MySQL Connector/Python Release Notes

Abstract

This document contains release notes for the changes in recent releases of MySQL Connector/Python.

For additional Connector/Python documentation, see MySQL Connector/Python Developer Guide.

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: 2026-04-14 (revision: 31062)

Table of Contents

Preface and Legal Notices

This document contains release notes for the changes in recent releases of MySQL Connector/Python.

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 Connector/Python 9

Changes in MySQL Connector/Python 9.7.0 (2026-04-22)

This release contains no functional changes, and is published to align its version number with that of the MySQL Server 9.7.0 release.

Changes in MySQL Connector/Python 9.6.0 (2026-01-21)

This release contains no functional changes, and is published to align its version number with that of the MySQL Server 9.6.0 release.

Changes in MySQL Connector/Python 9.5.0 (2025-10-22)

Note

These release notes were created with the assistance of MySQL HeatWave GenAI.

Functionality Added or Changed

- The connector is extended to include GenAI and Machine Learning capabilities. The package `mysql.ai` is added in this release and contains the following:

  - `genai`: GenAI package which exposes classes for embedding, LLM, and vector store. Embedding, LLM, and vector store are implemented as LangChain components making them easy to use and fully integrable with any of your LangChain pipelines.
  - `ml`: Machine learning package which exposes ML utilities. ML models are implemented as Scikit-Learn components making them easy to use and fully integrable with any of your Scikit-Learn pipelines.
  - `utils`: Utilities for AI-related helpers.

  See Streamlining GenAI and ML Application Development with the HeatWave Python SDK . (WL #17088)

- Added support for Python 3.14, and removed support for Python 3.9.

  From Python v3.14 and onwards, The module zstandard will be imported from Python's standard library instead of the externally installable couterpart. (WL #17138, WL #17045)

Changes in MySQL Connector/Python 9.4.0 (2025-07-22)

Note

These release notes were created with the assistance of MySQL HeatWave GenAI.

Functionality Added or Changed

- The extra option name for wheel package installation of dependencies for WebAuthn functionality has been changed from `fido2` to `webauthn`. (Bug #37806057)

- The pure-Python implementation of Connector/Python now supports Django `enum` objects and `ChoiceField` when executing queries using Django ORM. (Bug #116056, Bug #37047789)
- Connection pooling support for AsyncIO connections is now available.

  Thanks to Sean Stewart for contributing this patch. (Bug #114926, Bug #36733242)
- The version of Protobuf required by and installed with the X DevAPI has been updated to 5.29.4. (WL #16962)
- The LZ4 package, which is shipped with Connector/Python and installed with the X DevAPI by the `compression` installation option, has now been upgraded to version 4.4.4. (WL #16966)
- The version of OpenTelemetry required by and installed with the classic API by the `telemetry` installation option has been updated to 1.33.1. (WL #16963)

Bugs Fixed

- When making asynchronous connections to the server using Connector/Python, OCI plugin options like `oci_config_file` and `oci_config_profile` were not recognized and caused connections to fail. (Bug #38072835)
- Queries on text-type fields like Charfield and TextField failed when Django ORM filters (for example, `startswith`, `endswith`, `contains`, and `regex`) were used. It was because Connector/Python was using some deprecated MySQL commands behind the scene, and they have now been updated. (Bug #37820231)
- Binary data containing double percent signs (%%) was not correctly persisted in BLOB columns, but had the first percentage sign stripped. (Bug #118025, Bug #37859771)

  References: This issue is a regression of: Bug #37447394.
- Unlike when using the C extension, the pure-Python implementation could not convert NumPy float62 values into a MySQL type during a query. With this fix, conversion works, so that the C Extension and pure-Python implementations are now consistent in this respect. (Bug #117827, Bug #37774513)
- When using the Connector/Python C Extension with an unbuffered server-side cursor, `fetchmany(1)` failed to fetch more results after reading the second row in the result set. (Bug #117548, Bug #37627508)
- When using the Connector/Python C Extension, connections to a server using `sshtunnel` failed, with the connection hanging, or a segmentation fault. (Bug #108761, Bug #34844347, Bug #34950958)
- Installation of Connector/Python using the source distribution failed. With release 9.4.0, Connector/Python is now installable from source using `pip`. See Installing Connector/Python from a Source Distribution for details. (WL #16954)

Changes in MySQL Connector/Python 9.3.0 (2025-04-15)

Functionality Added or Changed

- Fixed the typing annotation of the `close()` function of `MySQLConnectionAbstract` and also made `disconnect()` an alias of `close()`, to improve clarity and consistency of the connection API.

Thanks, Parul Gupta, for the contribution. (Bug #117347, Bug #37541353)

- The deprecated cursors of the buffered raw type and all the named tuple types have now been removed from Connector/Python. Using them now results in a `ValueError` or `TypeError` depending on the invalid arguments passed. (WL #16327)
- The following methods of the Connector/Python API have been deprecated, with some of them replaced by new property methods:
  - MySQLConnection.get_server_version() replaced by the server_version property.
  - MySQLConnection.get_server_info() replaced by the server_info property.
  - MySQLConnection.is_connected() replaced by the connected property.
  - MySQLConnection.set_client_flags() replaced by the client_flags property.
  - MySQLConnection.set_unicode() replaced by the use_unicode property.
  - MySQLConnection.set_converter_class() replaced by the converter_class property.
  - MySQLCursor.fetchwarnings() replaced by the warnings property.
  - MySQLCursor.getlastrowid() replaced by the already existing lastrowid property.
  - MySQLCursor.stored_results() is simply deprecated for now.

  (WL #16752)

Bugs Fixed

- When executing a `LOAD DATA LOCAL INFILE` statement, Connector/Python now validates that the name of the client data file requested by the server (as part of the server's response) matches exactly the file name in the original statement sent by the client, or an error is raised. (Bug #37418436)
- A memory leak occurred when using prepared statements with the Connector/Python C extension. (Bug #37399636)
- When the C extension was used, exceptions with prepared statements caused a `MySQLInterfaceError` to be thrown. With this fix, the proper type of exception is thrown for each type of errors. (Bug #37275524)
- It was not possible to escape the string "%s" used in a query when it should not be treated as a parameter marker. This patch adds proper regular expression checks to escape the "%s" parameter marker when "%%s" is used in the query. (Bug #117087, Bug #37447394)
- The source distribution of Connector/Python on the PyPI repository was missing the `setup.py` file, which has now been added back to make the package buildable with `pip`. (Bug #113396, Bug #36098290)
- An unformatted error message was emitted for an unreachable host when a connection could not be established using a Unix socket. (Bug #113396, Bug #36098290)

Changes in MySQL Connector/Python 9.2.0 (2025-01-21)

Functionality Added or Changed

- Added two new connection options: `read_timeout` (time limit to receive a response from the server) and `write_timeout` (time limit to send data to the server). Both options default to `None`, which sets the timeout to indefinitely. (Bug #115788, Bug #36922645, WL #16381)
- Added the `read_timeout` and `write_timeout` options for both connections and cursors. These set the transaction timeout (sending and receiving data to and from the server), and default to `None` that sets the wait time to indefinitely. (WL #16381)
- The single and multiple statement execution mechanic was unified, when before a `multi` `cursor.execute()` option was required to execute multiple statements.

  Changes include removing the `multi` option and adding the optional `map_results` option, and adding the fetchsets() and nextset() methods. (WL #16285)

Bugs Fixed

- The configuration option parser was modified to use `ast.literal_eval()` instead of `eval()`. For additional information, see Connector/Python Option-File Support. (Bug #37145655)
- When passing in multiple statements, the connector did not properly traverse (and execute) each query when the last query was preceded by a comment. (Bug #36126909)

  References: This issue is a regression of: Bug #35710145.
- The mechanism to execute and retrieve multiple statements was improved. Changes include: the removal of `execute()`'s `multi` parameter, the addition of the `nextset()` and `fetchsets()` methods, and a new `map_results` option for the `execute()` method. For additional information, see Executing Multiple Statements. (Bug #35810050, WL #16285)
- Fixed two memory leaks coming from the `get_rows()` connection API command of the C-extension implementation. (Bug #115082, Bug #36702939)

Changes in MySQL Connector/Python 9.1.0 (2024-10-15)

Functionality Added or Changed

- Dropped support for building DEB packages in favor of using `pip` to install the connector on Debian-based platforms. (WL #16444)
- Added support for Python 3.13, and removed support for Python 3.8.

  Note that Python 3.13 enables `ssl.VERIFY_X509_STRICT` SSL validation by default, which means SSL certificates must now be RFC-5280 compliant when using Python 3.13 and higher. (WL #16306, WL #16307)
- Updated and expanded the metadata used by the PyPI project pages. (WL #16411)
- Added *OpenID Connect* support leveraging the new `authentication_openid_connect_client` client-side authentication plugin. *OpenID Connect* functionality is supported by MySQL Enterprise Edition Server 9.1.0 and later.

