MySQL 8.1 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL 8.1, up through MySQL 8.1.0. For information about changes in a different MySQL series, see the release notes for that series.

For additional MySQL 8.1 documentation, see the MySQL 8.1 Reference Manual, which includes an overview of features added in MySQL 8.1 (What Is New in MySQL 8.1), and discussion of upgrade issues that you may encounter for upgrades from MySQL 8.0 to MySQL 8.1 (Changes in MySQL 8.1).

Before upgrading to MySQL 8.1, review the information in https://dev.mysql.com/doc/refman/8.1/en/upgrading.html and perform any recommended actions. Perform the upgrade on a test system first to make sure everything works smoothly, and then on the production system.

Downgrade from MySQL 8.1 to MySQL 8.0, or from a MySQL 8.1 release to a previous MySQL 8.1 release, is not supported. The only supported alternative is to restore a backup taken before upgrading. It is therefore imperative that you back up your data before starting the upgrade process.

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: 2023-09-27 (revision: 27287)

Table of Contents

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL 8.1, up through MySQL 8.1.0.

Legal Notices

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:

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 8.1.0 (2023-07-18, Innovation Release)

Account Management Notes

- A new password-validation system variable now permits the configuration and enforcement of a minimum number of characters that users must change when attempting to replace their own MySQL account passwords. This new verification setting is a percentage of the total characters in the current password. For example, if `validate_password.changed_characters_percentage` has a value of 50, at least half of the characters in the replacement account password must not be present in the current password, or the password is rejected.

  This new capability is one several that provide DBAs more complete control over password management. For more information, see Password Management. (WL #15751)

Audit Log Notes

- In MySQL 8.0.33, the `audit_log` plugin added support for choosing which database to use to store JSON filter tables. It is now possible to specify an alternative to the default system database, `mysql`, when running the plugin installation script. Use the `audit_log_database` server system variable (or `-D database_name`) on the command line together with the alternative database name, for example:

  ```
  $> mysql -u root -D database_name -p < audit_log_filter_linux_install.sql
  ```

  For additional information about using `audit_log` plugin installation scripts, see Installing or Uninstalling MySQL Enterprise Audit. (Bug #35252268)

- The new `Audit_log_direct_writes` system variable is added to count direct writes into the audit file.

- MySQL Enterprise Audit allocates a temporary buffer to hold data that forms a single event written into the log file. The audit plugin buffers every query that arrives into the audit log. While this is effective for short queries, the server is not always capable of allocating extra memory to hold a long query. Now, the `audit_log` plugin is optimized not to use a temporary buffer when JSON-format logging is used. (WL #15403)

- MySQL Enterprise Audit now supports using the scheduler component to configure and execute a recurring task to flush the in-memory cache. For setup instructions, see Enabling the Audit Log Flush Task. (WL #15567)

Binary Logging

- Several functions now are added to the `libmysqlclient.so` shared library that enable developers to access a MySQL server binary log: `mysql_binlog_open()`, `mysql_binlog_fetch()`, and `mysql_binlog_close()`.

  Our thanks to Yura Sorokin for the contribution. (Bug #110658, Bug #35282154)

C API Notes

- Added the new `mysql_reset_connection_nonblocking()` C API function. It is the counterpart of the `mysql_reset_connection()` synchronous function, for use by applications that require asynchronous communication with the server. Our thanks to Meta for the contribution. (Bug #32202058, WL #15633)

- The new `mysql_get_connect_nonblocking_stage()` C API function permits applications to monitor the progress of asynchronous connections for the purpose of taking appropriate actions based on the progress. Our thanks to Meta for the contribution. (Bug #32202053, WL #15651)

- In the calling function, `len` is initialized to 0 and never changed if `net->vio` is null. This fix adds a check of `net` before dereferencing `vio`.

  Our thanks to Meta for the contribution. (Bug #30809590)

- A variable in the `async` client was uninitialized in certain code paths. It is fixed by always initializing the variable.

  Our thanks to Meta for the contribution. (Bug #30809590)

Compilation Notes

- **Microsoft Windows:** For Windows, improved `MSVC_CPPCHECK` support; and check for MSVC warnings similar to "maintainer" mode. For example, check after all third party configurations are complete. (Bug #35283401)

  References: See also: Bug #34828882.

- **Microsoft Windows:** For Windows builds, improved WIN_DEBUG_NO_INLINE=1 support; usage would exceed the library limit of 65535 objects. (Bug #35259704)
- Upgraded the bundled robin-hood-hashing from v3.8.1 to v3.11.5. (Bug #35448980)
- Removed the unused `extra/libcbor/doc/` directory as `extra/libcbor/doc/source/requirements.txt` inspired bogus pull requests on GitHub. (Bug #35433370)
- Updated the bundled ICU files from version 69.1 to version 73 for the `icu-data-files` package. (Bug #35353708)
- ZSTD sources bundled in the source tree were upgraded to ZSTD 1.5.5 from 1.5.0. (Bug #35353698)
- For SUSE-based systems, changed the default GCC version from version 9 to 12; which is the default compiler on these platforms. (Bug #35341000)
- MySQL did not compile correctly with GCC 12. (Bug #35327995)
- Initialize the internal `MEM_ROOT` class memory with garbage using the TRASH macro to make easier to reproduce bugs caused by reading initialized memory allocated from MEM_ROOT. (Bug #35277644)
- Fixed ODR violations due to multiple different instances of YYSTYPE and other symbols generated by Bison. This includes Bison implementation changes, such as replacing the --name-prefix argument on the Bison command line with api.prefix definitions. (Bug #35232738)
- We now determine stack direction at runtime rather than at configure time. (Bug #35181008)
- Added the `OPTIMIZE_SANITIZER_BUILDS` CMake option that adds `-O1 -fno-inline` to sanitizer builds. It defaults to ON. (Bug #35158758)
- Changed the minimum Bison version requirement from v2.1 to v3.0.4. For macOS, this may require installing Bison via a package manager such as Homebrew. (Bug #35154645, Bug #35191333)
- On Windows, the default for the `MSVC_CPPCHECK` CMake option has changed from OFF to ON. (Bug #35067705)
- MySQL now sets LANG=C in the environment when executing `readelf` to avoid problems with non-ASCII output.

