



Exploring MySQL 8 Innovations



...and how Percona can help.



Arunjith Aravindan
Senior MySQL DBA



Hyderabad, India 2025

PERCONA
UNIVERSITY

Exploring MySQL 8 Innovations

10:35AM

11:05 AM

Mastering Amazon Relational Database Service for MySQL

Building and configuring MySQL instances



Jeyaram Ayyalusamy
Arunjith Aravindan
Dr. P.V.Kumaraguru



O'REILLY®

Hands-On MySQL Administration

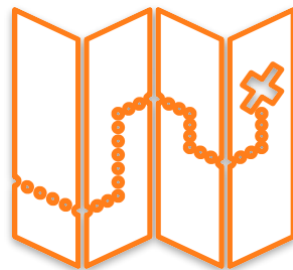
Managing MySQL on Premise and on the Cloud



Arunjith Aravindan
& Jeyaram Ayyalusamy

Agenda

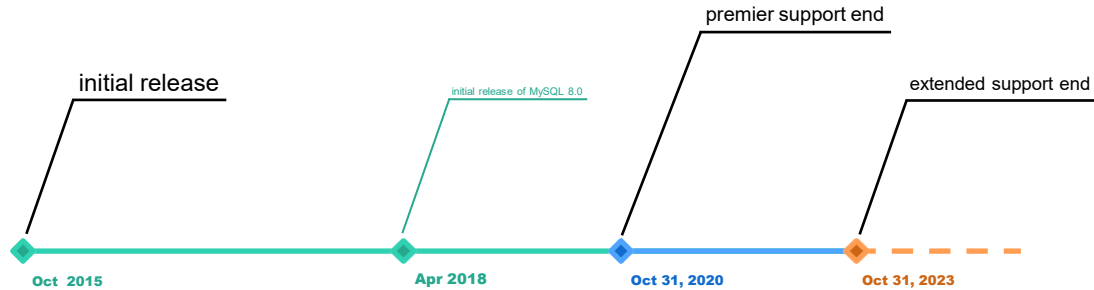
- Where are we in the MySQL lifecycle journey?
- Why are the instances still running on MySQL version 5.7?
- Upgrade to MySQL 8.0: facts or urban legends!
- MySQL 8 key changes!
- MySQL 8.0 vs. 8.4!



Where are we in MySQL lifecycle?