  The new `openid_token_file` connection option defines a path to a file containing the JWT formatted identity token. (WL #16341)

- Added GSSAPI 1.8.3 support, the version now used by default. (WL #16442)
- All client-side authentication plugins are now built while building the C-extension implementation, and bundled with both wheel and RPM packages. (WL #16452)

Bugs Fixed

- Specifying a TLS v1.3 cipher with the `tls_ciphersuites` connection option halted with a "No cipher can be selected." error. Now specifying `tls_ciphersuites` with TLSv1.3 is allowed with the C-extension implementation, but not enforced with the pure Python implementation where it's determined by the MySQL Server during TLS negotiation. (Bug #37055435)
- The `cursor.execute()` API command did not properly escape dictionary-based query parameterized strings when using the C-extension implementation of the connector. (Bug #37013057)
- An unformatted error message was emitted for an unreachable host when using the pure Python implementation. (Bug #115418, Bug #36765200)
- The connector sent two bytes for the collation in the response packet, instead of one. (Bug #114857, Bug #36577957)

Changes in MySQL Connector/Python 9.0.0 (2024-07-01)

Functionality Added or Changed

- The required Python Protobuf version installed with the X DevAPI connector was updated to version 4.25.3, as older versions could cause the connector to unexpectedly halt. Previously it allowed versions within the range 4.21.1 through 4.21.12. (WL #16284)
- Added a [telemetry] installation option that pulls in the OpenTelemetry API, SDK, and OTLP Exporter packages when installing Connector/Python. Optionally use `pip install mysql-connector-python[telemetry]` to install Connector/Python along with the OpenTelemetry v1.18.0 packages.

  Removed the bundled OpenTelemetry installation option that was initiated by passing in [opentelemetry] to pip, such as `pip install mysql-connector-python[opentelemetry]`. (WL #16283)
- Deprecated the following cursors: CMySQLCursorBufferedNamedTuple, CMySQLCursorNamedTuple, CMySQLCursorPreparedNamedTuple, CMySQLCursorPreparedRaw, MySQLCursorBufferedNamedTuple, MySQLCursorNamedTuple, MySQLCursorPreparedNamedTuple, and MySQLCursorPreparedRaw. (WL #16318)
- Debian, Solaris, and most RPM (except Enterprise Linux) packages are no longer available. Instead, using `pip` to manage Connector/Python is recommended.

Bugs Fixed

- With `mysql.connector.aio`, executing `cmd_change_user()` would emit a "Got packets out of order" error. (Bug #36664998)
- With *use_pure=True*, if the `sql_mode` mixed in `NO_BACKSLASH_ESCAPES` with another SQL mode, such as `STRICT_TRANS_TABLES`, then the client escaped strings as if `NO_BACKSLASH_ESCAPES` was not active. (Bug #36476195)
- The `aio.cursor.MySQLCursorBufferedRaw` method did not skip the conversion layer. For example, it would return a datetime.date entry as datetime.date instead of bytes. (Bug #36289767)

- For the DNS SRV functionality, changed the minimum dnspython dependency version to 2.6.1.

  Thanks to Michael Perrone for the contribution. (Bug #114985, Bug #36611371, WL #16350)

- With the C extension, the collation connection option was ignored. (Bug #114832, Bug #36570707)

Changes in MySQL Connector/Python 8.x

Changes in MySQL Connector/Python 8.4.0 (2024-04-30)

Functionality Added or Changed

- Added SASL authentication support for the C-extension using the GSSAPI (Kerberos) authentication method on Windows via the `authentication_ldap_sasl_client` plugin (see LDAP Pluggable Authentication). Support was already present for Linux and Windows, but on Windows for the pure Python implementation only. (WL #16053)
- The deprecated `authentication_fido` authentication plugin and associated `fido_callback` option were removed. Instead, use the `authentication_webauthn` auth_plugin. (WL #16127)
- Connector/Python is now ready to support the `VECTOR` data type when it becomes available with MySQL Enterprise Server. Limitation: the C extension won't support operations with VECTOR types for prepared statements until the MySQL C API supports it, which may be in MySQL 9.0.0. (WL #16164)
- Known limitation of this release: because the mysql_native_password authentication plugin is disabled by default as of MySQL Server 8.4.0, some unit tests may generate errors unless the plugin is enabled.

Bugs Fixed

- Expanded OpenTelemetry support to include query and protocol related API methods and properties for the connection object. (Bug #36227964)
- Fixed a memory leak in the X DevAPI interface C extension that occurred when adding a collection. (Bug #36167880)

Changes in MySQL Connector/Python 8.3.0 (2024-01-16)

Functionality Added or Changed

- The X DevAPI and Classic API were separated into their own packages. The `mysql-connector-python` package continues to install the Classic API, and a new `mysqlx-connector-python` package installs the X DevAPI. (WL #15836)
- For OpenTelemetry, added context propagation support for prepared statements with MySQL Commercial Server 8.3.0 or later with OpenTelemetry enabled. The C extension implementation must be built with `libmysqlclient` version 8.3.0 or later.

  Also, query attributes are now supported for prepared statements in the C extension when built with `libmysqlclient` version 8.3.0 or later. Previously, this functionality was supported only by the pure

Python implementation. Both implementations function only with MySQL Commercial Server 8.3.0 or later. (WL #15950)

- Added SASL authentication support using the GSSAPI (Kerberos) authentication method on Windows (support was already present for Linux) via the authentication_ldap_sasl_client plugin. This support is added only for the pure Python implementation. (WL #15985)
- Added support for asynchronous execution as per the Python Database API Specification v2.0 (PEP 249). The MySQL Connector/Python asyncio implementation allows non-blocking asynchronous interaction with a MySQL server using a new package named `mysql.connector.aio`. This new package only supports the pure Python implementation.

  The `mysql.connector.aio` package is fully compatible with the existing `mysql.connector` package implementation, and exposes the same interface but with the ability to use asynchronous execution. Asynchronous methods return asyncio coroutines that can await results. (WL #15523)

Bugs Fixed

- Binary strings were incorrectly converted to regular strings when using a prepared statement cursor with the pure Python implementation. (Bug #35912790)
- Django would report a deprecation warning about using `django.utils.timezone.utc`; updated code to use `datetime.timezone.utc` instead. (Bug #112366, Bug #35832148)
- Fixed `multi=True` usage when the query contained either code comments or stored procedures. (Bug #77548, Bug #21390859, Bug #35710145)

Changes in MySQL Connector/Python 8.2.0 (2023-10-25)

Functionality Added or Changed

- Added support for the WebauthN in FIDO (Fast Identity Online) authentication plugin (`authentication_webauthn`), and added a new `webauthn_callback` connection option.

  This deprecates the related authentication_fido implementation with its `fido_callback` connection option. Calling `fido_callback` now raises a deprecation error that suggests using `webauthn_callback` instead. (WL #15218)
- Added Python 3.12 support. (WL #15664)

Bugs Fixed

- For Django, the `raise_on_warnings` option was ignored if the `isolation_level` option was not set. (Bug #35755852)
- For OpenTelemetry, all spans generated on the server at connection time are now identified with the connector's connection span. (Bug #35733608)
- Handled a type-check issue discovered by the latest django-stubs release. (Bug #35547876)
- Fixed a memory leak in the X DevAPI interface C extension. (Bug #35141645)
- With the C extension, querying `information_schema.columns` returned byte strings instead of Unicode strings in the result values. (Bug #111427, Bug #35503506)

- With multiple simultaneous connections, the character set information is shared between connections which could be problematic if two connections were to different major MySQL server versions, such as MySQL 5.x and MySQL 8.x. (Bug #111426, Bug #35503377)

Changes in MySQL Connector/Python 8.1.0 (2023-07-18)

MySQL Connector/Python 8.1.0 is a new GA release version that supersedes the 8.0 series, and is recommended for use on production systems. This release can be used against MySQL Server version 5.7 and later.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- **Important Change:** With this release, we no longer supply 32-bit binaries for Connector/Python.
- Removed support for Python 3.7. (WL #15630)
- Removed DMG and MSI support; MSI and DMG installers are no longer supplied. The Wheel build distribution format remains fully supported, and is used by `pip`. (WL #15749)
- Added OpenTelemetry support. (WL #15629)

Bugs Fixed

- Connector/Python `setup.py` did not specify the minimum required Python version using `python_requires`, which meant that `pip` always installed the latest version of Connector/Python even if this was not compatible with the version of Python used on the system. (Bug #35338384)
- Fixed performance degradation when `multi=True` was used under certain conditions. This issue was introduced in Connector/Python 8.0.33. (Bug #35140271)

  References: This issue is a regression of: Bug #34655520.
- Closing a connection multiple times could cause a fatal deallocation error in Connector/Python 8.0.33. (Bug #111111, Bug #35425076)

  References: This issue is a regression of: Bug #35233031.
- Setting `compress=True` did not enable compression when using the C extension.

  Thanks to Daniël van Eeden for the contribution. (Bug #110879, Bug #35349093)
- An error was raised when setting the database name on a connection that included non-alphanumeric characters. This applied only to the pure Python implementation.