  Our thanks to Kento Takeuchi for the contribution. (Bug #111190, Bug #35442825)
- On macOS, MySQL would not compile if rapidjson was installed via Homebrew. The workaround was to `brew unlink rapidjson`. (Bug #110736, Bug #35311140)

  References: This issue is a regression of: Bug #35006191.
- MySQL would not build with `-DWITH_ZLIB=system`; it'd complain about not finding the system zlib library despite finding it. (Bug #110727, Bug #110745, Bug #35307674, Bug #35312227)

Component Notes

- MySQL Enterprise Edition now supports collecting server trace data in the OpenTelemetry format using the `component_telemetry` component. This data is then forwarded to a configurable endpoint where it can be used by any OpenTelemetry-compatible system.

  

  Note

  Telemetry traces are supported by MySQL Enterprise Edition on Linux platforms only.

  See Telemetry. (WL #15198)

Deprecation and Removal Notes

- **Important Change:** Since MySQL provides other means of performing database dumps and backups with the same or additional functionality, including `mysqldump` and MySQL Shell Utilities, the `mysqlpump` client utility program has become redundant, and is now deprecated. Invocation of this program now produces a warning. You should keep in mind that `mysqlpump` is subject to removal in a future version of MySQL, and move applications depending on it to another solution, such as those mentioned previously. (WL #15652)

- **Replication:** The `sync_relay_log_info` server system variable is deprecated in this release, and getting or setting this variable or its equivalent startup option `--sync-relay-log-info` now raises a warning.

  Expect this variable to be removed in a future version of MySQL; applications which make use of it should be rewritten not to depend on it before this happens. (Bug #35367005, WL #13968)

- **Replication:** The `binlog_format` server system variable is now deprecated, and subject to removal in a future version of MySQL. The functionality associated with this variable, that of changing the binary logging format, is also deprecated.

  The implication of this change is that, when `binlog_format` is removed, only row-based binary logging, already the default in MySQL 8.0, will be supported by the MySQL server. For this reason, new installations should use only row-based binary logging, and existing ones using the statement-based or mixed logging format should be migrated to the row-based format. See Replication Formats, for more information.

  The system variables `log_bin_trust_function_creators` and `log_statements_unsafe_for_binlog`, being useful only in the context of statement-based logging, are now also deprecated, and are thus also subject to removal in a future release of MySQL.

  Setting or selecting the values of any of the variables just mentioned now raises a warning. (WL #13966, WL #15669)

- **Group Replication:** The `group_replication_recovery_complete_at` server system variable is now deprecated, and setting it produces a warning. You should expect its removal in a future release of MySQL. (WL #15460)

- The `mysql_native_password` authentication plugin now is deprecated and subject to removal in a future version of MySQL. `CREATE USER`, `ALTER USER`, and `SET PASSWORD` operations now insert a deprecation warning into the server error log if an account attempts to authenticate using `mysql_native_password` as an authentication method. (Bug #35336317)

- Previously, if the `audit_log` plugin was installed without the accompanying audit tables and functions needed for rule-based filtering, the plugin operated in legacy filtering mode. Now, legacy filtering mode is deprecated. New deprecation warnings are emitted for legacy audit log filtering system variables. These deprecated variables are either read-only or dynamic.

  (Read-only) `audit_log_policy` now writes a warning message to the MySQL server error log during server startup when the value is not `ALL` (default value).

  (Dynamic) `audit_log_include_accounts`, `audit_log_exclude_accounts`, `audit_log_statement_policy`, and `audit_log_connection_policy`. Dynamic variables print a warning message based on usage:

  - Passing in a non-NULL value to `audit_log_include_accounts` or `audit_log_exclude_accounts` during MySQL server startup now writes a warning message to the server error log.

  - Passing in a non-default value to `audit_log_statement_policy` or `audit_log_connection_policy` during MySQL server startup now writes a warning message to the server error log. `ALL` is the default value for both variables.

  - Changing an existing value using `SET` syntax during a MySQL client session now writes a warning message to the client log.

  - Persisting a variable using `SET PERSIST` syntax during a MySQL client session now writes a warning message to the client log.

  (WL #11248)

- The use of the dollar sign (`$`) as the initial character of an unquoted identifier was deprecated in MySQL 8.0.32. In this release, the use of an unquoted identifier starting with the dollar sign and containing one or more dollar signs in addition to the first one generates a syntax error. Quoted identifiers, and unquoted identifiers that start with a dollar sign but contain no additional occurrences of this character, are not affected by this change. Use of an unquoted identifier with a leading dollar sign that is otherwise permitted continues to raise a warning.

  For more information, see Schema Object Names. (WL #15254)

  References: See also: Bug #34684193.

- MySQL enables control of FIPS mode on the server side and the client side using a system variable and client option. Application programs can use the `MYSQL_OPT_SSL_FIPS_MODE` option to `mysql_options()` to enable FIPS mode on the client. Alternatively, it is possible to handle FIPS mode directly through OpenSSL configuration files rather than using the current server-side system variable and client-side options. When MySQL is compiled using OpenSSL 3.0, and an OpenSSL library and FIPS Object Module are available at runtime, the server reads the OpenSSL configuration file and respects the preference to use a FIPS provider, if one is set. OpenSSL 3.0 is certified for use with FIPS.

  To favor the OpenSSL alternative, the `ssl_fips_mode` server system variable, `--ssl-fips-mode` client option, and the `MYSQL_OPT_SSL_FIPS_MODE` option now are deprecated and subject to removal in a future version of MySQL. A deprecation warning prints to standard error output when an application uses the `MYSQL_OPT_SSL_FIPS_MODE` option or when a client user specifies the `--ssl-fips-mode` option on the command line, through option files, or both.