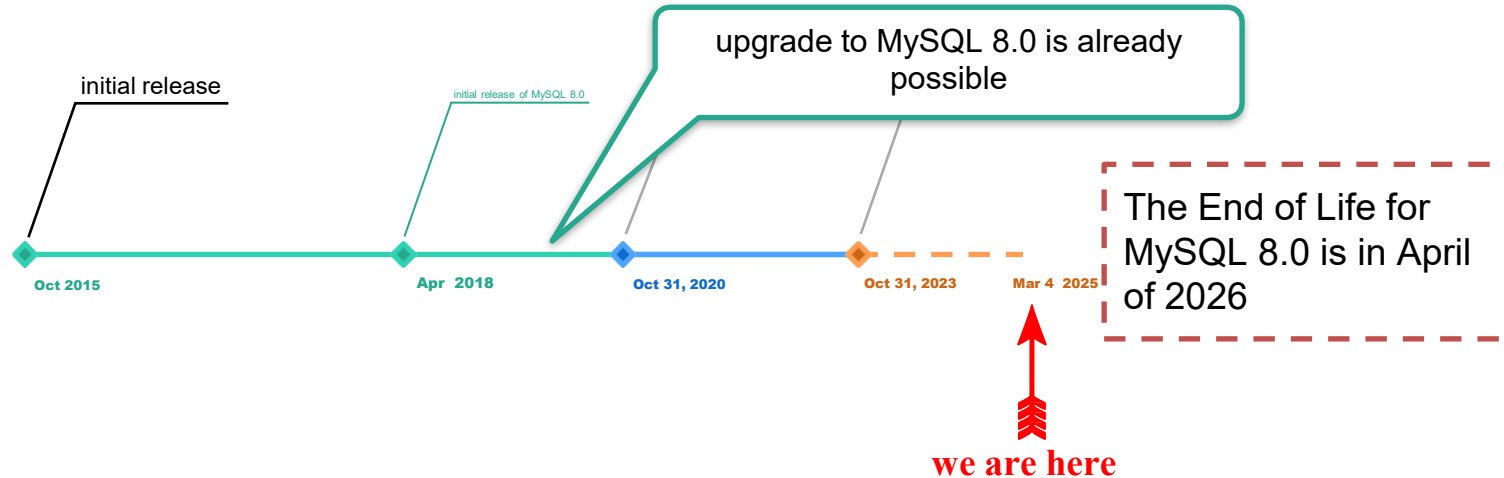


MySQL 5.7 lifecycle



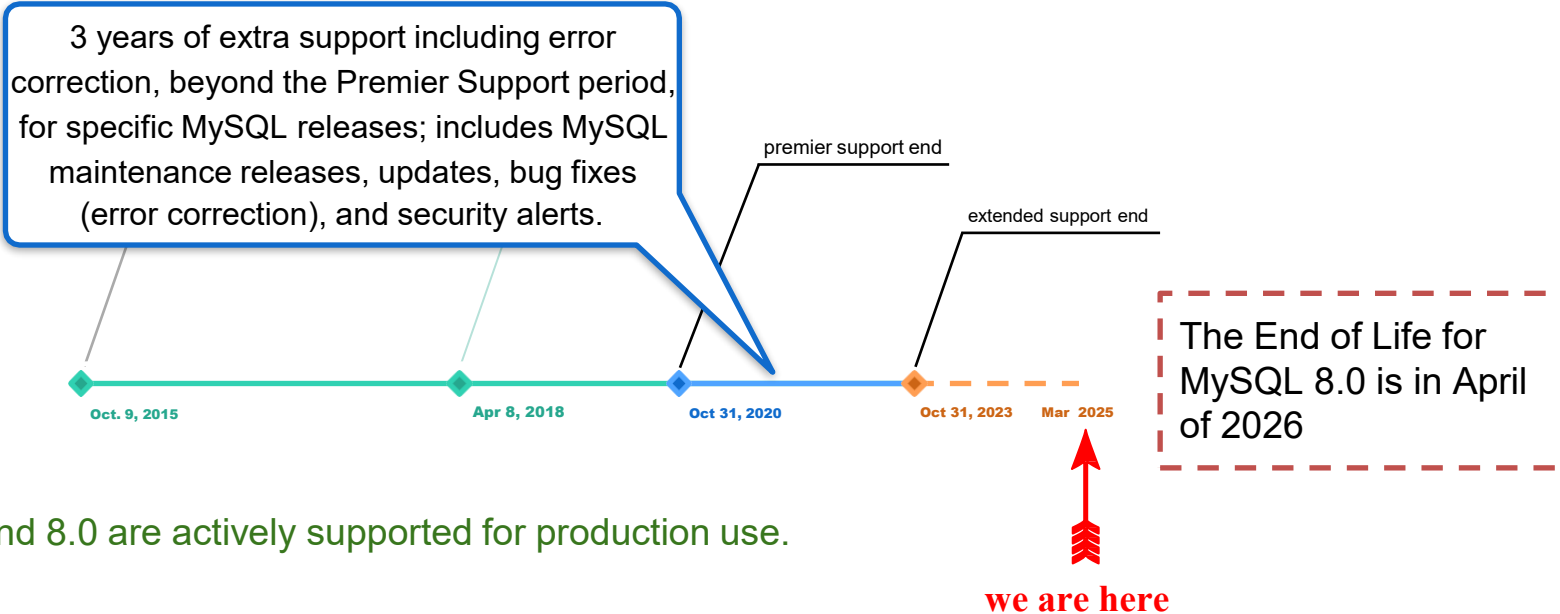
- Lifecycle of MySQL 5.7 was spanned across 8 years
- Extended support ended on MySQL 5.7

MySQL 5.7 & 8.0 lifecycle



MySQL 8.4 and 8.0 are actively supported for production use.

MySQL 5.7 extra support from Percona



MySQL 8.4 and 8.0 are actively supported for production use.

MySQL 8.4: Long Term Support (LTS) Release

Release Date: April 2024

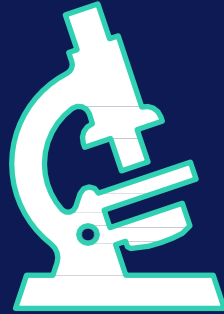
LTS Designation: MySQL 8.4.0 is designated as a Long Term Support (LTS) release.

Support Duration:

- Premier Support: 5 years
- Extended Support: 3 additional years
- Total Extended Support End: Approximately April 2032.

- The LTS release is the final minor version within its major release series.
- 8.4 is the 8.x LTS, then the next major release will be MySQL 9.0.

Why are the instances still running on MySQL version 5.7?



What prevents you from upgrading to MySQL 8.0?

answers from Percona customers

Technical reasons

- version 8.0 unfit for use with the existing stack
- unsolvable backend application dependencies
- original development staff has left the company
- might have tried upgrade to 8.0 in the past and failed
- often customers with large, monolithic business applications

What prevents you from upgrading to MySQL 8.0?

answers from Percona customers

Technical reasons

- version 8.0 unfit for use with the existing stack
- unsolvable backend application dependencies
- original development staff has left the company
- might have tried upgrade to 8.0 in the past and failed
- often customers with large, monolithic business applications

Perception of cost and risk

- have read about the horrors of the upgrade
- are afraid of the unnamed risks and issues
- understand that the cost of software and services' maintenance is non-zero

What prevents you from upgrading to MySQL 8.0?

answers from Percona customers

Technical reasons

- version 8.0 unfit for use with the existing stack
- unsolvable backend application dependencies
- original development staff has left the company
- might have tried upgrade to 8.0 in the past and failed
- often customers with large, monolithic business applications

Perception of cost and risk

- have read about the horrors of the upgrade
- are afraid of the unnamed risks and issues
- understand that the cost of software and services' maintenance is non-zero

See no evil, hear no evil

- “let’s not fix things which ain’t broken”
- “let’s pretend this problem does not exist”

Upgrading from MySQL 5.7 to 8.0. Facts or urban legends?



