retain their derived-table column names, allowing these references to resolve correctly.

  Our thanks to Hope Lee and the team at AliBaba for the contribution. (Bug #112112, Bug #35722845)

- Using a view created with one or more CTE definitions could result in a syntax error.

  Our thanks to Vitali Vinahradski for the contribution. (Bug #111559, Bug #35536775)

- A cached table could retain temporary-NULL state after a multi-table UPDATE involving a trigger, causing later valid statements to warn or fail under strict SQL mode. Errors were returned similar to the following:

  ```
  ERROR 1048 (23000): Column 'columnName' cannot be null
  ```

  Our thanks to Shun Yi and the team from Alibaba for the contribution. (Bug #110962, Bug #35373634)

- A `SELECT` using an indexed unsigned `DOUBLE` column could return the row containing zero instead of an empty result when compared with a negative constant expression having a string result type.

  The out-of-range conversion is now recognized as an impossible range.

  Our thanks to Kun Qin for the contribution. (Bug #110125, Bug #35100734)

- `INSERT` statements into a partitioned table could access and lock all partitions when an `AUTO_INCREMENT` column participated in the partitioning expression, even when every row supplied an explicit deterministic value. Partition pruning now proceeds when every value is constant, non-NULL, and not an auto-generated zero. Pruning remains disabled for nonconstant values, `NULL`, or zero unless `NO_AUTO_VALUE_ON_ZERO` applies.

  Our thanks to Hope Lee for the contribution. (Bug #109601, Bug #34972980)

- `TIMESTAMPDIFF()` returned unexpected results when used with a `WHERE` clause. (Bug #108143, Bug #34499767)

- The `explain_filename()` implementation was simplified by removing redundant code.

  Our thanks to Hiroyuki Sasagawa for the contribution. (Bug #104169, Bug #33064492)

Packaging Notes

- As of this release, MySQL no longer supports Enterprise Linux 7 (and associated glibc 2.17 generic).

Performance Schema Notes

- Fixed an issue relating to `OBJECT_INSTANCE_BEGIN` columns. (Bug #39449066)

- A fix in a previous version caused a performance overhead in prepared statement digests. (Bug #38809758)

  References: This issue is a regression of: Bug #36445818.

- Queries with the same `SQL_DIGEST` could produce identical summaries, even if they had distinct characteristics, such as differing `ORDER BY` clauses. The digest computation has been improved to better distinguish between these queries.

  Our thanks to Kaiwang Chen for the contribution. (Bug #31103386, Bug #115943, Bug #36997723)

Replication Notes

- **Important Change:** MySQL 26.7 introduces the Change Stream Applier (CSA), a new implementation of the replica SQL applier for multi-threaded replication. CSA is an opt-in, per-channel alternative to the Multi-Threaded Applier (MTA). It uses a modular scheduling and execution design to improve parallel transaction application, make replication resources configurable per channel, and provide a foundation for future replication enhancements. CSA continues to use the existing replication channel commands and the relay log. Receiver-thread and storage-engine interfaces are unchanged, allowing supported channels to be migrated without changing the source-side replication protocol.

  CSA enables the following:

  - Configuring the replication applier independently for each channel.
  - Configuring one to 1,024 CSA worker threads per channel, rather than relying on one global value for all channels.
  - Separation of transaction application from commit ordering. When a worker cannot yet commit, it can apply another dependency-ready transaction instead of waiting idle.
  - Allows parallel reads of relay-log-backed transactions and release cached event memory as transactions are consumed.
  - Purges an inactive relay log file after all of its consumers have completed successfully and all preceding relay log files are also eligible for purging; a failed consumer retains its required file and prevents later files from being purged past it.
  - Improves responsiveness to `STOP REPLICA` and keep other channels running when one CSA channel encounters an error.
  - Exposes CSA configuration and worker state through Performance Schema.