  Prior to being deprecated, the `ssl_fips_mode` server-side system variable was dynamically settable. It is now a read-only variable (accepts `SET PERSIST_ONLY`, but not `SET PERSIST` or `SET GLOBAL`). When specified on the command line or in the `mysqld-auto.cnf` option file (with `SET PERSIST_ONLY`) a deprecation warning prints to the server error log. (WL #15631)

- The `mysql_ssl_rsa_setup` program originally provided a simple way for community users to generate certificates manually, if OpenSSL was installed on the system. Now, `mysql_ssl_rsa_setup` is deprecated because MySQL Community Edition no longer supports using yaSSL as the SSL library, and source distributions no longer include yaSSL. Instead, use MySQL server to generate missing SSL and RSA files automatically at startup (see Automatic SSL and RSA File Generation). (WL #15668)

- The `keyring_file` and `keyring_encrypted_file` plugins now are deprecated. These keyring plugins are superseded by the `component_keyring_file` and `component_keyring_encrypted_file` components. For a concise comparison of keyring components and plugins, see Keyring Components Versus Keyring Plugins. (WL #15659)

- Previously, the MySQL server processed a version-specific comment without regard as to whether any whitespace followed the MySQL version number contained within it. For example, the comments `/*!80034KEY_BLOCK_SIZE=1024*/` and `/*!80034 KEY_BLOCK_SIZE=1024*/` were handled identically. Beginning with this release, when the next character following the version number in such a comment is neither a whitespace character nor the end of the comment, the server issues a warning: `Immediately starting the version comment after the version number`

`is deprecated and may change behavior in a future release. Please insert a whitespace character after the version number`.

You should expect the whitespace requirement for version-specific comments to become strictly enforced in a future version of MySQL.

See Comments, for more information. (WL #15686)

- The MySQL client library currently supports performing an automatic reconnection to the server if it finds that the connection is down and an application attempts to send a statement to the server to be executed. Now, this feature is deprecated and subject to removal in a future release of MySQL.

  The related `MYSQL_OPT_RECONNECT` option is still available but it is also deprecated. C API functions `mysql_get_option()` and `mysql_options()` now write a deprecation warning to the standard error output when an application specifies `MYSQL_OPT_RECONNECT`. (WL #15766)

Logging Notes

- To aid in troubleshooting in the event of an excessively long server shutdown, this release introduces a number of new messages that are written to the MySQL error log during shutdown, including those listed here:

  - Startup and shutdown log messages for the MySQL server, including when it has been started with `--initialize`
  - Log messages showing start and end of shutdown phases for plugins
  - Log messages showing start and end of shutdown phases for components
  - Start-of-phase and end-of-phase log messages for connection closing phases
  - Log messages showing the number and IDs of threads still alive after being forceably disconnected, and potentially causing a wait

  See The Error Log, for more information. (WL #15369)

Performance Schema Notes

- The type used for the Performance Schema `clone_status` table's `gtid_executed` column has been changed from `VARCHAR(4096)` to `LONGTEXT`. (Bug #109171, Bug #34828542)

Spatial Data Support

- The EPSG data set containing spatial reference system data for spatial calculations has been upgraded from version 9.3 to version 9.7. (Bug #28615740)

SQL Syntax Notes

- **JSON:** It is now possible to capture `EXPLAIN FORMAT=JSON` output in a user variable using a syntax extension added in this release. `EXPLAIN FORMAT=JSON INTO var_name stmt` works with any explainable statement *`stmt`* to store the output in the user variable *`var_name`*, where it can be retrieved for later use in analysis. This value is a valid JSON document and can be inspected and manipulated with MySQL JSON functions such as `JSON_EXTRACT()`. (See JSON Functions.)

  The `INTO` clause is supported only with `FORMAT=JSON`; the value of the `explain_format` system variable has no effect on this requirement. If the statement cannot be executed (due to, for instance, a syntax error), the user variable is not updated.

  `INTO` is not supported for `EXPLAIN ANALYZE` or `EXPLAIN FOR CONNECTION`.

For additional information and examples, see Obtaining Execution Plan Information. (Bug #35362996, WL #15588, WL #15606)

- `CURRENT_USER()` can now be used as a default value for `VARCHAR` and `TEXT` columns in `CREATE TABLE` and `ALTER TABLE ... ADD COLUMN` statements.

  The functions `SESSION_USER()`, `USER()`, and `SYSTEM_USER()` are also supported in all of the cases just mentioned. By way of example, the following sequence of statements now works similarly to what is shown here, with the precise output dependent on your environment:

```
mysql> SELECT CURRENT_USER();
+-------------------+
| CURRENT_USER()    |
+-------------------+
|  sakila@localhost |
+-------------------+
1 row in set (0.00 sec)

mysql> CREATE TABLE t (
     >  c1 VARCHAR(288) DEFAULT (USER()),
     >  c2 VARCHAR(288) DEFAULT (CURRENT_USER()),
     >  c3 VARCHAR(288) DEFAULT (SESSION_USER()),
     >  c4 VARCHAR(288) DEFAULT (SYSTEM_USER())
     > );
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO t VALUES ROW();
Query OK, 1 row affected (0.01 sec)

mysql> TABLE t;
+-------------------+-------------------+-------------------+-------------------+
| c1                | c2                | c3                | c4                |
+-------------------+-------------------+-------------------+-------------------+
|  sakila@localhost |  sakila@localhost |  sakila@localhost |  sakila@localhost |
+-------------------+-------------------+-------------------+-------------------+
1 row in set (0.00 sec)
```

  When used in this way, these functions are also included in the output of `SHOW CREATE TABLE` and `SHOW COLUMNS`, and referenced in the `COLUMN_DEFAULT` column of the Information Schema `COLUMNS` table where applicable.

  If you need to insure that values having the maximum possible length can be stored in such a column, you should make sure that the column can accommodate at least 288 characters (255 for the user name and 32 for the host name, plus 1 for the separator `@`). For this reason—while it is possible to use one of these functions as the default for a `CHAR` column, it is not recommended due to the risk of errors or truncation of values. (Bug #17809, Bug #11745618)

Functionality Added or Changed

- **Important Change; Replication:** The default value for the `SOURCE_RETRY_COUNT` option of the `CHANGE REPLICATION SOURCE TO` statement has been changed to 10. This means that, using the default values for this option and for `SOURCE_CONNECT_RETRY` (60), the replica waits 60 seconds between reconnection attempts, and keeps attempting to reconnect at this rate for 10 minutes before timing out and failing over.

  This change also applies to the default value of the `--master-retry-count` server option. You should note that this option is deprecated and therefore subject to removal in a future MySQL release. Use `SOURCE_RETRY_COUNT` with `CHANGE REPLICATION SOURCE TO`, instead.