Upgrading MySQL from 5.7 to 8.0

Are there any real
differences between
MySQL 5.7 and 8.0?

Where are the
missing 6.x and 7.x
releases?

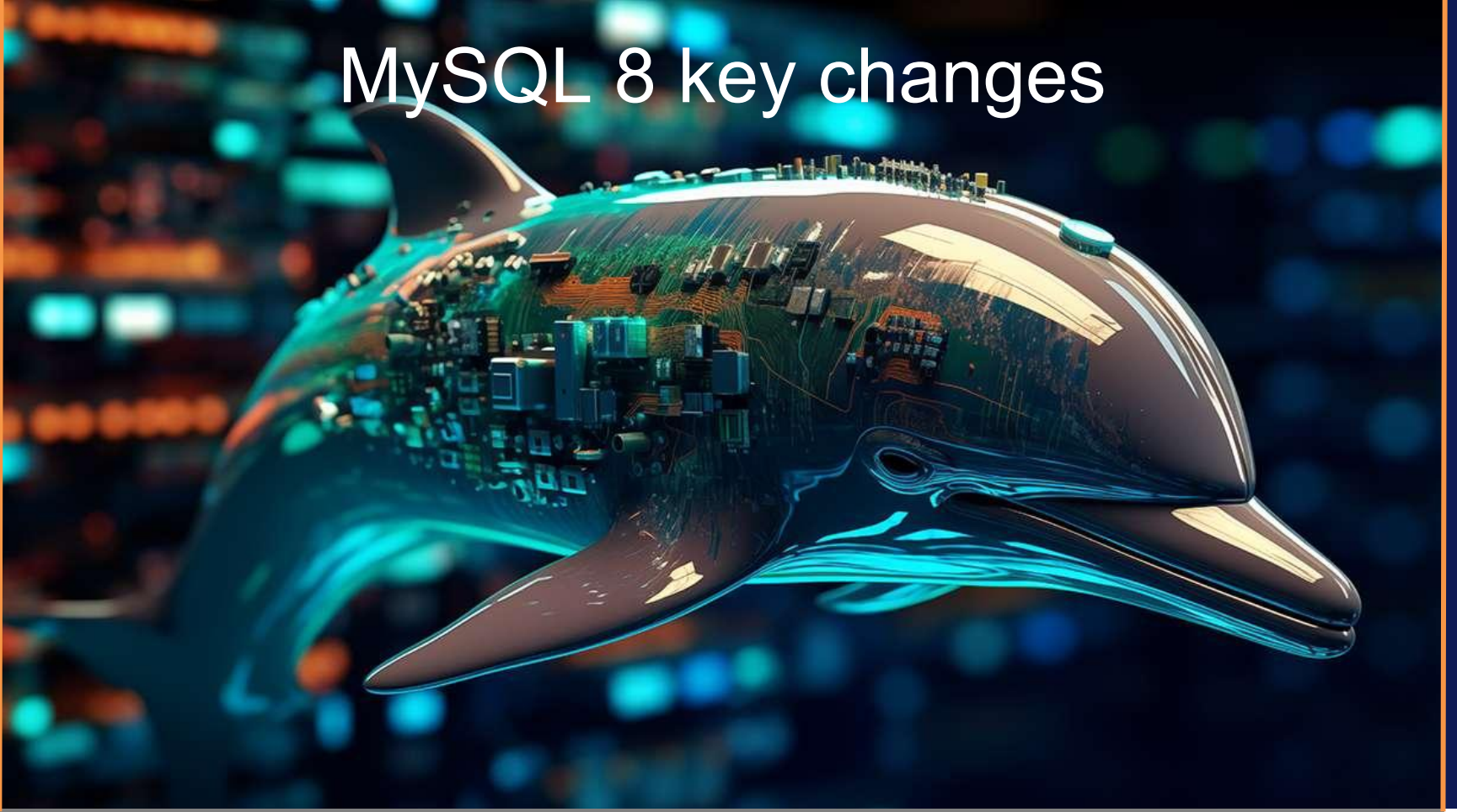
Upgrading MySQL from 5.7 to 8.0

Are there any real
differences between
MySQL 5.7 and 8.0?

Where are the
missing 6.x and 7.x
releases?