  The following channel options are added:

  - `APPLIER_VERSION = 1|2`: Selects the applier implementation.
    - `1`: Multi-Threaded Applier. Default for new channels.
    - `2`: CSA
  - `APPLIER_WORKER_COUNT`: CSA-only. Sets the number of CSA workers for the channel. Range is 1-1024. If omitted for a new channel, `replica_parallel_workers` is used.
  - `APPLIER_EVENT_MEMORY_LIMIT`: CSA-only. Sets a per-channel cache limit for binlog events. Default is 0 when omitted for a new channel. The effective cache limit is at least equal to the value of `replica_max_allowed_packet`, a lower configured value produces a warning.

  For example:

```
CHANGE REPLICATION SOURCE TO
  APPLIER_VERSION = 2,
  APPLIER_WORKER_COUNT = 8,
  APPLIER_EVENT_MEMORY_LIMIT = 1073741824
  FOR CHANNEL 'channel_name';
```

  To update an existing channel, you must stop the SQL thread first.

  The following features are not supported when CSA is enabled:

  - Statement-based or mixed-format binary logs.

  - File-position replication channels.
  - Replication from GTID-disabled sources or `ASSIGN_GTIDS_TO_ANONYMOUS_TRANSACTIONS`.
  - `GTID_MODE` values other than `ON`.
  - `GTID_ONLY=0` or `REQUIRE_ROW_FORMAT=0`.
  - Delayed replication (`SOURCE_DELAY`).
  - `sql_replica_skip_counter` for skipping portions of transactions.
  - `START REPLICA ... UNTIL` conditions other than `SQL_BEFORE_GTIDS` and `SQL_AFTER_GTIDS`.
  - replica_pending_jobs_size_max; use `APPLIER_EVENT_MEMORY_LIMIT` for CSA event-cache control instead.
  - `IGNORE_SERVER_IDS`, unnecessary for GTID replication because applied GTIDs are skipped automatically.
  - Legacy VCLE.
  - Extended applier statistics.

  See Change Stream Applier. (WL #10500)

- **Replication:** Fixed an issue relating to client BINLOG execution with REPLICATION_APPLIER. (Bug #39474825)
- **Replication:** Fixed an issue relating to heartbeat handling. (Bug #39377010)
- **Replication:** On writable replicas with log_replica_updates disabled, concurrent local commits and applier work could cause a race condition. (Bug #39259437)
- **Replication:** Fixed an issue relating to event rotate handling. (Bug #39253359)
- **Replication:** A replication receiver configured with SOURCE_RETRY_COUNT=0 could stop after a later transient connection loss, despite SOURCE_RETRY_COUNT=0 allowing unlimited reconnect attempts. The inconsistency could also cause the receiver to exit without logging an additional reconnect error.

  Replication now consistently treats SOURCE_RETRY_COUNT=0 as unlimited across reconnect paths, allowing the receiver to continue retrying and catch up after repeated transient failures. (Bug #39086875)
- **Replication:** Fixed an issue relating to error handling. (Bug #38924622)
- **Replication:** `ALTER USER` statements with explicit empty authentication data were rewritten for the binary log without preserving the empty authentication string. This resulted in `password_expired` errors. (Bug #38478657)
- **Replication:** If a replication channel was configured using an applier user, the privilege check for `CREATE` and `DROP` statements failed if those privileges are granted only at the database-specific level for that user. (Bug #37587555)
- **Replication:** Under certain circumstances, multiple simultaneous queries on the `binary_log_transaction_compression_stats` table could cause the server to stop with an error. (Bug #36196212)

- **Replication:** If `explicit_defaults_for_timestamp` was disabled, row-based replica apply or mysqlbinlog replay could stop when recomputing a stored, generated `TIMESTAMP NOT NULL` column from `NULL` input. Errors were returned similar to the following:

  ```
  Could not execute Write_rows event on table tablename; Column 'timestamp_col2'
  cannot be null, Error_code: 1048; handler error No Error!
  ```

  The row is now applied using the field default and a warning is emitted. (Bug #120128, Bug #39182574)

- Fixed several issues relating to validations. (Bug #39234600, Bug #38673268, Bug #38673285, Bug #38712932, Bug #39253416, Bug #39282368, Bug #39253383, Bug #39253491, Bug #39254867, Bug #39254885, Bug #39254896, Bug #39254914, Bug #39282350, Bug #39319907)

- As of this release, the default value of `group_replication_communication_stack` is changed from `XCOM` to `MYSQL`. (WL #15709)

Security Notes

- MySQL now supports post-quantum cryptography (PQC) in TLS 1.3 connections for OpenSSL 3.5.0 or later. This enables negotiation of PQC-capable TLS key-exchange groups, including hybrid classical/PQC groups, and advertising of PQC TLS handshake signature algorithms.