  Thanks to Brent Gardner for the contribution. (Bug #110469, Bug #35212199)
- A **`UnicodeDecodeError`** error was raised when using a complex query that produced a long field name alias. (Bug #110422, Bug #35278365)
- When switching to compressed mode while using the pure Python implementation, a compressed packet could have contained an incomplete MySQL packet.

  Thanks to Allen Long for the contribution. (Bug #93643, Bug #29115406, WL #15591)
- Updated the `protobuf` version requirement to versions 4.21.1 through 4.21.12, inclusive. Previously this was versions 3.11.0 through 3.20.3, inclusive. (WL #15672)

Changes in MySQL Connector/Python 8.0.33 (2023-04-18)

Functionality Added or Changed

- Adopted function annotation enforcement by `mypy`, which makes the `mysqlx` module compliant with PEP 8. (WL #15401)
- Added logger functionality to configure the logging system, which by default outputs severity level `WARNING` and higher to `sys.stderr`. (WL #15435)
- Added a `oci_config_profile` connection option to define a profile set in `oci_config_file`; the default value is `DEFAULT`. These options are for the `authentication_oci_client` plugin used with the Oracle Cloud Infrastructure (OCI) to support ephemeral key pairs and security tokens. (WL #15483)
- On Linux and macOS, added a script that builds and runs a Docker container to execute the test suite. Only a running MySQL server is needed to run this test suite; see `CONTRIBUTING.rst` for `./tests/docker/runner.sh` usage information. (WL #15528)

Bugs Fixed

- The C extension set `mysql_native_password` as the default authentication method; now the default set by MySQL, `caching_sha2_password`, is used instead. (Bug #35233031)
- It was possible for the exception closing an invalid connection to be interpreted as a `Bad Message` by the server; now the socket is closed instead of calling `CMD_QUIT`. (Bug #35015758)
- A deprecation warning is now raised if any of the commands `COM_FIELD_LIST`, `COM_REFRESH`, `COM_SHUTDOWN`, `COM_PROCESS_INFO`, or `COM_PROCESS_KILL` are used, since all of these are now deprecated by MySQL Server. (Bug #27489972)
- The prepared statement cursor lacked support for the following options: `dictionary`, `named_tuple`, and `raw`. (Bug #27359063)

  References: See also: Bug #23339387.
- The C extension truncated bytes that contained a `\x00` byte when using `NO_BACKSLASH_ESCAPES` mode in MySQL.

  As a workaround, use `converter_class=MySQLConverter` as a connection option. (Bug #109651, Bug #34984850)
- On macOS with the C extension, an exception was raised when using `authentication_oci_client`. In addition, some related libraries were not bundled, such as `libfido2`. (Bug #109430, Bug #34910625)
- Executing `executemany()` with `insert` statements that used `ON DUPLICATE KEY UPDATE` demonstrated poor performance; this was improved by fixing the associated regular expressions that match INSERT statements. (Bug #99575, Bug #31355895)
- Added the `SESSION_TRACK` constant to replace `SESION_TRACK`, which is now deprecated. (Bug #96347, Bug #30103652)
- Increased the data chunk size from 8 KB to 128 KB to improve performance for large inputs. (Bug #77789, Bug #21476351)

Changes in MySQL Connector/Python 8.0.32 (2023-01-17)

Functionality Added or Changed

- Removed `distutils` support, which is deprecated in Python 3.10 and removed in Python 3.12. (WL #14861)
- Adopted type hint enforcement for function and class attributes with `mypy`; this is compliant with PEP 8 for module mysql.connector. The integration includes a git pre-commit hook for mypy. (WL #15036)
- On Windows, added a `kerberos_auth_mode` connection option which is set to either "SSPI" (default) or "GSSAPI". This allows choosing between SSPI and GSSAPI at runtime for the `authentication_kerberos_client` authentication plugin on Windows. Previously, only the SSPI mode was supported on Windows. For general usage information, see Kerberos Pluggable Authentication. This option is ignored on other platforms such as Linux which support GSSAPI only.

  Limitation: GSSAPI cannot be used with the pure Python implementation on Windows using authentication with a username and password; this is a limitation of the C library used in the `python-gssapi` package used by the pure Python implementation of Connector/Python. (WL #15348)

Bugs Fixed

- **Microsoft Windows:** The Connector/Python MSI did not detect and install with Python 3.11. A workaround is to use `pip install mysql-connector-python` instead. (Bug #108911, Bug #34773422)
- Using `USE_TZ=True` in the Django settings would raise this exception: `ValueError: Not naive datetime (tzinfo is already set)`. (Bug #34727432)
- Removed debug messages that showed authentication data. (Bug #34695103)
- `protobuf` must now be version 3.11.0 through 3.20.3, inclusive. (Bug #34690501)
- Connecting to MariaDB failed with an unsupported character set because the default MySQL character set collation was specific to MySQL 8.0. Now the MySQL 5.7 character set is used by default instead, but is switched to a 8.0 character set if the queried server uses version 8.0. (Bug #34675508)
- Incorrect `MySQLCursor.statement` values were returned with `cursor.execute(query_string, multi=True)` under the following conditions: The query string contained two or more queries separated by a semicolon, and a query other than the first one used a literal or identifier containing an odd number of backticks, single quotation marks, or double quotation marks. (Bug #34655520)
- On Windows, changed the security support provider (SSP) from Kerberos to Negotiate. Negotiate selects either Kerberos or NTLM as the SSP. (Bug #34556157)
- When using a prepared cursor, a DATETIME column containing 00:00:00 as the time value caused a Python *date* object was returned instead of a *datetime*.

  Thanks to Rong Zhao for the contribution. (Bug #108733, Bug #34689812)
- The `MySQLCursor.executemany()` method failed to batch insert data because the regular expression (RE) sentinel did not detect batch cases correctly; this meant using a one-on-one insert instead, which led to poor performance.

  Thanks to Alex Cazacu for the contribution. (Bug #108145, Bug #34499578)

- Added a new `init_command` connection option to specify a query which is executed immediately after the connection is established.

  Thanks to Sander van de Graaf for the contribution. (Bug #108076, Bug #34467201)

- Russian characters were not handled correctly by the C extension version of the X DevAPI driver. We fix this by encoding string values to their byte string representations before sending them to `protobuf`. (Bug #106585, Bug #33904362)

- Fetching results from a prepared cursor using the pure Python implementation failed when a `VARBINARY` column contained bytes that could not be decoded. Such bytes are now returned directly if they cannot be decoded.

  Thanks to Naoki Someya for the contribution. (Bug #96280, Bug #30089671)

- Removed multiple reference leaks and removed redundant code.

  Thanks to Vilnis Termanis for the contribution. (Bug #90862, Bug #28020811)

- Cursors (both the pure Python and C extension versions) use a single `SELECT` to retrieve procedure result parameters following a procedure call, but one `SET` was used per parameter when setting the input parameters. This is now optimized by always using a single SET call, even for multiple parameters.

  Thanks to Vilnis Termanis for the contribution. (Bug #89345, Bug #27426532)

- Improved warning handling throughout the Connector.

  Thanks to Vilnis Termanis for the contribution. (Bug #82366, Bug #24364556)

- Added a `MySQLCursorPreparedDict` class option that is similar to `MySQLCursorPrepared` except that the former returns a fetched row as a dictionary in which column names are used as keys while the latter returns a row as a traditional record (a tuple).

  Thanks to Luke Weber for the contribution. (Bug #81573, Bug #23339387)

- It is now possible to use dictionaries as parameters in prepared statements using the `%(param)s` format as placeholders.

  Thanks to Luke Weber for the contribution. (Bug #81572, Bug #23342572)

- Using `MySQLConverter.escape()` on datetime objects raised the error `TypeError: an integer is required`. Now this method no longer attempts to escape values which are not bytes or string types. (Bug #80679, Bug #22906307)

- Not all parameters were added to the `INSERT` statement when using `INSERT IGNORE` with `cursor.executemany()`.

  Thanks to Takashi Ichii for the contribution. (Bug #75824, Bug #20504804)

Changes in MySQL Connector/Python 8.0.31 (2022-10-11)

Functionality Added or Changed

- Added support for Python 3.11. (WL #15156)

Bugs Fixed

- The pure Python implementation unexpectedly delivered `TEXT` columns having binary (`_bin`) collations as byte objects. (Bug #33987119)
- Changed the `warning_count` property from private to public in the `cursor` class; this is used to retrieve the number of warnings generated by the previously executed operation. (Bug #27634910)
- When using buffered cursors with the C extension, it was possible `cursor.executemany()` to return `InternalError: No result set available` for `SELECT` statements. (Bug #21529893)
- The `MySQLConnection.set_charset_collation()` method returned an **`UnboundLocalError`** exception when given a empty character set name; now the default character set is used in such cases. (Bug #21402805)
- On macOS, compiling the C Extension implementation assumed that `gcc` and `g++` were the default compilers. Now the compilation process checks the `CC` and `CXX` environment variables for the use of `clang`.