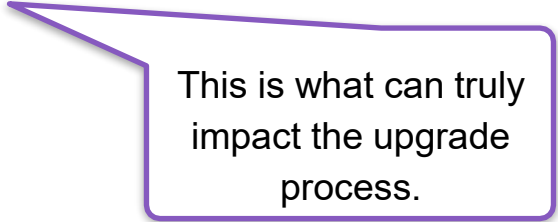
the answer to both
questions is "YES"

MySQL 8 key changes

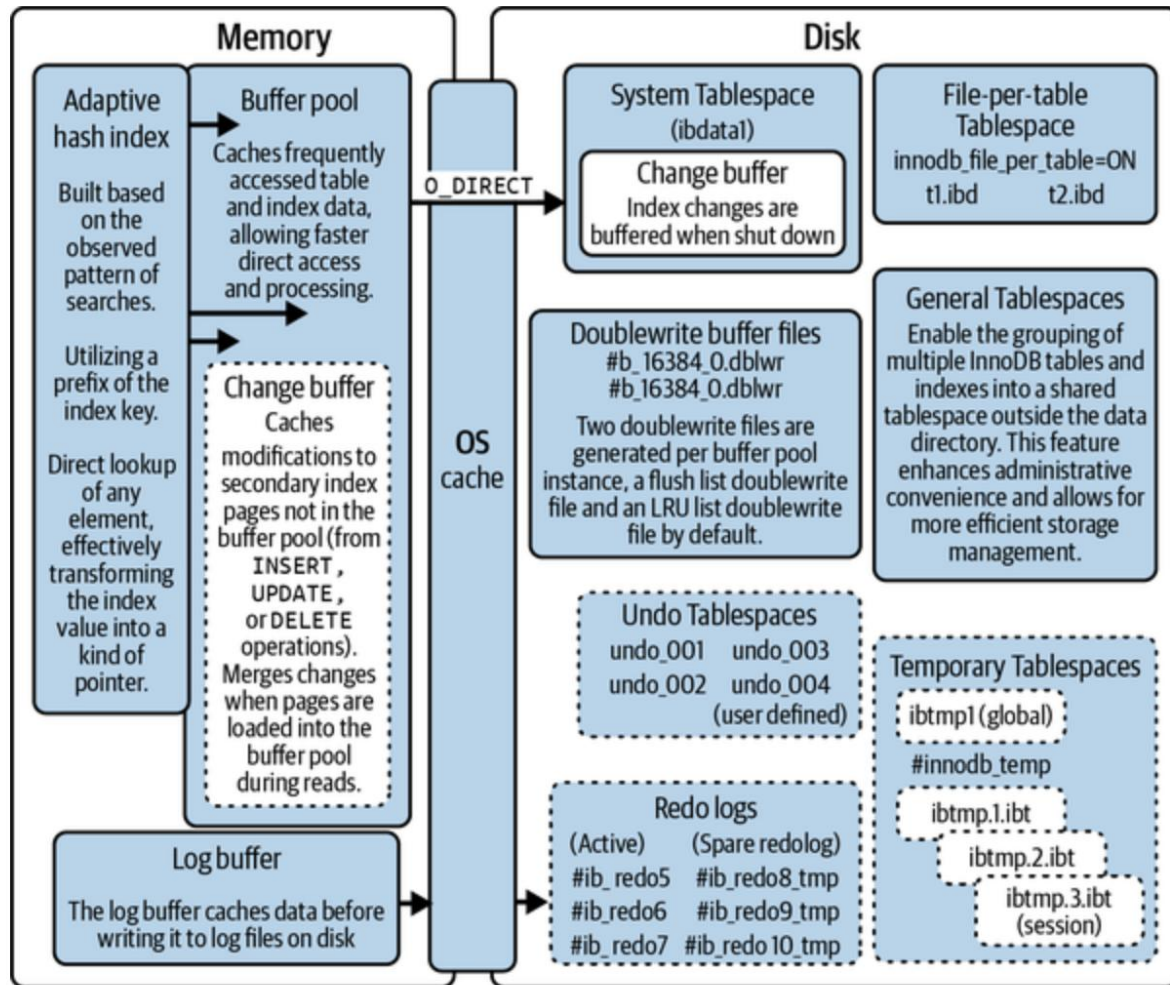


Review of key changes

- **Major new capabilities and changes incorporated**
 - General changes
 - InnoDB changes
 - Security and Account Management changes
- **Deprecations**
- **Removals**
- **Bonus upgrade impediment**



This is what can truly impact the upgrade process.



General changes in MySQL 8.0 vs. 5.7

- new [transaction data dictionary](#) storing information about database objects
- atomic DDL statement now combines data dictionary updates, storage engine operations and binary log writes associated with a DDL operation
- MySQL server now automatically runs all necessary upgrade tasks (system tables, objects in other schemas like sys schema and user schema) at the next startup, so no need to manually run “mysql_upgrade” as of version 8.0.16
- SSL session reuse supported by default with the timeout setting
- MySQL server now supports for creation and management of resource groups and permits assigning threads to particular groups
- table encryption can be managed globally by defining and enforcing encryption defaults
- [default character set](#) is changed from “latin1” to “utf8mb4”, which now has several new collations available
- added support for expressions as default values for the BLOB, TEXT, GEOMETRY and JSON data types
- new type of backup lock added, which permits DMLs during an online backup, while preventing operations which could result in an inconsistent snapshot
- TCP/IP port can be now configured specifically for administrative connections, even if “max_connections” level is already reached on the primary port
- added support for [invisible indexes](#), which are not used by the optimiser and make it possible to test the effect of removing an index without actually removing it
- added support for document store, for developing both SQL and NoSQL applications with a single DB engine