  See CHANGE REPLICATION SOURCE TO Statement, as well as Asynchronous Connection Failover for Sources, for further information. (WL #15702)

- **Important Change:** For platforms on which OpenSSL libraries are bundled, the linked OpenSSL library for MySQL Server has been updated from OpenSSL 1.1.1 to OpenSSL 3.0. The exact

version is now 3.0.9. More information on changes from 1.1.1 to 3.0 can be found at https://www.openssl.org/docs/man3.0/man7/migration_guide.html. (Bug #35475140, WL #15614)

- **Important Change:** MySQL version numbers used in versioned comments now support a major version consisting of one or two digits (previously, only a single digit was supported for this value). See Comments, for more information about how this change affects handling of version-specific comments in MySQL. (WL #15687)

- **Important Change:** Dropped support for Enterprise Linux 6 (and associated glibc 2.12 generic), SUSE 12, Debian 10, MacOS 12, Ubuntu 18.04 and 20.04, Windows 10 and Server 2012R2; and 32-bit versions are no longer built.

- **Replication:** When running in debug mode, `mysqlbinlog` now prints all `Rows_log_event` flags (and not only `STMT_END_F`), and now asserts with `UNKNOWN_FLAG(0xN)` if it encounters an invalid flag.

  Our thanks to Meta for this contribution. (Bug #33172581)

- **Group Replication:** Any statement that fetches values of system status variables fetches them all, and acquires a read lock on them all as well. This includes statements such as `SHOW STATUS LIKE 'Uptime'` and `SELECT * FROM performance_schema.global_status WHERE VARIABLE_NAME='Uptime'`. In addition, the following operations all acquire a write lock on the status variables:

  - `START GROUP_REPLICATION` and `STOP GROUP_REPLICATION` statements
  - Setting `group_replication_force_members` or `group_replication_message_cache_size`
  - Invoking `group_replication_get_write_concurrency()` or `group_replication_set_communication_protocol()`
  - Automatic rejoin
  - Change of primary with `group_replication_single_primary_mode` enabled

  This meant that a `SHOW STATUS` started after one of the operations just listed was required to wait until the operation was complete before returning.

  Now in such cases, the statement fetching status variables immediately returns their cached values instead of waiting. (Bug #35373030)

  References: See also: Bug #35312441.

- **Group Replication:** Before it elects a new primary, `group_replication_set_as_primary()` waits for all transactions to finish, including all DML operations that are currently being processed. In

this release, this function now also waits for all ongoing DDL statements, such as `ALTER TABLE`, to complete as well.

Listed here are all operations now considered to be DDL statements by `group_replication_set_as_primary()`:

- `ALTER TABLE`
- `ANALYZE TABLE`
- `CACHE INDEX`
- `CHECK TABLE`
- `CREATE INDEX`
- `CREATE TABLE`
- `DROP INDEX`
- `LOAD INDEX`
- `OPTIMIZE TABLE`
- `REPAIR TABLE`
- `TRUNCATE TABLE`
- `DROP TABLE`

This also includes any open cursors (see Cursors).

For more information, see the description of the `group_replication_set_as_primary()` function, in the MySQL 8.1 Manual. (Bug #34664197, WL #15497)

- **Group Replication:** For better diagnosis and troubleshooting of network instabilities, MySQL Group replication adds a number of variables in this release providing network, control message, and data message statistics for each member of Group Replication. This makes it possible to observe directly the time spent in each of several steps in Group Replication operations.

  Group Replication also adds a new `MEMBER_FAILURE_SUSPICIONS_COUNT` column to the Performance Schema `replication_group_communication_information` table, which shows how many times each group member has been seen as suspect by the local node. For example, in a group with three members, the value of this column should look something like this:

  ```
  {
    "d57da302-e404-4395-83b5-ff7cf9b7e055": 0,
    "6ace9d39-a093-4fe0-b24d-bacbaa34c339": 10,
    "9689c7c5-c71c-402a-a3a1-2f57bfc2ca62": 0
  }
  ```

  These enhancements also help pinpoint how much time and network resources are consumed by user-initiated or background operations, which can then be correlated with overall performance.

  See Group Replication Status Variables, for more information. (WL #13849)

  References: See also: Bug #34279841.

- Binary packages that include curl rather than linking to the system curl library have been upgraded to use curl 8.1.2. (Bug #35329529)

- MySQL now implements client-side Server Name Indication (SNI), which is an extension to the TLS protocol. Client applications can pass a server name to the `libmysqlclient` C API library with the

new `MYSQL_OPT_TLS_SNI_SERVERNAME` option for mysql_options(). Similarly, each MySQL client program now includes a `--tls-sni-servername` command option to pass in a name. The new `Tls_sni_server_name` server status variable indicates the name if one is set for the session. Our thanks to Meta for the contribution. (Bug #33176362, WL #14839)

- Comments in the `mysql` client are now enabled by default. To disable them, start `mysql` with the `--skip-comments` option.

  Our thanks to Daniël van Eeden for the contribution. (Bug #109972, Bug #35061087, WL #15597)

- Implemented a `SHOW PARSE_TREE` statement in debug builds to display the JSON-formatted parse tree for a `SELECT` statement. This statement is not supported in release builds, and is available only in debug builds, or by compiling the server using `-DWITH_SHOW_PARSE_TREE`. (WL #15426)

- Previously, invalid SSL server and CA certificates were not identified as problematic until after the server started or after an invalid certificate was loaded at runtime. Now, the new `tls-certificates-enforced-validation` system variable permits a DBA to enforce certificate validation at server startup or when using the `ALTER INSTANCE RELOAD TLS` statement to reload certificates at runtime. With enforcement enabled, discovering an invalid certificate halts server invocation at startup, prevents loading invalid certificates at runtime, and emits warnings. For more information, see Configuring Certificate Validation Enforcement. (WL #13470)

- New server system variables now control the amount of time MySQL accounts that connect to a MySQL server using LDAP pluggable authentication must wait when the LDAP server is down or unresponsive. The default timeout is 30 seconds for the following simple and SASL-based LDAP authentication variables:

  - `authentication_ldap_simple_connect_timeout`
  - `authentication_ldap_simple_response_timeout`
  - `authentication_ldap_sasl_connect_timeout`
  - `authentication_ldap_sasl_response_timeout`

  Connection and response timeouts are configurable through the system variables on Linux platforms only. For more information, see Setting Timeouts for LDAP Pluggable Authentication. (WL #14757)

- Previously, MySQL Server generated and emitted activity-monitoring events through plugin APIs. Now, the server emits events using component APIs. At the same time, to provide backward compatibility with plugins that use audit plugin APIs (such as `audit_log`, `MYSQL_FIREWALL`, `CONNECTION_CONTROL`, `Rewriter`, and so on), the server also implements an intermediate layer that generates required events through plugin APIs. Some of the related error messages may have an `EVENT_TRACKING_` prefix, rather than the current `MYSQL_AUDIT_` prefix. (WL #12652)

Bugs Fixed

- **Incompatible Change; Replication:** Setting server variables equal to SQL `NULL` as options on the command line should not be possible and is not supported. Beginning with this release, setting any of these to `NULL` is disallowed, and attempting to do is rejected with an error.