  Thanks to Jonathan Ringer for the contribution. (Bug #107841, Bug #34373612)
- Selecting binary data that began with `0x00` returned an empty string. (Bug #107568, Bug #34283402)
- Calling a stored procedure by its fully qualified name (of the form *`database_name.procedure_name`*) generated a **`ProgrammingError`** (SQL syntax) error. (Bug #107406, Bug #34217492)
- The binary protocol halted unexpectedly when passed a time value of `0`, while the expected behavior is to receive a `00:00:00` time when such a payload is passed to it. (Bug #91974, Bug #28491115)
- Aligned the exception types raised by pure Python and the C extension when assigning a non-existent database or when executing an invalid query using the `connection.info_query()` method. (Bug #91315, Bug #28295478)
- Destroying a connection object now raises a **`ProgrammingError`** exception stating the cursor is not connected. Previously, it raised a **`ReferenceError`** exception about weakly referenced objects, which related to how cursors contain a weak reference to the connection. (Bug #76181, Bug #21463298)

Changes in MySQL Connector/Python 8.0.30 (2022-07-26)

Functionality Added or Changed

- Added support for OpenSSL 3.0. (WL #14815)
- Added or renamed collations to align with MySQL Server 8.0.30. This includes adding support for language-specific `utf8mb4` collations and renaming all existing `utf8_` collations to `utf8mb3_`. In addition, `utf8` is now an alias for `utf8mb4`. Support for MySQL 5.7 collations is preserved for connections to a MySQL 5.7 server. (WL #15212)
- Implemented an authentication mechanism to instantiate only required authentication plugins according to the authentication handshake performed with MySQL Server. Previously, all available authentication plugins (and their associated third-party modules) were loaded upon initialization. (WL #14822)
- Improved PEP 7 and PEP 8 style guideline enforcement by adding `Black` and `Isort` tooling for PEP 8, and `clang-format` for PEP 7. The integration includes git pre-commit hooks.

In addition, `Pylint` was also tested and integrated with a git pre-commit hook. (WL #15035, WL #15137)

- Removed support for the `bdist_deb`, `bdist_macos`, and `bdist_rpm` build targets from `setup.py`. Instead, use the packages built by Oracle Release Engineering. (WL #15176)

Bugs Fixed

- Empty strings (`""`) in the collection fields used in methods like `set()` that are chained with `modify()`are no longer allowed; now only the dollar sign (`$`) is allowed when trying to replace the entire document. (Bug #34260344)
- The required version of `protobuf` is now 3.11.0 through 3.20.1, inclusive, due to a breaking change introduced in Python `protobuf` 4.21.0. (Bug #34231226)
- C extension and pure Python cursor types are not interchangeable when using connection objects that are not of the same type; now we check for supported cursor classes and raise a **`ProgrammingError`** if the cursor is not of a compatible type. (Bug #34223015)
- The C Extension did not function with the `NO_BACKSLASH_ESCAPES` SQL mode enabled. We fix this by calling the `mysql_real_escape_string_quote()` C API function instead of `mysql_real_escape_string_quote()`. Prior to upgrading to this release, you can work around this problem by using the pure Python implementation instead (`use_pure=True`). (Bug #107434, Bug #34228442)
- For Django, allow setting the connection isolation level with an `isolation_level` `OPTIONS` entry in the Django `DATABASES` configuration. (Bug #107174, Bug #34127959)
- The `connect()` method's `failover` argument now accepts a tuple, as was documented, when previously it accepted only a list of dictionaries.

  Thanks to Ville Skyttä for the patch. (Bug #106632, Bug #33923516)
- For Django, added the missing `DatabaseIntrospection` `_parse_constraint_columns()` method. (Bug #105993, Bug #33827760)
- Fixed potential rounding errors when using arithmetic expressions with decimal data types; decimal values are no longer quoted. (Bug #92790, Bug #28821983)
- Disabled SSL usage with Unix socket connections. (Bug #91552, Bug #92260, Bug #28295504, Bug #28880051)

Changes in MySQL Connector/Python 8.0.29 (2022-04-26)

Functionality Added or Changed

- When a connection disabled SSL Mode using `ssl_disabled=True` (or with `ssl-mode='DISABLED'` for X Protocol connections), specifying other TLS or SSL connection options caused Connector/Python to throw an error. For example, defining `tls_version` with `ssl_disabled=True` raised such an error. Now, disabling SSL now means that additional TLS and SSL connection options are ignored. (WL #14852)
- The Connector/Python C extension now supports FIDO pluggable authentication, and adds a new `fido_callback` connection option. (WL #14860)

- The version of `protobuf` used by the C extension is updated to version 3.19.4. (WL #15080)
- Removed support for Python 3.6.

  On EL7, EL8, and SUSE Linux: A `python3-protobuf` RPM package is not available for Python 3.8 on these platforms, so the dependency was removed from the RPM specification; instead this package must be installed manually with `pip install protobuf` or similar. (WL #14824)
- Expanded Kerberos Authentication Protocol support to include Windows clients using the pure Python implementation of Connector/Python. (WL #14665)
- Django support now allows a custom class for data type conversion in the Django backend, typically used to convert non-default data types. This is implemented as a subclass of `mysql.connector.django.base.DjangoMySQLConverter`. (WL #14679)

Bugs Fixed

- **Microsoft Windows:** When using the C extension with `prepared=True`, `BIGINT` data was seen as `MYSQL_TYPE_LONG` instead of `MYSQL_TYPE_LONGLONG`, which raised overflow errors with values greater than 2147483647. (Bug #96588, Bug #105254, Bug #30203754, Bug #33481203)
- Upon connecting to the server, Connector/Python executed a number of `SHOW [SESSION] VARIABLES` statements to retrieve system variable values. Such statements involve locking in the server, so they are now avoided in favor of `SELECT @@var_name`. (Bug #33861549)
- When using the C extension, an expression used as a column without an alias raised a **`UnicodeDecodeError`** error. (Bug #33747585)
- The deprecated `utf8mb3` character set is now recognized because the `utf8` alias shows `utf8mb3` in the Information Schema and `SHOW` statements in MySQL 8.0.28 and later. For additional information, see The utf8 Character Set (Deprecated alias for utf8mb3). (Bug #33729842)
- With *consume_results=True*, the C extension implementation threw an error with some methods such as cmd_refresh(), reset_session() and commit() if there were pending results to fetch. Now, unread results are handled and consumed. (Bug #21528553)
- When connecting to MySQL Server versions previous to 5.7.35, servers failed to execute `change_user` commands due to the default collation used by Connector/Python, a problem which affected connection pooling functionality by raising an exception when closing a pooled connection. Now Connector/Python reconnects instead of emitting the exception. (Bug #104569, Bug #33203161)
- Previously, unsuccessful decoding a byte array raised a **`UnicodeDecodeError`** error. Now in such the raw bytes or byte array is returned instead.

  Thanks to Meik Milevczik for the contribution. (Bug #93065, Bug #28877987)
- Added support for Decimal parsing. Thanks to Luke Weber for the patch. (Bug #81571, Bug #23338623)
- Added a context manager to open files in the optionfiles module to be closed correctly. Thanks to Vilnis Termanis for the patch. (Bug #81519, Bug #23324748)
- Attempts to insert Python `ByteArray` data into binary table columns failed when using the c-extension implementation of Connector/Python. (Bug #77822, Bug #21498719)
- Added support for NaN in the float to MySQL conversion; it's now set to *None*. (Bug #74934, Bug #20065830)

Changes in MySQL Connector/Python 8.0.28 (2022-01-18)

Deprecation and Removal Notes

- The TLSv1 and TLSv1.1 connection protocols were previously deprecated in Connector/Python 8.0.26 and support for them is removed starting with this release. Instead, use TLSv1.2 or TLSv1.3.

  Note: the `tls_versions` connection option now defaults to '["TLSv1.2", "TLSv1.3"]' when before it was '"TLSv1", "TLSv1.1", "TLSv1.2", "TLSv1.3"'. (WL #14814)

Functionality Added or Changed

- Connector/Python can now establish connections using Multi-Factor Authentication (MFA), such that up to three passwords can be specified. The new `password1`, `password2`, and `password3` connection options are available for specifying the first, second, and third MFA passwords, respectively. The `password1` option is a synonym for the existing `password` option. (WL #14667, WL #14720)
- Added Python 3.10 support. (WL #14813)

Bugs Fixed

- A MySQL decimal field type was returned as a string instead of a decimal type. (Bug #33486094)
- Fixed compiler warnings that were shown when building Connector/Python. (Bug #33410592)
- Fixed the JSON conversion class in CMySQLConnection; and fixed the related exception raised when using a conversion class in a C extension connection object. (Bug #33409819, Bug #105012)
- Sent invalid 'params' argument definitions in execution methods to the server instead of raising an error; and this generated an error unrelated to the invalid parameter definition. (Bug #27358941)

Changes in MySQL Connector/Python 8.0.27 (2021-10-19)

Functionality Added or Changed

- In Connector/Python 8.0.26, the capability was introduced for applications that use the classic MySQL connections for accounts that use the `authentication_kerberos` server-side authentication plugin, provided that the correct Kerberos tickets are available or can be obtained from Kerberos. That capability was available on client hosts running Linux only. It is now available on client hosts running Windows.