InnoDB changes in MySQL 8.0 vs. 5.7

- the maximum [auto-increment counter](#) value is now persistent across the server restarts
- in an event of index tree corruption, InnoDB writes a corruption flag to the redo log, which makes the corruption flag crash-safe
- new dynamic variable “innodb_deadlock_detect” may be used to disable deadlock detection
- InnoDB temporary tables are now created in session temporary tablespaces (*.ibt files)
- system tables and data dictionary tables are now created in a single InnoDB tablespace names mysql.ibd, in the MySQL data directory
- by default undo-logs now reside in two undo tablespaces that are created when the MySQL instance is initialised; undo logs are no longer created in the system tablespace
- new “innodb_dedicated_server” variable (disabled by default) can be used to automatically configure the options based on the detected memory availability
- tablespace files can be moved or restored to a new location while the server is offline using the “innodb_directories” option

Security and account management changes in MySQL 8.0 vs. 5.7

- the grant tables in mysql system database are now InnoDB (transactional) tables
- a new “caching_sha2_password” authentication plugin is available; like the “sha256_password” plugin, it implements SHA-256 password hashing, but uses caching to address latency issues
- added support for roles which are names collections of privileges; roles can be created and dropped, and have privileges granted or revoked; roles can be granted to or revoked from users
- introduced account categories concept incorporated, with system and regular users distinguished by SYSTEM_USER
- password history is now maintained, enabling restrictions on password reuse
- MySQL now supports FIPS mode, if compiled using OpenSSL with OpenSSL library and FIPS Object Module available at runtime. (Federal Information Processing Standards)
- administrators are now enabled to configure user accounts such that too many failed consecutive login attempts lead to temporary account locking
- as of MySQL 8.0.27 the multi-factor authentication is supported which makes it possible to have up to three authentication methods enabled per account

somewhat outdated list of differences on MySQL website: <https://dev.mysql.com/blog-archive/the-complete-list-of-new-features-in-mysql-8-0/>

MySQL 8.0 vs. 8.4!

- MySQL native password has long been depreciated, but it is no longer loaded by default.
- The InnoDB_flush_method has changed from fsync to O_DIRECT on Linux. O_DIRECT if supported, otherwise fsync.
- Using the terms MASTER and SLAVE in replication commands may finally be replaced by SOURCE and REPLICA.
- innodb_change_buffering default set to none.
- innodb_adaptive_hash_index default set to OFF.
- And 143 bugs were fixed.

List of differences on MySQL website https://docs.oracle.com/cd/E17952_01/mysql-8.4-en/mysql-nutshell.html

New capabilities and changes. Already worried?

General changes in MySQL 8.0 vs. 5.7

- new transaction data dictionary storing information about database objects
- atomic DDL statement now combines data dictionary updates, storage engine operations and binary log writes associated with a DDL operation
- MySQL server now automatically runs all necessary upgrade tasks (system tables, objects in other schemas like sys schema and user schema) at the next startup, so no need to manually run `mysql_upgrade` as of version 8.0.16
- SSL session reuse supported by default with the timeout setting
- MySQL server now supports for creation and management of resource groups and permits assigning threads to particular groups
- table encryption can be managed globally by defining and enforcing encryption defaults
- default character set is changed from latin1 to utf8mb4, which now has several new collations available
- added support for expressions as default values for the BLOB, TEXT, GEOMETRY and JSON data types
- new type of backup lock added, which permits DMLs during an online backup, while preventing operations which could result in an inconsistent snapshot
- TCP/IP port can be now configured specifically for administrative connections, even if `max_connections` level is already reached on the primary port
- added support for invisible indexes, which are not used by the optimizer and make it possible to test the effect of removing
- added support and No
- it's now possible using GLOBAL

© Copyright 2023 Percona® LLC. All rights reserved

InnoDB changes in MySQL 8.0 vs. 5.7

- the maximum auto-increment counter value is now persistent across the server restarts
- in an event of index tree corruption, InnoDB writes a corruption flag to the redo log, which makes the corruption flag crash-safe
- new dynamic variable `"innodb_deadlock_detect"` may be used to disable deadlock detection
- InnoDB temporary tables are now created in the shared temporary tablespace `"tmp1"`
- system tables and data dictionary tables are now created in a single InnoDB tablespace named `mysql.ibd`, in the MySQL data directory
- by default undo-logs now reside in two undo tablespaces that are created when the MySQL instance is initialised; undo logs are no longer created in the system tablespace
- new `"innodb_dedicated_server"` variable (disabled by default) can be used to automatically configure the options based on the detected memory availability
- tablespace files can be moved or restored to a new location while the server is offline using the `"innodb_directories"` option