  The following system variables are introduced:

  - `force_pqc`: only accept main TLS connections negotiated with a PQC- compatible key exchange group.
  - `admin_force_pqc`: only accept admin TLS connections negotiated with a PQC- compatible key exchange group.
  - `replication_force_pqc`: require asynchronous replication TLS connections to use a PQC-compatible key exchange group.
  - `group_replication_force_pqc`: require Group Replication recovery TLS connections to use a PQC-compatible key exchange group.
  - `mysqlx_force_pqc`: only accept X Plugin TLS connections negotiated with a PQC-compatible key exchange group.
  - `tls_kex`: TLS key exchange groups to use for the main connection channel.
  - `admin_tls_kex`: TLS key exchange groups to use for the admin connection channel.
  - `replication_tls_kex`: TLS key exchange groups to use for asynchronous replication TLS connections.
  - `group_replication_tls_kex`: TLS key exchange groups to use for Group Replication recovery TLS connections.
  - `mysqlx_tls_kex`: TLS key exchange groups to use for X Plugin TLS connections.
  - The following boolean system variables define which algorithm is advertised. If enabled, PQC is used. If disabled, only classical TLS handshake signature algorithms are advertised.

    Note

    If enabled, but PQC is not available, the OpenSSL default is used.

    - `use_pqc_sign`: which algorithm is advertised on the main connection channel.

- `admin_use_pqc_sign`: which algorithm is advertised on administration connections.
- `replication_use_pqc_sign`: which algorithm is advertised on asynchronous replication connections.
- `group_replication_use_pqc_sign`: which algorithm is advertised on Group Replication recovery connections.
- `mysqlx_use_pqc_sign`: which algorithm is advertised on X Plugin connections.

The following status variables are introduced:

- `Tls_key_exchange_algorithm`: indicates the TLS key exchange group negotiated for the current TLS connection.
- `Tls_sign_algorithm`: indicates the TLS handshake signature algorithm negotiated for the current TLS connection.
- `Mysqlx_force_pqc`: value of the effective configuration applied to the current X plugin TLS channel.
- `Mysqlx_tls_kex`: value of the effective configuration applied to the current X plugin TLS channel.
- `Mysqlx_use_pqc_sign`: value of the effective configuration applied to the current X plugin TLS channel.

(Bug #39536947, WL #17245)

Thread Pool Notes

- The default value of `thread_pool_max_unused_threads` is changed from 2 to 32. (Bug #39405207)
- MySQL Thread Pool plugin, formerly only available in MySQL Enterprise Edition, is now available in MySQL Community Edition 26.7.0. (WL #17296)

X Plugin Notes

- Fixed several issues in the X Plugin. (Bug #39268692, Bug #39268863)

Functionality Added or Changed

- **Important Change:** MySQL Server now uses calendar-based versioning for releases after the 9.7 LTS series. Version values use Year.Month.Patch (YY.M.P) format. MySQL 26.7.0 is the first such calendar-version release.

  The following are added to the `mysql_version.h` to identify the previous LTS version:

  - `MYSQL_PREVIOUS_LTS_VERSION`: The previous LTS version number. For example, `"9.7.0"` is the previous LTS version for MySQL 26.7.0.
  - `MYSQL_PREVIOUS_LTS_VERSION_ID`: The identifier of the previous LTS version. For example, `"9.7.0"` is represented as `90700`.

  The versioning lineage defined in `mysql_version.h` enables clear upgrade and downgrade paths.

  See MySQL Releases: Innovation and LTS, Upgrade Paths, and Downgrading MySQL. (WL #17310)

- MySQL upgrade performance is improved by streamlining the `ALTER TABLE` statements which run on mysql.general_log and mysql.slow_log.