  For more information about Kerberos authentication, see Kerberos Pluggable Authentication. (WL #14664)
- Added *wheel* packages for the commercial edition. (WL #14669)
- Improved the DMG package as previously it assumed that XCode was installed to provide the Python installation. Additionally, there are now two separate DMG files; one for x86-64 and another for ARM. These installers now provides an option to choose which Python version to use, and defaults to the version provided by XCode. (WL #14688)

- Added a new `converter_str_fallback` connection option that allows enabling the conversion to str of value types not supported by the Connector/Python converter class, or by a custom converter class. It defaults to False. (WL #14689)
- `Known limitation`: Python 3.10 is not yet supported, and therefore operating systems including Python 3.10 (such as Fedora 35) do not support this or earlier versions of Connector/Python.

  This issue was resolved in 8.0.28.
- Applications that use legacy MySQL connections can now establish connections without passwords for accounts that use the `authentication_oci` server-side authentication plugin, provided that the correct configuration entries are available to map to one unique user in a specific Oracle Cloud Infrastructure tenancy.

  To ensure correct account mapping, the client-side Oracle Cloud Infrastructure configuration must contain a fingerprint of the API key to use for authentication (`fingerprint` entry) and the location of a PEM file with the private part of the API key (`key_file` entry). Both entries should be specified in the `[DEFAULT]` profile of the configuration file.

  Unless an alternative path to the configuration file is specified with the new `oci_config_file` connection option, the following default locations are used:

  - `~/.oci/config` on Linux or Posix host types
  - `%HOMEDRIVE%%HOMEPATH%/.oci/config` on Windows host types

  If the MySQL user name is not provided as a connection option, then the operating system user name is substituted. Specifically, if the private key and correct Oracle Cloud Infrastructure configuration are present on the client side, then a connection can be made without giving any options. (WL #14710)

Bugs Fixed

- Using the C-extension, attempting to connect with chained SSL certificates using ssl_verify_identity=True did not function. The workaround was to use the pure Python implementation. (Bug #33177337)
- Printing an mysqlx.result.row object output the generic representation of a class in Python rather than the string value. (Bug #28641350)

Changes in MySQL Connector/Python 8.0.26 (2021-07-20)

Deprecation and Removal Notes

- The TLSv1.0 and TLSv1.1 connection protocols now are deprecated and support for them is subject to removal in a future Connector/Python version. Using them emits a 'DeprecationWarning' level error. (WL #14542)

Functionality Added or Changed

- Allow running unit tests against an external server that is not controlled by the testing framework. (Bug #80135, Bug #22594547, WL #14634)

- Connector/Python now supports Query Attributes when they have been enabled on the server, see Query Attributes for details. New Connector/Python cursor methods for this functionality are `add_attribute(name, value)`, `clear_attributes()`, and `get_attributes()`. Functionality works on both prepared and non-prepared statements. (WL #14237)
- Connector/Python now implements the authentication_kerberos_client plugin to support Kerberos authentication for classic MySQL protocol connections. (WL #14440)

Bugs Fixed

- Removed Django's MySQLdb module dependency. (Bug #32947160)
- Fixed the option file parser when using the include directive in configuration files with override sections. Thanks to Garen Chan for the patch. (Bug #32838010, Bug #103546)
- On Windows, updated the MSI installer to distribute additional libraries needed by `authentication_ldap_client.dll`; this includes `libsasl.dll`, `saslSCRAM.dll`, and `libcrypto-1_1-x64.dll`. (Bug #32789076)
- If a document id provided for replacement in the Collection.replace_one() or Collection.add_or_replace_one() methods contains an "_id" different from the original document, then the "_id" definition was ignored. Now, an error is raised. (Bug #32778827)
- The X DevAPI implementation converted binary types to strings instead of bytes. (Bug #32623479)
- The CEXT implementation raised an exception when a SELECT statement fetched an INT ZEROFILL data type that became left 0-padded. Now, the ZEROFILL_FLAG flag is checked and handled as expected. (Bug #31528783, Bug #99955)

Changes in MySQL Connector/Python 8.0.25 (2021-05-11)

This release contains no functional changes, and is published to align its version number with that of the MySQL Server 8.0.25 release.

Changes in MySQL Connector/Python 8.0.24 (2021-04-20)

Functionality Added or Changed

- Removed Python 2.7 and 3.5 support, and added Python 3.9 support. (Bug #89480, Bug #32144255, Bug #32192619, Bug #32001787, Bug #27477376, WL #14027, WL #14239, WL #14240)
- Improved server disconnection handling of X Protocol connections now creates a log entry and returns an error message, as needed, after Connector/Python receives a connection-close notice from the server. Connector/Python detects three new types of warning notices.

  Connection idle notice. This notice applies to a server connection that remains idle for longer than the relevant timeout setting. Connector/Python closes the connection when it receives the notice in an active session or while a new session is being created. An attempt to use the invalid session returns the "`Connection closed. Reason: connection idle too long`" error message.

  Server shutdown notice. If a connection-close notice is received in a session as a result of a server shutdown, Connector/Python terminates the session with the "`Connection closed. Reason:`

`server shutdown`" error message. All other sessions that are connected to the same endpoint are removed from the pool, if connection pooling is used.

Connection killed notice. If the connection being killed from another client session, Connector/Python closes the connection when it receives the notice in an active session or while a new session is being created. An attempt to use the invalid session returns the "`Connection closed. Reason: connection killed by a different session`" error message.

(WL #14212, WL #13492)

- If a classic MySQL protocol connection experiences a server timeout, Connector/Python now reports more precise disconnection information from the server. (WL #14424)

Bugs Fixed

- On Windows, improved the MSI's "Destination Folder" dialogue to include what's being installed in this folder, such as Connector/Python example files. (Bug #32532744)
- For the C-extension, executing prepared statements emitted errors when placeholders were defined without associated parameters. Now they are not executed. (Bug #32497631)
- For prepared statements any type or argument was accepted, which could produce undesired results. Now the use of list or type objects for the argument is enforced, and passing in other types raise an error. (Bug #32496788)
- Added Django 3.2 support while preserving compatibility with Django 2.2, 3.0, and 3.1. (Bug #32435181)
- Added context manager support for pooled connections; a feature added to standard connections in 8.0.21. (Bug #32029891)
- Replaced the deprecated PyUnicode_GetSize with PyUnicode_GET_LENGTH to fix the casting of Python's unicode to std::string. (Bug #31490101, Bug #99866)
- Binary columns were returned as strings instead of 'bytes' or 'bytearray'. (Bug #29622520, Bug #30349010, Bug #30416704, Bug #94944, Bug #96999, Bug #97177, Bug #97723)
- Prepared statements without parameters would violate the MySQL protocol by sending unnecessary extra bytes. (Bug #101479, Bug #32120659)

Changes in MySQL Connector/Python 8.0.23 (2021-01-18)

Deprecation and Removal Notes

- Deprecated Python 2.7 support; a warning was added to describe Python 2.7 support being removed in Connector/Python 8.0.24. (WL #14238)

Functionality Added or Changed

- Previously, Connector/Python added client support for the MySQL Enterprise Edition SASL LDAP authentication plugin with SCRAM-SHA-1 as an authentication method. Connector/Python now also supports SCRAM-SHA-256 as an alternative authentication method for classic MySQL protocol connections. SASL-based LDAP authentication does not apply to clients running macOS. (WL #14263)

- Added SASL authentication protocol support using the GSSAPI (Kerberos) authentication method for the pure python implementation (this is not available for the c-ext version of Connector/Python).

  This functionality requires the GSSAPI pypi module, which provides both low-level and high-level wrappers around the GSSAPI C libraries. The GSSAPI pypi module requires MIT kerberos installed opn the system to function and request tickets to authenticate Connector/Python with the MySQL server when the user is IDENTIFIED WITH authentication_ldap_sasl and the authentication_ldap_sasl plugin is configured to use the GSSAPI mechanism.