© Copyright 2023 Percona® LLC. All rights reserved

Security and account management changes in MySQL 8.0 vs. 5.7

- the grant tables in `mysql` system database are now InnoDB (transactional) tables
- a new `"caching_sha2_password"` authentication plugin is available; like the `"sha256_password"` plugin, it implements SHA-256 password hashing, but uses caching to address latency issues
- added support for roles which are names collections of privileges; roles can be created and dropped, and have privileges granted or revoked; roles can be granted to or revoked from users
- user account categories concept incorporated, with system and regular users distinguished by `SYSTEM_USER` privilege
- password history is now maintained, enabling restrictions on password reuse
- MySQL now supports FIPS mode, if compiled using OpenSSL with OpenSSL library and FIPS Object Module available at runtime
- administrators are now enabled to configure user accounts such that too many failed consecutive login attempts lead to temporary account locking
- as of MySQL 8.0.27 the multi-factor authentication is supported which makes it possible to have up to three authentication methods enabled per account

© Copyright 2023 Percona® LLC. All rights reserved

PERCONA

we are only
getting
started

Deprecations

- the “utf8mb3” character set is deprecated, with “utf8mb4” recommended instead; “utf8mb3” is still valid in MySQL 8.0
- the “sha256_password” plugin is deprecated and “caching_sha2_password” is recommended to use instead, as its superset
- the “mysql_upgrade” client is deprecated because of its capabilities for upgrading the system tables in the mysql system schema, and objects in other schemas have been moved to MySQL server; as of MySQL 8.0.16 the server performs all these tasks
- the “validate_password” plugin has been reimplemented to use the server component infrastructure; it is still available but deprecated
- the “ENGINE” clause for all the ALTER TABLESPACE and DROP TABLESPACE statements is deprecated
- the “PAD_CHAR_TO_FULL_LENGTH” SQL mode is deprecated
- the “AUTO_INCREMENT” support is deprecated for columns of type FLOAT and DOUBLE (and any synonyms); consider removing AUTO_INCREMENT attribute from such columns, or convert them to an integer type
- the “UNSIGNED” attribute is deprecated for columns of type FLOAT, DOUBLE and DECIMAL (and any synonyms); consider using a simple CHECK constraint instead
- FLOAT(M,D) and DOUBLE(M,D) syntax to specify the number of digits for columns of type FLOAT and DOUBLE (and any synonyms) is a nonstandard MySQL extension and is deprecated
- The nonstandard C-style “&&”, “||”, and “!” operators that are synonyms for the standard SQL AND, OR, and NOT operators, respectively, are deprecated
- The “mysql_upgrade_info” file, which creates a text file in the data directory and used to store the MySQL version number is deprecated
- The “relay_log_info_file” system variable and “–master-info-file” option are deprecated; the use of files for the relay log info log and master info log has been superseded by crash-safe slave tables, which are the default in MySQL 8.0.
- The use of the “MYSQL_PWD” environment variable to specify a MySQL password is deprecated.

Deprecations. You must act ASAP!

Deprecations

- the "utf8mb3" character set is deprecated, with "utf8mb4" recommended instead; "utf8mb3" is still valid in MySQL 8.0
- the "sha256_password" plugin is deprecated and "caching_sha2_password" is recommended to use instead, as its superset
- the "mysql_upgrade" client is deprecated because of its capabilities for upgrading the system tables in the mysql system schema, and objects in other schemas have been moved to MySQL server; as of MySQL 8.0.16 the server performs all these tasks
- the "validate_password" plugin has been reimplemented to use the server component infrastructure; it is still available but deprecated
- the "ENGINE" clause for all the ALTER TABLESPACE and DROP TABLESPACE statements is deprecated
- the "PAD_CHAR_TO_FULL_LENGTH" SQL mode is deprecated
- the "AUTO_INCREMENT" support is deprecated for columns of type FLOAT and DOUBLE (and any synonyms); consider removing AUTO_INCREMENT attribute from such columns, or convert them to an integer type
- the "UNSIGNED" attribute is deprecated for columns of type FLOAT, DOUBLE and DECIMAL (and any synonyms); consider using a simple CHECK constraint instead
- FLOAT(M,D) and DOUBLE(M,D) syntax to specify the number of digits for columns of type FLOAT and DOUBLE (and any synonyms) is a nonstandard MySQL extension and is deprecated
- The nonstandard C-style "&&", "||", and "!!" operators that are synonyms for the standard SQL AND, OR, and NOT operators, respectively, are deprecated
- The "mysql_upgrade_info" file, which creates a text file in the data directory and used to store the MySQL version number is deprecated
- The "relay_log_info_file" system variable and "master-info-file" option are deprecated; the use of files for the relay log info log and master info log has been superseded by crash-safe slave tables, which are the default in MySQL 8.0.
- The use of the "MYSQL_PWD" environment variable to specify a MySQL password is deprecated.

© Copyright 2023 Percona LLC. All rights reserved.