  The following variables are excepted from this restriction: `admin_ssl_ca`, `admin_ssl_capath`, `admin_ssl_cert`, `admin_ssl_cipher`, `admin_tls_ciphersuites`, `admin_ssl_key`, `admin_ssl_crl`, `admin_ssl_crlpath`, `basedir`, `character_sets_dir`, `ft_stopword_file`, `group_replication_recovery_tls_ciphersuites`, `init_file`, `lc_messages_dir`, `plugin_dir`, `relay_log`, `relay_log_info_file`, `replica_load_tmpdir`, `ssl_ca`, `ssl_capath`, `ssl_cert`, `ssl_cipher`, `ssl_crl`, `ssl_crlpath`, `ssl_key`, `socket`, `tls_ciphersuites`, and `tmpdir`.

  See Server System Variables, for more information. (Bug #109387, Bug #34897517)

- **Important Change:** The default value of the `connection_memory_chunk_size` server system variable, when introduced in MySQL 8.0.28, was mistakenly set at 8912. This fix changes the default to 8192, which is the value originally intended. (Bug #35218020)

- **NDB Cluster:** The fix for a previous issue introduced a slight possibility of unequal string values comparing as equal, if any Unicode 9.0 collations were in use, and the collation hash methods calculated identical hash keys for two unequal strings. (Bug #35168702)

  References: See also: Bug #27522732. This issue is a regression of: Bug #30884622.

- **InnoDB:** An error due to the bulk load of data that is larger than the InnoDB page size has been fixed. (Bug #35332046, Bug #110813)

- **InnoDB:** Possible congestion due to purging a large number of system threads has been fixed. (Bug #35289390, Bug #110685)

- **InnoDB:** Errors when `innodb_thread_concurrency` set to 999 have been fixed. (Bug #34925101)

- **InnoDB:** Performance regression due to hash function changes in MySQL 8.0.30 have been fixed. (Bug #34870256)

- **InnoDB:** Errors that can occur with the character sets `ucs2`, `utf16`, and `utf32` have been fixed. (Bug #34790366)

- **InnoDB:** The rules for aggregating entries in the redo log have been fixed. (Bug #34752625, Bug #108944)

- **InnoDB:** Contradictory warning and error messages for recovery in read-only mode when the redo log is not empty have been fixed. (Bug #34506094, Bug #108177)

- **InnoDB:** Several errors due to tablespace deletion and the buffer pool have been fixed. (Bug #34330475, Bug #107689)

- **InnoDB:** An error due to multiplication in `ibd2sdi` has been fixed. (Bug #33172685, Bug #104474)

- **InnoDB:** Errors that can cause buffer pool exhaustion have been fixed. (Bug #27238364)

- **Packaging; Group Replication:** The group replication plugin from the Generic Linux packages did not load on some platforms that lacked a compatible version of `tirpc`. (Bug #35323208)

- **Replication:** Changes in `session_track_gtids` were not always propagated correctly. (Bug #35401212)

- **Replication:** By design, all DDL operations (including binary log operations such as purging the binary log) acquire a shared lock on the `BACKUP_LOCK` object, which helps to prevent simultaneous backup and DDL operations. For binary log operations, we checked whether any locks existed on `BACKUP_LOCK` but did not check the types of any such locks. This caused problems due to the fact that binary log operations should be prevented only when an exclusive lock is held on the `BACKUP_LOCK` object, that is, only when a backup is actually in progress, and backups should be prevented when purging the binary log.

  Now in such cases, instead of checking for locks held on the `BACKUP_LOCK` object, we acquire a shared lock on `BACKUP_LOCK` while purging the binary log. (Bug #35342521)

- **Replication:** In all cases except one, when `mysqlbinlog` encountered an error while reading an event, it wrote an error message and returned a nonzero exit code, the exception being for the active binary log file (or any binary log where the format_description_log_event had the

`LOG_EVENT_BINLOG_IN_USE_F` flag set), in which case it did not write a message, and returned exit code 0, thus hiding the error.

Now `mysqlbinlog` suppresses only those errors which are related to truncated events, and when doing so, prints a comment rather than an error message. This fix also improves the help text for the `--force-if-open` option. (Bug #35083373)

- **Replication:** Compressed binary log event handling was improved. (Bug #33666652)
- **Replication:** A transaction consisting of events each smaller than 1 GiB, but whose total size was larger than 1 GiB, and where compression did not make it smaller than 1 GiB, was still written to the binary log as one event bigger than 1 GiB. This made the binary log unusable; in effect, it was corrupted since neither the server nor other tools such as `mysqlbinlog` could read it.

  Now, when the compressed data grows larger than 1 GiB, we fall back to processing the transaction without any compression. (Bug #33588473)
- **Replication:** The multithreaded applier wrote messages similar to `Multi-threaded slave: Coordinator has waited 312251 times hitting slave_pending_jobs_size_max; current event size = 8176` into the error log, although they did not belong there. (Bug #32587480)
- **Replication:** Executing either of the statements `FLUSH BINARY LOGS` or `SET GLOBAL binlog_checksum = CRC32` after setting the session transaction access mode to `READ ONLY` led to an unplanned shutdown. Execution of either of these statements causes rotation of the binary log; before doing so, it is necessary to update the `mysql.gtid_executed` table, but this was rejected due to the session transaction access mode being `READ ONLY`.

  We fix this by allowing the binary log rotation to proceed by ignoring `READ ONLY` access mode, as when the server is running in read-only or super-read-only mode. (Bug #109894, Bug #35041573)
- **Group Replication:** In a group replication setup, when there was a source of transactions other than the applier channel, the following sequence of events was possible:

  1. Several transactions being applied locally were already certified, and so were associated with a ticket, which we refer to as Ticket 2, but had not yet been committed. These could be local or nonlocal transactions.
  2. A view is created with Ticket 3, and must wait on transactions from Ticket 2.
  3. The view change (VC1) entered the GR applier channel applier and waited for the ticket to change to 3.
  4. Another group change, and another view change (VC2), occurred while the transactions from Ticket 2 were still completing.