  This also adds a new `krb_service_principal` option, which must be a string in the form "primary/instance@realm" such as "ldap/ldapauth@MYSQL.COM" where "@realm" is optional. The "@realm" defaults to the default realm, as configured in the `krb5.conf` file. (WL #14213)

Bugs Fixed

- Removed the MySQL client-plugins dependency, although it's recommended. This plugin allows MySQL to use its newer authentication methods, thus it's required for connecting to accounts using the new authentication methods such as `caching_sha2_password`. (Bug #32114921, Bug #101464)
- Fixed the AttributeError raised when getting the connection ID from a closed CMySQLConnection. (Bug #31882419, Bug #100825)
- Fixed support for named tuple (an invalid exception was generated) and dictionary cursors (incorrect type of result was returned). (Bug #29195610)
- Fixed cursor.fetchone() and cursor.fetchmany() to comply with PEP 249, which specifies that an exception must be raised if the previous call to cursor.execute*() does not produce any result set or no call was issued yet. (Bug #26834307, Bug #87815)
- Fixed the microsecond conversion from MySQL datetime to Python datetime when using fractional values with the C extension enabled. For example, 2016-10-20 15:40:23.8 from MySQL resulted in datetime.datetime(2016, 10, 20, 15, 40, 23, 8) instead of datetime.datetime(2016, 10, 20, 15, 40, 23, 800000). Thanks to Vilnis Termanis for the patch. (Bug #24938411, Bug #31556777, Bug #83479, Bug #100034)
- Updated associated MySQL server error messages and codes to the latest version. (Bug #102220, Bug #32370045, WL #14215)

Changes in MySQL Connector/Python 8.0.22 (2020-10-19)

Functionality Added or Changed

- Added Django 3.0 support while preserving compatibility with Django 2.2. Removed support for Django 1.11 with Python 2.7. (Bug #91699, Bug #99069, Bug #28367300, Bug #30835272, WL #13380)
- Refactored the Connector/Python build system by removing artifacts of old implementations, improved debugging, and now statically link the C extensions. This also exposes the distutils commands, to allow the end-user build packages. (Bug #79766, Bug #81203, Bug #81242, Bug #87493, Bug #99923, Bug #23194976, Bug #23208553, Bug #26660893, Bug #31529348, WL #13997)
- Previously, the client-side `mysql_clear_password` authentication plugin was not supported. Now, it is permitted to send passwords without hashing or encryption by using `mysql_clear_password` on the client side together with any server-side plugin that needs a clear text password, such as for

LDAP pluggable authentication. Connector/Python returns an exception if the `mysql_clear_password` plugin is requested but the connection is neither encrypted nor using Unix domain sockets. For usage information, see Client-Side Cleartext Pluggable Authentication. (WL #13994)

- Connections made using the MySQL Enterprise Edition SASL LDAP authentication plugin now are supported on Windows and Linux, but not on macOS. Connector/Python implements the `SCRAM-SHA-1` authentication method of the SASL authentication protocol. (WL #14110)
- The new compression-algorithms connection option sets the order by which supported algorithms are negotiated and selected to send compressed data over X Protocol connections. The algorithms available are specified by the server and currently include: lz4_message, deflate_stream, and zstd_stream. Supported algorithm aliases are lz4, deflate, and zstd. Unknown or unsupported values are ignored.

  Example usage:

```
session = mysqlx.get_session({
    "host": "localhost",
    "port": 33060,
    "user": "root",
    "password": "s3cr3t",
    "compression": "required",
    "compression-algorithms": ["lz4', "zstd_stream"]
})
```

  (WL #13995)
- For enhanced security of the existing `allow_local_infile` connection string option, the new `allow_local_infile_in_path` option allows restricting LOCAL data loading to files located in this designated directory. (WL #14098)
- The pure Python and C extension implementations were combined into a single package; this applies to both DEB and RPM packages. (WL #13985)

Bugs Fixed

- Fixed a memory leak in the C-extension implementation when using the Decimal data type. Thanks to Kan Liyong for the patch. (Bug #31335275, Bug #99517)
- Copyright and License headers were missing in the Python modules generated by protoc. (Bug #31267800)
- When creating an index on a collection, if a collation was specified but the field is not of the type TEXT, then an error message was generated with a wrong field type. It'd always report it as GEOJSON. (Bug #27535063)
- The reset connection command was missing from the C-extension implementation, which is required to reuse a connection from the pool. As such, connection pooling is now allowed with the C-extension implementation. (Bug #20811567, Bug #27489937)

Changes in MySQL Connector/Python 8.0.21 (2020-07-13)

Functionality Added or Changed

- On macOS, the official supported Protobuf version is versions before 3.12.0. (Bug #31390263)

- Implemented context managers to define the runtime context to establish when executing a with statement. It was added to the Connection and Cursor objects in the classic protocol, and to Session in the X DevAPI. Thanks to WEN-FENG SHIH for the patch. (Bug #19586444, Bug #71663, Bug #28779784, Bug #89113, WL #13847)

- Added compression support to X Protocol connections. Supported compression algorithms are *zlib/deflate* and *lz4*. Specifically, the supported algorithms (in order of preference) are lz4_message and deflate_stream. The compression threshold is set at 1000 bytes.

  A new `compress` X DevAPI connection option accepts either *required*, *disabled*, or *preferred* (default).

  - *preferred*: If Connector/Python and the server cannot reach consensus on algorithm or styles, then no compression is used; and this is logged.
  - *required*: Like *preferred*, except the connection is terminated with an error if the connector and server are unable to reach agreement.
  - *disabled*: Compression is not used.

  (WL #12501)

- *Document Store:* Connector/Python now provides JSON schema validation for a collection to enforce a certain structure that documents must adhere to before they are permitted to be inserted or updated. Schema validation is performed by the server, which returns an error message if a document in a collection does not match the schema definition or if the server does not support validation.

  The `schema.create_collection` method added a new validation parameter as a dictionary or string representation of a JSON schema specification. The level of enforcement (`off` or `strict`, `strict` by default) and schema definition are specified as per this example:

```
coll = schema.create_collection("longlang", validation={
    "level": "strict",
    "schema":   {
        "id": "http://json-schema.org/geo",
        "$schema": "http://json-schema.org/draft-06/schema#",
        "description": "A geographical coordinate",
        "type": "object",
        "properties": {
            "latitude": {
                "type": "number"
            },
            "longitude": {
                "type": "number"
            }
        },
        "required": ["latitude", "longitude"]
    }
})
```

  In addition, a new `schema.modify_collection` method permits the schema validation of an existing collection to be reset. The `validation` collection option must include either a modified `level` value or `schema` value, or both. (WL #13059)

Bugs Fixed

- The connect_timeout option applied to all blocking socket operations but now properly only applies to the timeout when establishing the connection. (Bug #30996790)

- In X DevAPI implementation, an error was raised when using the fractional part in DATETIME types. The error: "ValueError: Datetime mapping scenario unhandled" (Bug #30950184)

- Because MySQL stores TEXT types as BLOB and JSON as LONGBLOB, the TEXT and JSON types are now converted to *str* and the rest of the BLOB types as bytes. Previously, as an example, a column of type TEXT that only contained digits was read as type=integer by Connector/Python. (Bug #29808262, Bug #95437)
- Connector/Python assumed that MySQL libraries were under lib/ when compiling the C extension, but now uses the mysql_config flags which adds the correct include and libraries paths. Thanks to Daniël van Eeden for the patch. (Bug #29181907, Bug #93846)
- Attempting to change values using the Table object would yield an "Unknown column 'doc' in 'field list'" exception when using the X DevAPI. (Bug #28627768, Bug #27602636)

Changes in MySQL Connector/Python 8.0.20 (2020-04-27)

Functionality Added or Changed

- For X DevAPI applications, when creating a new connection, if the connection data contains several target hosts that have no explicit priority assigned, the behavior of the failover logic now is the same as if all those target hosts have the same priority. That is, the next candidate for making a connection is chosen randomly from the remaining available hosts. Alternatively, the new `priority` attribute optionally sets the priority for each host. If two hosts have the same priority then one is chosen at random. (WL #13334)

Bugs Fixed

- Removed the `dnspython` dependency, and now raise an exception if DNS SRV functionality is used without it installed. (Bug #30764641, Bug #98225)
- The fetchmany() method failed to function properly when the last fetch was not a full batch; and would hang. Thanks to Bruce Feng for the patch. (Bug #29847862, Bug #30608703, Bug #30250184, Bug #91971, Bug #93510, Bug #97830)
- The minimum Protobuf prerequisite version changed from 3.6.1 to 3.0.0.
- To make RPM and DEB package installation seamless, implicit dependencies on Python protobuf 3.0.0 or higher was added to the packages for most platforms. But as there are no Python protobuf versions fulfilling that requirement on EL7, SLES 12, nor Ubuntu 16.04, their pure Python packages depend on the C extension package to provide the same functionality.

  These dependencies are handled automatically by tools like YUM and APT. Users installing using "rpm" or "dpkg" needs to install both the pure Python and C extension packages.

  

  Note

  Note that this means that the connection option "use_pure=True" can't be used on EL7, SLES 12, nor Ubuntu 16.04 unless the system has Python protobuf 3.0.0 or higher installed.

Changes in MySQL Connector/Python 8.0.19 (2020-01-13)

Functionality Added or Changed

- Added DNS SRV support.

  To automatically resolve any SRV record available in a target DNS server or service discovery endpoint, use the `mysqlx+srv` scheme+extension in a X DevAPI connection string, or `mysqlx+srv` for the classic protocol, or by enabling the dns-srv=True (or dns_srv=True) connection option when using keyword arguments or dictionaries. (WL #13372)

- Added two new connection options that evaluate during the TLS handshake to restrict the negotiated TLS protocols and ciphers; along with those configured on the server that can further restrict the final choices. The new options are `tls-versions` to define the allowed TLS protocol versions, and `tls-ciphersuites` for the allowed cipher suites. These definitions are comma-separated, and accepted by the *getSession()* and *getClient()* methods.