Our thanks to Karry Zhang and the team at Alibaba for the contribution. (Bug #118132, Bug #37944523)

Bugs Fixed

- Fixed an issue in Spatial Data handling. (Bug #39489004)
- Fixed several issues in the Clone plugin. (Bug #39252316, Bug #39245805, Bug #39245844, Bug #39253040, Bug #39253103)
- It was not possible to start the server on Oracle Linux or Red Hat Linux 9 if `innodb_redo_log_encrypt=ON` was defined in the configuration file. (Bug #39181231)
- Fixed an issue relating to connection-attribute parsing. (Bug #39116965)
- An assertion failure could be returned for an UPDATE statement assigning the result of an IN subquery, if the subquery used a derived table with UNION. (Bug #38911659)
- Some Polish characters were replaced by question marks (?) in client messages. (Bug #38036138)
- Fixed an issue relating to table maintenance statement error handling after table and metadata-lock cleanup.

  Our thanks to Yin Peng and the team from Tencent for the contribution. (Bug #37797437)
- Deleting a row from a table with a self-referencing foreign key caused replication to fail. (Bug #119996, Bug #39091367)
- An `INSERT ... SELECT` through a view with `ON DUPLICATE KEY UPDATE` could cause an assertion failure. (Bug #119805, Bug #38901085)
- Server upgrade checks could fail when validating events created under a different `sql_mode` than the one used in the target version, causing the upgrade to abort although the event was valid in its original context. (Bug #119672, Bug #39077967)
- During LOAD DATA, fields can be marked temporarily nullable while input rows are parsed. If a row supplies NULL for a NOT NULL column, LOAD DATA could leave temporary NULL state on the table fields before returning to the same thread. A later INSERT in that same session can then treat valid non-NULL values as NULL and return ER_BAD_NULL_ERROR.

  Our thanks to Feng Chuanheng for the contribution. (Bug #118060, Bug #38939847)
- After `LOAD DATA` failed because a row supplied `NULL` for a `NOT NULL` column, a subsequent `INSERT` containing valid values in the same session could be rejected with `ER_BAD_NULL_ERROR`. Errors were returned similar to the following:

  ```
  ERROR 1263 (22004): Column set to default value; NULL supplied to NOT NULL column 'b' at row
  ERROR 1048 (23000): Column 'b' cannot be null
  ```

  Our thanks to Feng Chuanheng for the contribution. (Bug #118060, Bug #38939847)
- The variable has_unique_key was assigned unnecessarily.

  Our thanks to Alex Xing for the contribution. (Bug #118057, Bug #37876712)
- Upgrading from older 8.x releases with thousands of tables, views, routines, and events caused the memory consumed by server to grow continuously, leading to significant memory spikes.

  Memory management is improved for these scenarios. (Bug #117983, Bug #37832455, Bug #38031020)

- NOW() accepted values outside the defined range (0-6) and did not return an error. (Bug #116307, Bug #37141651)
- Rewriter patterns containing trailing semicolons did not load and match consistently with query digests produced by normal statement execution.

  Our thanks to Kaiwang Chen and the team at Tencent for the contribution. (Bug #116222, Bug #37100271)
- `key_object_cache_mutex` and `key_object_loading_cond` were not registered.

  Our thanks to Huaxiong Song for the contribution. (Bug #115524, Bug #36809659)
- The following were not properly defined as reserved keywords:

  - `CUBE`
  - `EXTERNAL`
  - `QUALIFY`
  - `TABLESAMPLE`

  The following were not properly defined as non-reserved keywords:

  - `MANUAL`
  - `PARALLEL`

  (Bug #114874, Bug #36584265)
- The behavior of `INSERT()` was incorrect for multibyte characters. It inserted characters even if the given position was beyond the input string.

  Our thanks to Dirkjan Bussink for the contribution. (Bug #113962, Bug #36289412)
- Under certain circumstances, `LEAST()` and `GREATEST()` could return unexpected results.

  Our thanks to Jinsheng Ba for the contribution. (Bug #113304, Bug #36065440)
- Fixed a memory leak in `statement_digest()` and `statement_digest_text()`. (Bug #104115, Bug #33073320)