  This gave rise to the following issue: There was a window wherein the last transaction from Ticket 2 had already marked itself as being executed but had not yet popped the ticket; VC2 popped the ticket instead but never notified any of the participants. This meant that VC1 continued to wait indefinitely for the ticket to change, and with the additional effect that the worker could not be killed.

  We fix this by checking for the need to break each second so that this loop is responsive to changes in the loop condition; we also register a new stage, so that the loop is more responsive to kill signals. (Bug #35392640)

  References: See also: Bug #35206392, Bug #35374425.
- **Group Replication:** Executing `SET GLOBAL group_replication_force_members = host:port` and `SHOW STATUS LIKE 'group_replication_primary_member'` on the host in parallel sometimes led to a timeout while waiting for a new view. (Bug #35312441)

- **Group Replication:** Removed a memory leak discovered in `Network_provider_manager::open_xcom_connection()`. (Bug #34991101)

- **Group Replication:** When a group action was sent to the group and the connection was killed on the coordinator, group members were in different states, with members which received the coordinated action waiting for the member that executed it, and the member which started execution having nothing to process, which caused problems with coordination of the group.

  Now in such cases, we prevent this issue from occurring by causing group actions to wait until all members have completed the action. (Bug #34815537)

- **Group Replication:** Cleanup of resources used by OpenSSL connections created indirectly by group replication was not carried out as expected at all times. We fix this by adding cleanup functionality that can be called at any time such connections are created by group replication. (Bug #34727136)

- **Group Replication:** In some cases, the MySQL server continued to accept connections intended for group replication even after the group replication plugin had commenced shutdown. (Bug #34398622)

- **Microsoft Windows:** On Windows, a new *MySQL Configurator* application was added to help configure a MySQL server. It replaces the *MySQL Installer* application that installed and configured MySQL products in previous versions of MySQL. *MySQL Configurator* (`mysql_configurator.exe`) is included with both the MSI and Zip archive packages. (Bug #35461041)

- **JSON:** When the result of `JSON_VALUE()` was an empty string and was assigned to a user variable, the user variable could in some cases be set to `NULL` instead, as shown here:

  ```
  mysql> SELECT JSON_VALUE('{"fname": "Joe", "lname": ""}', '$.lname') INTO @myvar;
  Query OK, 1 row affected (0.01 sec)

  mysql> SELECT @myvar = '', @myvar IS NULL;
  +-------------+----------------+
  | @myvar = '' | @myvar IS NULL |
  +-------------+----------------+
  |        NULL |              1 |
  +-------------+----------------+
  1 row in set (0.00 sec)
  ```

  With this fix, the query just shown now returns `(1, 0)`, as expected. (Bug #35206138)

- **JSON:** Some JSON schemas were not always processed correctly by `JSON_SCHEMA_VALID()`. (Bug #109296, Bug #34867398)

- Some combinations of regular expression functions and arithmetic functions were not always evaluated correctly. (Bug #35462660)

- In rare cases, MySQL server could exit rather than emit an error message as expected. (Bug #35442407)

- The internal resource-group enhancement added in MySQL 8.0.31 and refactored in MySQL 8.0.32 is now reverted. (Bug #35434219)

  References: Reverted patches: Bug #34702833.

- An in-place upgrade from MySQL 5.7 to MySQL 8.0, without a server restart, could result in unexpected errors when executing queries on tables. This fix eliminates the need to restart the server between the upgrade and queries. (Bug #35410528)

- A fix in MySQL 8.0.33 made a change for `ORDER BY` items already resolved so as not to resolve them again (as is usually the case when a derived table is merged), but this did not handle the case in which an `ORDER BY` item was itself a reference. (Bug #35410465)

References: This issue is a regression of: Bug #34890862.

- Changes in `session_track_gtids` were not always handled correctly. (Bug #35401212)
- Some pointers were not always released following statement execution. (Bug #35395965)
- In `Item_func_min_max::cmp_datetimes()`, it was sometimes possible to set `null_value` when the current item was not actually nullable. (Bug #35380560, Bug #35492532)
- Some instances of subqueries within stored routines were not always handled correctly. (Bug #35377192)
- Fortified parsing of the network packet data sent by the server to the client. (Bug #35374491)
- Some queries using `INTERSECT` were not always processed correctly. (Bug #35362424)
- A `SELECT` statement within a prepared statement unexpectedly returned different results on successive executions. (Bug #35340987)

  References: This issue is a regression of: Bug #35060385.
- Encryption enhancements now strengthen compliance and remove the use of deprecated APIs. (Bug #35339886)
- When a column reference given by table name and column name was looked up in the function `find_item_in_list()`, we ignored that the item searched for might not have a table name, as it was not yet resolved. We fix this by making an explicit check for a null table name in the sought-after item. (Bug #35338776)
- Deprecated the `lz4_decompress` and `zlib_decompress` command-line utilities that exist to support the deprecated `mysqlpump` command-line utility. (Bug #35328235)
- Certain queries using `NULLIF()` led to an assertion. The issue was found to originate in `Item_func_nullif::resolve_type_inner()`, where, if the original data type was a temporal type, the type was adjusted to a string type but the result type was not also adjusted accordingly, which could later lead to later inconsistencies. This is fixed by setting the result type in such cases to `STRING_RESULT`. (Bug #35323398)
- On Linux, the `mysql` client's `ssl_session_data_print` command now saves files with an 0600 absolute mode (permissions) instead of the default 0644; when passing in the optional *filename* parameter. (Bug #35304195)
- Queries using `LIKE '%...%'` ran more poorly than in previous versions of MySQL. (Bug #35296563)
- We calculate the cost of `MATERIALIZE` paths by adding the cost of materialization to the sum of the cost of the child paths. If the number of output rows is undefined for a child, we ignore that child, as we assume that the cost of that child is then also undefined. If the child was an `AGGREGATE` path with implicit grouping, the number of output rows could be set to 1, even when the cost was undefined. We fix this by checking in such cases whether the cost of the child is actually defined, and—if it is not—skipping it. (Bug #35240913)

  References: See also: Bug #33834146, Bug #34302461.
- In `Bounded_queue::push()`, when `Key_generator::make_sortkey()` returns `UINT_MAX` (error), then no key has been produced; now when this occurs, we no longer update the internal queue.