  `tls-versions`: accepts one or more of the following: TLSv1, TLSv1.1, TLSv1.2, and TLSv1.3. Other values generate an error. Example usage: *mysqlx://myserver/db?tls-versions=[TLSv1.2,TLSv1.3]*

  `tls-ciphersuites`: accepts IANA cipher suite names, as listed on IANA's TLS Cipher Suites page. Unsupported or unknown values are ignored. Example usage: *mysqlx://myserver/db?tls-ciphersuites=[TLS_DHE_PSK_WITH_AES_128_GCM_SHA256, TLS_CHACHA20_POLY1305_SHA256]* (WL #12738)

- The internal X Protocol namespace changed from *xplugin* to *mysqlx*. MySQL Server removed xplugin namespace support in v8.0.19; for Connector/Python this means:

  - With Connector/Python v8.0.19 and higher, some X DevAPI Protocol operations do not function with MySQL Server 8.0.18 and lower, operations such as Schema.create_collection(), Schema.get_collections(), Schema.get_tables(), and Collection.create_index().
  - Connector/Python 8.0.19 can connect to MySQL Server 8.0.18 and lower, as both the 'xplugin' (with deprecation warnings) and 'mysqlx' namespaces can be used.

  (WL #13531)

Bugs Fixed

- Fixed the reserved SSL authentication filed; it changed from 23 to 22. Thanks to Qianqian Bu for the patch. (Bug #30270760, Bug #96770)

  References: This issue is a regression of: Bug #29855733.

- Fixed *LOAD DATA INFILE LOCAL* handling; the file handle was not closed. Thanks to Micah Gale for the patch. (Bug #29417117, Bug #94496)

Changes in MySQL Connector/Python 8.0.18 (2019-10-14)

Functionality Added or Changed

- Connector/Python connections now set CAN_HANDLE_EXPIRED_PASSWORDS to indicate it can handle sandbox mode for expired passwords. This indicates that Connector/Python does not execute SET commands by a connection with an expired password, an operation that's disallowed by MySQL Server 8.0.18 and higher. (Bug #95263, Bug #29743839, WL #13335)

- Connector/Python failed to install using the macOS DMG. (Bug #93492, Bug #29018765)
- Added Python 3.8 support. (WL #13194)
- On Windows, added platform dependent MSI installers that install and update Connector/Python for all supported Python versions on the system. Downloading and installing separate packages for each version is no longer required. (WL #13330)

Bugs Fixed

- The `/usr/lib/mysqlx` folder was not created after executing `setup.py` from commercial packages. (Bug #29959309)
- A table scan for a float using the C Extension caused a memory leak. (Bug #29909157)
- Added `read_default_file` as an alias for `option_files` to increase MySQLdb compatibility. (Bug #25349794, Bug #84389)
- Connector/Python 8.0.17 does not properly negotiate the highest TLS protocol version supported by both the client and server. As such, because MySQL 5.6/5.7 platform packages (DEB and RPM) include YaSSL prior to 5.6.45/5.7.27, and YaSSL only supports up to TLS 1.1, systems setting a minimum TLS protocol version above 1.1 (such as Debian 10 that sets MinProtocol=TLSv1.2) do not function with Connector/Python 8.0.17.

  As a workaround, the wheel (pip) packages function properly as they are built using glibc and bundle OpenSSL instead of YaSSL.

  Connector/Python 8.0.18 adds a `tls-versions` option to define the TLS version to use.

Changes in MySQL Connector/Python 8.0.17 (2019-07-22)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Prepared statement support was added to the C extension's (use_pure=False) implementation. (Bug #27364973, Bug #21670979, Bug #77780, WL #12085)
- Added `CONTRIBUTING.rst` and replaced `README.txt` with `README.rst`.

  Thanks to Daniël van Eeden for the `README.rst` patch. (Bug #20862622, Bug #76643, WL #12735)
- Added connection attribute support for the classic connector; new connection attributes can be passed in with the "conn_attrs" connection argument. Thanks to Daniël van Eeden for the patch. Example usage:

```
test_config = {'user': 'myuser', 'port':3306, 'host':'localhost'}
test_config['conn_attrs'] = {"foo": "bar", "_baz": "qux", "hello": "world"}
_ = connect(**test_config)
```

  Default connection attributes are set for both the pure and c-ext implementations, but these attributes are different due to limitations of the client library.

  For general information about connection attributes, see Performance Schema Connection Attribute Tables. (Bug #16562193, Bug #21072758, Bug #77003)
- Document fields containing arrays can now be indexed by setting array to true in an index *fields* definition. (WL #12227)

- Added support for the `OVERLAPS` and `NOT OVERLAPS` operators; which is equivalent to the SQL JSON_OVERLAPS() function.

  These binary operators are used with a general "expression operator expression" syntax; and the expressions return a JSON array or object. Example usage: *["A", "B", "C"] overlaps $.field* (WL #12737)

- Added support for the *utf8mb4_0900_bin* collation added in MySQL Server 8.0.17. (WL #13155)

Bugs Fixed

- Executing a *Collection.find()* without first fetching results would raise an AttributeError with an unclear message. (Bug #29327931)
- An error was generated when used with the combination of MySQL 5.7, Python 3, and having the C-extension enabled. (Bug #28568665)

Changes in MySQL Connector/Python 8.0.16 (2019-04-25)

Functionality Added or Changed

- Two informative text files were added: `INFO_BIN` contains information about the build environment used to produce the distribution, and `INFO_SRC` provides information about the product version and the source repository from which the distribution was produced. (Bug #29454706, WL #12297)
- Django 1.11 is now the minimum supported Django version. (WL #12863)
- For X DevAPI applications, Connector/Python now supports the ability to send connection attributes (key-value pairs that application programs can pass to the server at connect time). Connector/Python defines a default set of attributes, which can be disabled or enabled. In addition, applications can specify attributes to be passed in addition to the default attributes. The default behavior is to send the default attribute set.

  For X DevAPI applications, specify connection attributes as a `connection-attributes` parameter in a connection string, or setting `connection-attributes` as a dictionary inside the connection settings parameter under the `connection-attributes` key. Both the `mysqlx.get_session()` and `mysqlx.get_client()` methods can receive this information.

  The `connection-attributes` parameter value must be empty (the same as specifying `true`), a `Boolean` value (`true` or `false` to enable or disable the default attribute set), or a list or zero or more `key=value` specifiers separated by commas (to be sent in addition to the default attribute set). Within a list, a missing key value evaluates as an empty string. An example connection string:

  ```
  mysqlx://user:password@host:33060/schema?connection-attributes=[foo=bar,baz=qux,quux]
  ```

  Application-defined attribute names cannot begin with `_` because such names are reserved for internal attributes.

  If connection attributes are not specified in a valid way, an error occurs and the connection attempt fails.

  For general information about connection attributes, see Performance Schema Connection Attribute Tables. (WL #12488)

- Connector/Python now has improved support for resetting sessions in connection pools. Returning a session to the pool drops session-related objects such as temporary tables, session variables, and

transactions, but the connection remains open and authenticated so that reauthentication is not required when the session is reused. (WL #12489)

- Protobuf was updated to Protobuf 3.6.1. (WL #12864)
- For X DevAPI, performance for statements that are executed repeatedly (two or more times) is improved by using server-side prepared statements for the second and subsequent executions. This happens internally; applications need take no action and API behavior should be the same as previously. For statements that change, repreparation occurs as needed. Providing different data values or different `offset()` or `limit()` values does not count as a change. Instead, the new values are passed to a new invocation of the previously prepared statement. (WL #12225)

Bugs Fixed

- Added a "username" alias for the "user" connection argument. Thanks to Matthew Woods for the patch. (Bug #29324966, Bug #94248)
- Solaris 11 package files had the expected owner/group set as pb2user/common instead of root/bin. (Bug #29278489)
- CRUD operations would not allow referencing a renamed column (AS SomeLabel) from the fetched result. (Bug #29001628)
- Fixed a memory corruption issue that caused an unexpected halt when fetching fields. (Bug #28479054)
- Querying an empty LONG BLOB raised an IndexError exception. (Bug #27897881, Bug #90519)

Changes in MySQL Connector/Python 8.0.15 (2019-02-01)

Bugs Fixed

- The default value of the `allow_local_infile` option changed from True to False. (Bug #94046, Bug #29260128)

Changes in MySQL Connector/Python 8.0.14 (2019-01-21)

Functionality Added or Changed

- A default schema name can now be defined from the connection string. Statements executed using the session are executed against this default schema if no other schema is specified. An example connection string: `mysqlx://username:password@localhost:33160/myschema?ssl-mode=enabled` where *myschema* becomes the default schema name. An error is emitted if the schema does not exist.