- Deprecations are changes in the MySQL which will lead to Removals in future releases
- While the features still exist, their usage will spill errors into log files and console out

the time to adjust
you applications is NOW!,
if you are on 8.0
already

Removals

- the “innodb_locks_unsafe_for_binlog” system variable was removed; instead use “READ COMMITTED” isolation level which provides similar functionality
- using “GRANT” to create users; instead, use “CREATE USER”; following this practice makes the “NO_AUTO_CREATE_USER SQL” mode immaterial for GRANT statements, so it too is removed, and an error now is written to the server log when the presence of this value for the sql_mode option in the options file prevents mysqld from starting
- using GRANT to modify account properties other than privilege assignments, including authentication, SSL, and resource-limit properties; instead, establish such properties at account-creation time with “CREATE USER” or modify them afterward with “ALTER USER”
- “IDENTIFIED BY PASSWORD ‘auth_string’” syntax for “CREATE USER” and GRANT; instead, use “IDENTIFIED WITH auth_plugin AS ‘auth_string’” for “CREATE USER” and “ALTER USER”, where the ‘auth_string’ value is in a format compatible with the named plugin
- the PASSWORD() function; additionally, PASSWORD() removal means that “SET PASSWORD ... = PASSWORD(‘auth_string’)” syntax is no longer available

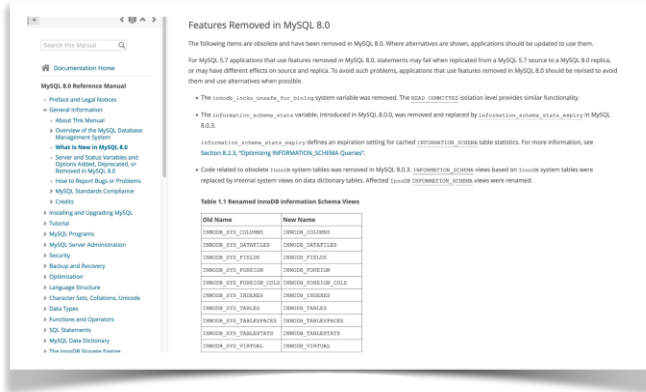
- the “old_passwords” system variable
- the [query cache was removed](#); removal includes the following:
 - the FLUSH QUERY CACHE and RESET QUERY CACHE statements
 - these system variables: query_cache_limit, query_cache_min_res_unit, query_cache_size, query_cache_type, query_cache_wlock_invalidate.
 - these status variables: Qcache_free_blocks, Qcache_free_memory, Qcache_hits, Qcache_inserts, Qcache_lowmem_prunes, Qcache_not_cached, Qcache_queries_in_cache, Qcache_total_blocks.
 - These thread states: checking privileges on cached query, checking query cache for a query, invalidating query cache entries, sending cached result to the client, storing result in the query cache, Waiting for query cache lock.
- the “tx_isolation” and “tx_read_only” system variables have been removed; instead use “transaction_isolation” and “transaction_read_only”
- the “sync_frm” system variable has been removed because *.frm files have become obsolete.
- the secure_auth system variable and “–secure-auth” client option have been removed; also the “MYSQL_SECURE_AUTH” option for the mysql_options() C API function was removed
- the “log_warnings” system variable and “–log-warnings” server option have been removed; instead use the “log_error_verbosity” system variable
- the global scope for the “sql_log_bin” system variable was removed; “sql_log_bin” has session scope

Removals - continued (1)

- the unused “date_format”, “datetime_format”, “time_format”, and “max_tmp_tables” system variables are removed
- the deprecated “ASC” or “DESC” qualifiers for “GROUP BY” clauses are removed; queries that previously relied on “GROUP BY” sorting may produce results that differ from previous MySQL versions; to produce a given sort order, provide an “ORDER BY” clause
- the parser no longer treats N as a synonym for NULL in SQL statements; use NULL instead; this change does not affect text file import or export operations performed with “LOAD DATA” or “SELECT ... INTO OUTFILE”, for which NULL continues to be represented by N
- the client-side “--ssl” and “--ssl-verify-server-cert” options have been removed;
 - use “--ssl-mode=REQUIRED” instead of “--ssl=1” or “--enable-ssl”;
 - use “--ssl-mode=DISABLED” instead of “--ssl=0”, “--skip-ssl”, or “--disable-ssl”
 - use “--ssl-mode=VERIFY_IDENTITY” instead of “--ssl-verify-server-cert” options
- The “mysql_install_db” program has been removed from MySQL distributions;
 - data directory initialisation should be performed by invoking “mysqld” with the “--initialize” or “--initialize-insecure” option instead;
 - in addition, the “--bootstrap” option for “mysqld” that was used by “mysql_install_db” was removed, and the “INSTALL_SCRIPTDIR” CMake option that controlled the installation location for “mysql_install_db” was removed
- The “mysql_plugin” utility was removed
 - alternatives include loading plugins at server startup using the “--plugin-load” or “--plugin-load-add” option, or at runtime using the “INSTALL PLUGIN” statement
- The “resolveip” utility is removed
 - instead use “nslookup”, “host”, or “dig”