  As part of this fix, `push()` now returns true on error. (Bug #35237721)
- The `authentication_oci` plugin is fixed to allow federated and provisioned users to connect to a DB System as a mapped Proxy User using an ephemeral key-pair generated through the OCI CLI. (Bug #35232697)

- Some queries using common table expressions were not always processed correctly. (Bug #35231475)
- The internal function `compare_pair_for_nulls()` did not always set an explicit return value. (Bug #35217471)
- Removed the clang-tidy checks that clash with the MySQL coding style. (Bug #35208735)
- Some subqueries using `EXISTS` in both the inner and outer parts of the query were not handled correctly. (Bug #35201901)
- Rotated audit log files now always reset the ID value of the bookmark to zero, rather than continuing the value from the previous file. (Bug #35200070)
- Errors were not always propagated correctly when evaluating items to be sorted by filesort. (Bug #35195181)

  References: See also: Bug #35145246.
- In certain cases, `UNIX_TIMESTAMP()` was evaluated prematurely. (Bug #35174730)
- When attempting to transform a scalar subquery to a derived table, we saw the top level query is implicitly grouped, so we moved the grouping into a first derived table. If, after this, we did not perform the original transformation, the initial transform had still been carried out, which should have been valid, but we neglected to look at join conditions in subqueries when substituting reference fields. In such cases we also did not descend into any subqueries other than derived table subqueries. (Bug #35170671)
- The fix for a previous issue with `ROLLUP` led to a premature server exit in debug builds. (Bug #35168639)

  References: This issue is a regression of: Bug #33830659.
- Simplified the implementation of `Item_func_make_set::val_str()` to make sure that we never try to reuse any of the input arguments, always using the local string buffer instead. (Bug #35154335, Bug #35158340)
- The transform of a scalar subquery into a join with a derived table where the subquery is in the `SELECT` list and the containing query is implicitly grouped should be allowed, but was rejected when the `subquery_to_derived` optimizer switch was enabled. (Bug #35150438)
- When transforming subqueries to a join with derived tables, with the containing query being grouped, we created an extra derived table in which to do the grouping. This process moved the initial select list items from the containing query into the extra derived table, replacing all of the original select list items (other than subqueries, which get their own derived tables) with columns from the extra derived table.

  This logic did not handle `DEFAULT` correctly due to the manner in which default values were modelled internally. This fix adds support for `DEFAULT(`*`expression`*`)` in queries undergoing the transform previously mentioned. This fix also solves an issue with item names in metadata whereby two occurrences of the same column in the select list were given the same item name as a result of this same transform. (Bug #35150085, Bug #35101169)
- A query of the form `SELECT * FROM t1 WHERE (SELECT a FROM t2 WHERE t2.a=t1.a + ABS(t2.b)) > 0` should be rejected with `Subquery returns more than 1 row`, but when the `subquery_to_derived` optimization was enabled, the transform was erroneously applied and the query returned an incorrect result. (Bug #35101630)
- Handling of certain potentially conflicting `GRANT` statements has been improved. (Bug #35089304)
- A query using both `MEMBER OF()` and `ORDER BY DESC` returned only a partial result set following the creation of a multi-valued index on a JSON column. This is similar to an issue fixed in

MySQL 8.0.30, but with the addition of the `ORDER BY DESC` clause to the prblematic query. (Bug #35012146)

References: See also: Bug #106621, Bug #33917625.

- The debug server asserted on certain operations involving `DECIMAL` values. (Bug #34973932)
- The nullability of `ANY` subqueries was sometimes incorrect because the nullability of the left operand was not taken into account. We fix this by marking an `ANY` subquery as nullable whenever the left operand is nullable. (Bug #34940790)
- All instances of adding and replacing expressions in the select list when transforming subqueries to use derived tables and joins have been changed so that their reference counts are maintained properly. (Bug #34927110)
- Aggregation of item type from multiple arguments required processing in multiple internal functions; this has been simplified such that it is now performed in one function only. This should improve the efficiency of this process, which is used for expressions that are the results of set operations, and those that are output from the `CASE` operator (and the associated functions `COALESCE()` and `IF()`), as well as `LEAD()` and `LAG()`. (Bug #34847836)
- Index Merge (see Index Merge Optimization) should favor ROR-union plans (that is, using RowID Ordered Retrieval) over sort-union plans if they have similar costs, since sort-union requires additionally sorting of the rows by row ID whereas ROR-union does not.

  For each part of a `WHERE` clause containing an `OR` condition, the range optimizer gets the best range scan possible and uses all these range scans to build an index merge scan (that is, a sort-union scan). If it finds that all the best range scans are also ROR-scans, the range optimizer always proposes a ROR-union scan because it is always cheaper than a sort-union scan. Problems arose when the best range scan for any one part of an `OR` condition is not a ROR-scan, in which case, the range optimizer always chose sort-union. This was true even in cases, where it might be advantageous to choose a ROR-scan (even though it might not be the best range scan to handle one part of the OR condition), since this would eleminate any need to sort the rows by row ID.

  Now, in such cases, when determining the best range scan, the range optimizer also detects whether there is any possible ROR-scan, and uses this information to see whether each part of the `OR` condition has at least one possible ROR-scan. If so, we rerun the range optimizer to obtain the best ROR-scan for handling each part of the `OR` condition, and to make a ROR-union path. We then compare this cost with the cost of a sort-union when proposing the final plan. (Bug #34826692, Bug #35302794)
- Selecting from a view sometimes raised the error `Illegal mix of collations ... for operation '='` when the collation used in the table or tables from which the view definition selected did not match the current session value of `collation_connection`. (Bug #34801210)
- If a view (`v1`) accessed another view (`v2`), and if `v2` was recreated, then `SHOW COLUMNS FROM v1` reported an invalid view error. This issue occurred when the user was granted privileges to all resources (`*.*`), but not table-level or column-level privileges. It is fixed by removing the condition that caused an omission of the proper table-level check. (Bug #34467659)
- Queries using `DISTINCT` treated 0 and -0 differently. (Bug #34361437)
- `ANALYZE TABLE` with `UPDATE HISTOGRAM` or `DROP HISTOGRAM` invalidated the `TABLE_SHARE`, which meant that subsequent queries were required to wait for all queries then running to terminate before the old `TABLE_SHARE` could be freed and a new one initialized with the updated collection of histograms for the table. This could introduce long waits, as queries issued after the `TABLE_SHARE` was invalidated had to wait for any existing long-running queries that referenced the old `TABLE_SHARE` to terminate.