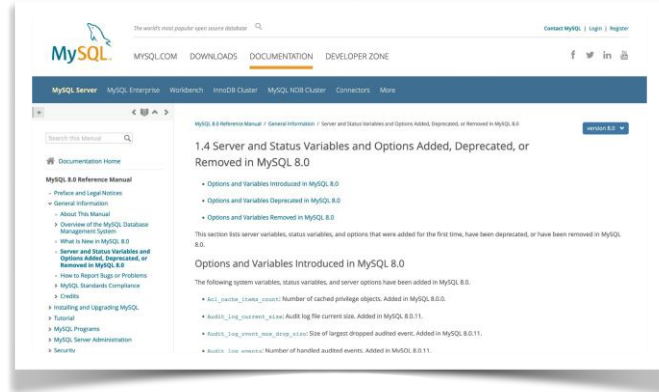
Removals - continued (2)

**more information
about removed features**



<https://dev.mysql.com/doc/refman/8.0/en/mysql-nutshell.html#mysql-nutshell-removals>

**more information
about removed server
and status variables**



<https://dev.mysql.com/doc/refman/8.0/en/added-deprecated-removed.html>

Removals

- the `inmod:` looks, instead, for `CREATE` system variable was removed; unsafe, use `REDO COMMITTED` option and `inmod` variable instead
- using `GRANT` to create users; instead, use `CREATE USER`; following this practice makes the `GRANT` statements more readable, and the `GRANT` statements, so it too is removed, and an error now is written to the server log when the presence of this `GRANT` mode option in the options file prevents myself from starting
- using `GRANT` to modify account properties other than privilege assignments; updating authentication, SSL, and resource limits; instead, use `ALTER USER` for the properties at account-creation time with `CREATE USER` or `ALTER USER` afterwards
- `"IDENTIFIED BY PASSWORD 'auth_string'"` syntax for `CREATE USER` and `GRANT`; instead, use `"IDENTIFIED WITH PLUGIN 'auth_string' FOR 'CREATE USER' AS 'auth_string'"`, where `auth_string` value is in a format compatible with the named plugin
- the `PASSWORD()` (function; additionally, `PASSWORD()` removed means that `"SET PASSWORD = PASSWORD('auth_string')"` is not supported

[illegible]

- the unused "date_format", "dateiname", "format", "time_format", and "time_inn_name" tables system variables
- the deprecated "ASC" or "DESC" qualifiers for "GROUP BY" clauses are removed; queries that previously relied on "GROUP BY" sorting may produce results that differ from previous MySQL versions; to provide a given sort order, provide an "ORDER BY" clause
- the parser no longer treats N as a synonym for NULL in SQL statements; use NULL instead; this change does not affect the file parser
- the parser no longer performs "SELECT ... INTO OUTFILE" for which NULL continues to be replaced by N
- the client-side "-ssl" and "-ssl-verify-server-cert" options have been removed
 - use "-ssl-mode=REQUIRED" instead of "-ssl-1" or "-enable-ssl"
 - use "-mode=DISABLED" instead of "-ssl-0", "-skip-ssl", or "-disable-ssl"
 - use "-mode=VERIFY_IDENTITY" instead of "-ssl-verify-server-cert" options

[more information
about removed features](#)

more information
about removed server
and status variables

checking for removals' is the key to success for a successful upgrade

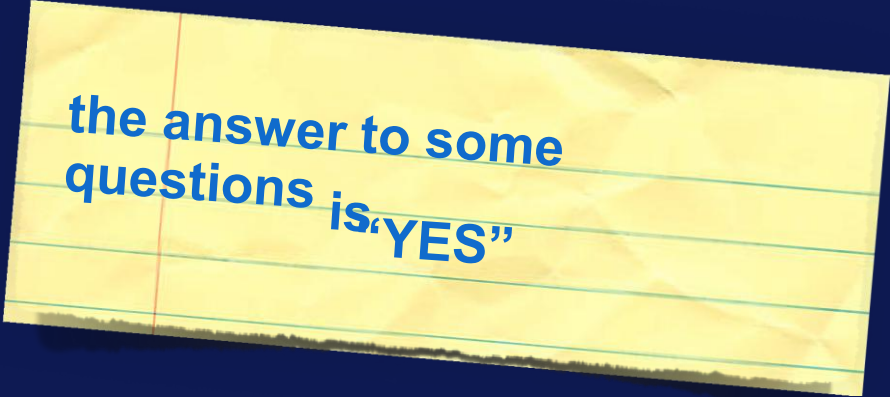
ΔΥ PERCONA

Bonus upgrade impediment

- Minor version upgrades in MySQL 8.0 are **unidirectional**
- There is no **easy** way of performing a quick minor version rollback



Q&A

A yellow sticky note with blue horizontal lines and a vertical red margin line on the left. The text is written in blue ink and is slightly tilted to the right.

the answer to some
questions is "YES"

Talk to me!



Arunjith Aravindan

<https://www.linkedin.com/in/arunjith-a-0b9b4652/>

<https://www.facebook.com/arunjith.a>