  This fix changes the behavior of the histogram commands to mark tables for reopening instead of invalidating the `TABLE_SHARE`. Instead of having a single set of table histograms cached on the

`TABLE_SHARE`, we now maintain a collection of reference-counted sets of table histograms on the share. When the histograms on a given table are modified, we now insert a new snapshot of the set of histograms into the collection on the `TABLE_SHARE` and mark it current. When a table object is opened, it acquires a pointer to the current snapshot of the set of histograms for the table from the share, and when the table object is closed it releases its pointer back to the share.

By using multiple reference-counted versions of histogram statistics for a table we avoid the potential wait for synchronization of all queries on the table around the reinitialization of the `TABLE_SHARE` when histograms are updated or dropped. (Bug #34288890, Bug #35419418)

- Valid MySQL commands (`use` and `status`) and C API functions (`mysql_refresh`, `mysql_stat`, `mysql_dump_debug_info`, `mysql_ping`, `mysql_set_server_option`, `mysql_list_processes`, and `mysql_reset_connection`) could write an error message to the audit log, even though running the command or calling the function emitted no such error. (Bug #33966181)

- Increased the maximum fixed array size to 8192 instead of 512. This fixes an issue with `mysqladmin` extended status requests, which can exceed 512 entries.

  Our thanks to Meta for the contribution. (Bug #30810617)

- The `mysqldump --column-statistics` option attempted to select from `information_schema.column_statistics` against MySQL versions before 8.0.2, but this now generates the `warning column statistics not supported by the server` and sets the option to false.

  Our thanks to Meta for the contribution. (Bug #28782417)

- The function used by MySQL to get the length of a directory name was enhanced. (Bug #28047376)

- Executing a query with an implicit aggregation should return exactly one row, unless the query has a `HAVING` clause that filters out the row, but a query with a `HAVING` clause which evaluated to `FALSE` sometimes ignored this, and returned a row regardless. (Bug #14272020)

- Some complex queries using multiple common table expressions were not always handled correctly. (Bug #112021, Bug #35284734, Bug #35694546)

  References: See also: Bug #111994, Bug #35686058. This issue is a regression of: Bug #34377854.

- The presence of an unused window function in a query, along with an `ORDER BY` that could have been eliminated, led to an unplanned server exit. (Bug #111585, Bug #35168639, Bug #35204224, Bug #35545377)

  References: This issue is a regression of: Bug #35118579.

- `ORDER BY RANDOM_BYTES()` had no effect on query output. (Bug #111252, Bug #35148945, Bug #35457136)

- Improved the `mysql` client's *status* output; the Protocol row now includes the compression algorithm and zstd level.

  Our thanks to Daniël van Eeden for the contribution. (Bug #110950, Bug #35369870)

- The MySQL source code documentation was missing the following information about C API protocols: `zstd_compression_level` is only sent when `CLIENT_ZSTD_COMPRESSION_ALGORITHM` is set.

  Our thanks to Daniël van Eeden for the contribution. (Bug #110939, Bug #35365351)

- In certain cases, `VALUES ROW()` did not handle expressions which evaluated to `NULL` correctly. (Bug #110925, Bug #35363550)

- The `QUOTE()` function returned unexpected results with columns selected from a table having the `utf16` character set. (Bug #110672, Bug #35286970)
- Fixed an issue which could occur when loading user-defined functions. (Bug #110576, Bug #35242734)
- Concurrent execution of `FLUSH STATUS`, `COM_CHANGE_USER`, and `SELECT FROM I_S.PROCESSLIST` could result in a deadlock. A similar issue was observed for concurrent execution of `COM_STATISTICS`, `COM_CHANGE_USER`, and `SHOW PROCESSLIST`.

  Our thanks to Dmitry Lenev for the contribution. (Bug #110494, Bug #35218030)
- The `mysqldump` utility could generate invalid INSERT statements for generated columns. (Bug #110462, Bug #35208605)
- For `mysqldump`: usage would unexpectedly halt when used against tables with functional indexes. (Bug #110452, Bug #35205310)
- An impossible `WHERE` similar to `WHERE int_col = 05687.3E-84` was not always handled correctly. (Bug #110434, Bug #35200367)
- The loading and unloading of UCA character sets has been rewritten to improve memory handling when cycling through initialization and deinitialization. (Bug #109540, Bug #110836, Bug #34969838, Bug #35341006)
- During optimization, range-select tree creation uses logic which differs based on the left-hand side of the `IN()` predicate. For a field item, each value on the right-hand side is added to an OR tree to create the necessary expression. In the case of a row item comparison (example: `WHERE (a,b) IN ((n1,m1), (n2, m2), ...)`), an expression in disjunctive normal form (DNF) is needed. A DNF expression is created by adding an `AND` tree with column values to an `OR` tree for each set of RHS values, but instead the `OR` tree was added to the `AND` tree causing the tree merge to require exponential time due to $O(n^2)$ runtime complexity. (Bug #108963, Bug #34758905)
- When using `SELECT` to create a table and the statement has an expression of type `GEOMETRY`, MySQL could generate an empty string as the column value by default. To resolve this issue, MySQL no longer generates default values for columns of type `GEOMETRY` under these circumstances. Our thanks to Tencent for the contribution. (Bug #107996, Bug #34426943)
- Removed an assertion encountered when creating fields of type `YEAR` for temporary tables holding results of `UNION` operations. (Bug #107826, Bug #34370933, Bug #35282236)
- For index skip scans, the first range read set an end-of-range value to indicate the end of the first range, but the next range read did not clear the stale end-of-range value and applies this stale value to the current range. Since the indicated end-of-range boundary had already been crossed in the previous range read, this caused the reads to stop, causing multiple rows to be missed in the result.

  We fix this by making sure in such cases that the old end-of-range value is cleared. (Bug #107460, Bug #34235624, Bug #34982949)

Index

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y