  In addition, a new `get_default_schema()` method was added to retrieve this default schema's name. It returns the string "NONE" if a default schema name was not provided. (WL #12607)
- The count() method's error message was unclear when the table or collection was missing. (WL #12493)

Bugs Fixed

- The minimum Protobuf prerequisite version changed from 2.6.0 to 3.0.0. (Bug #29042229)

- On Windows, Python 2.7 binaries only contain the pure Python (and not the C extension) implementation yet the MSI installer still had the VC 2015 pre-requisite. (Bug #28992304)
- Improved error handling for classic protocol connections using the X protocol port. (Bug #28962337, WL #12240)
- Attempting to connect to a default schema with insufficient privileges would yield a "does not exist" error instead of "access denied." (Bug #28942938)
- Improved error handling for connection strings that contain invalid schema names, such as an empty string. (Bug #28933922)
- On Mac OS X, the requirements section was missing from the bundled README files. (Bug #28744076)
- The Debian package descriptions were improved. (Bug #28737774)
- The connector assumed that all values were expressions, which could lead to errors. The expr() method is now required to designate values as expressions. (Bug #28646344, Bug #92416)
- With Python 2.7 and use_pure=False, unicode characters in table object operations would cause an unexpected halt. (Bug #28280321)
- With "pure_python=False" set, mysql.connector would still use the Python implementation if the C extension was not available. (Bug #27794178, Bug #28201289)
- A new `ssl_verify_identity` connection option was added to verify the server name against the server certificate's common name (CN) and subject alternative names (SANs). Previously, only the C extension implementation performed this verification by default. This functionality exists to help prevent man-in-the-middle type attacks. (Bug #27434751)
- Connecting to MySQL Server 8.0.16 or greater using X Protocol could yield an error due to an unexpected notice from the server. (WL #12492)
- The classic connector's error message was improved when used against an X Protocol socket. A "Protocol mismatch" error is now expected. (WL #12492)

Changes in MySQL Connector/Python 8.0.13 (2018-10-22)

Functionality Added or Changed

- Added Python 3.7 support. (Bug #27081809, Bug #87818, WL #12239)
- To go with the existing `mysqlx.get_session(conn_str)` method, a new `mysqlx.get_client(conn_str, options)` method was added that creates a connection pool handler that provides a `get_session()` method to create and retrieve connections from the pool. The collection pooling options are:
  - `enabled`: enables or disables connection pooling; boolean and defaults to true.
  - `max_size`: maximum number of connections available in the pool; positive integer and defaults to 25.
  - `max_idle_time`: maximum number of milliseconds a connection can be idle in the queue before being closed; integer >= 0 and defaults to 0 (infinite).

- `queue_timeout`: maximum number of milliseconds a request will wait for a connection to become available; integer >= 0 and defaults to 0 (infinite).

  This is different than connect_timeout that's used for non-pooling. In a pooling scenario there are already connections in the pool, so queue_timeout controls how long to wait for a connection in the pool.

Example usage:

```
client = mysqlx.get_client(
    {
    'host': 'localhost',
    'port': 33060,
    'user': 'mike',
    'password': 'password'
    },
    { pooling: {
        enabled: true,
        max_idle_time: 5000,
        max_size: 25,
        queue_timeout: 20000
        }
    }
)
```

Closing a session attached to the pool makes the connection available in the pool for subsequent get+session() calls, while closing (destroying) the pool effectively closes all server connections. (WL #11897)

- Added a `connection-timeout` connection timeout query parameter. This defines the length of time (milliseconds) the client waits for a MySQL server to become available in the given network addresses. It was added to both the mysqlx.get_session() (non-pooling sessions) and mysqlx.get_client() (pooling sessions) interfaces. This option defaults to 10000 (10 seconds). The value 0 disables the timeout so the client will wait until the underlying socket (platform dependent) times out.

  Example usages:

```
mysqlx.get_session("root@localhost?connect-timeout=0");
mysqlx.get_session("root@[localhost:33060, 127.0.0.1:33060]?connect-timeout=5000");
```

  In a multi-host scenario, the `connect-timeout` value applies to each individual host. (WL #12226)

Bugs Fixed

- On Windows, the 32-bit MSI failed to install. The registry key path was updated to allow the CEXT prerequisite check to execute and pass. (Bug #28395599, Bug #28464866)
- Subsequent collection.add() method calls would leak memory if the C extension was enabled. (Bug #28278352)
- Missing bind() parameters could cause an unclear error message or unexpectedly halt. (Bug #28037275)
- The username and password fields are now quoted to allow special characters when making X DevAPI connections. (Bug #27528819, Bug #89614)

Changes in MySQL Connector/Python 8.0.12 (2018-07-27)

- Deprecation and Removal Notes
- Installation Notes

- Functionality Added or Changed
- Bugs Fixed

Deprecation and Removal Notes

- Deprecated the `Row.get_string()` method in favor of `__getitem__`. (Bug #26834200, Bug #87777)
- To increase compliance with the X DevAPI, these Connector/Python changes were made:
  - `DatabaseObject`: Deprecated: `am_i_real()` and `who_am_i()`. Added: `get_session()` and the `session` property.
  - `Collection.modify()`: Deprecated: `limit(x, y)`'s second parameter, `where(condition)`, and `change(CollectionField, ExprOrLiteral)`. Changed: the `modify(condition)` condition is now mandatory.
  - `Collection.find()`: Deprecated: `limit(x, y)`'s second parameter and where(condition). Added: `limit(x).offset(y)`.
  - `Collection.remove()`: Deprecated: `limit(x, y)`'s second parameter and `where(condition)`. Changed: the `modify(condition)` condition is now mandatory.
  - `Table.select()`: Deprecated: `limit(x, y)`'s second parameter and `sort()`. Added: `limit(x).offset(y)`.
  - `Table.delete()`: Deprecated: `limit(x, y)`'s second parameter and `sort()`. Removed: `delete(x)`'s parameter in favor of using `where()` instead. Added: `order_by()`.
  - `Table.update()`: Deprecated: `limit(x, y)`'s second parameter, and the `sort()` method. Added: `order_by()`.
  - `Session`: Added: `get_schemas()`.
  - `Result`: Deprecated: `get_document_id()` and `get_generated_insert_id()`. Moved: `get_affected_items_count()` to the `BaseResult` class.
  - `RowResult`: Added: `get_columns()`.
  - `SqlResult`: Added: `has_data()`.
  - `Column`: Renamed: `ColumnMetaData` to `Column`. Added properties: `schema_name`, `table_name`, `table_label`, `column_name`, `column_label`, `type`, `length`, `fractional_digits`, `collation_name`, `character_set_name`.

  (WL #11898, WL #13058)

Installation Notes

- Because the Microsoft Visual C++ 2017 Redistributable installer deletes the Microsoft Visual C++ 2015 Redistributable registry keys that identify its installation, standalone MySQL MSIs may fail to detect the Microsoft Visual C++ 2015 Redistributable if both it and the Microsoft Visual C++ 2017 Redistributable are installed. The solution is to repair the Microsoft Visual C++ 2017 Redistributable via the Windows Control Panel to recreate the registry keys needed for the runtime detection. Unlike the standalone MSIs, MySQL Installer for Windows contains a workaround for the detection problem. (Bug #28345281, Bug #91542)

Functionality Added or Changed

- Removed MySQL Fabric support. (WL #11932)
- An RPM package for installing ARM 64-bit (aarch64) binaries of Connector/Python on Oracle Linux 7 is now available in the MySQL Yum Repository and for direct download.

  Known Limitation for this ARM release: You must enable the Oracle Linux 7 Software Collections Repository (ol7_software_collections) to install this package, and must also adjust the libstdc++7 path. See Yum's Platform Specific Notes for additional details.

Bugs Fixed

- The default character set changed from 'utf8' (an alias to the deprecated 'utf8mb3' character set) to 'utf8mb4'. (Bug #28188883)
- Fixed datetime conversion compatibility between Django 2.0 and MySQL 8.0.

  A workaround was to use Connector/Python's pure Python implementation instead the C extension by setting "use_pure=True" in Django's database options. (Bug #27962293, Bug #90541)
- Connecting with a collation unknown to Connector/Python would yield an unknown character set error. It now properly references the unknown collation. (Bug #27277937)
- The `get_row()` and `get_rows()` behavior differed with the C (connections with CMySQLConnection) and pure Python (connections with MySQLConnection) implementations of the connector. The resolved differences are:
  - With the pure Python implementation, all data was returned as bytearrays; while the C implementation returned all data as Python types with CMySQLConnection (cext). Both now return Python types.
  - With the pure Python implementation, they returned a tuple with (row(s), eof), but with the C Extension they only returned the row(s). Now both implementations return the tuple form; (row(s), eof).
  - For queries returning a result, with pure Python the warning count was part of the returned eof. With the C extension, warning count was only available using the warning_count property of the connection object. Related, the warning_count property was not available in the pure Python implementation. Now, result includes the warning count for both implementations.
  - Fetching rows using pure Python would automatically set the unread_rows property to False. With the C extension, explicitly calling the free_result() method of the connection object was required. Related, free_result() was only available with the C extension. Now, both implementations set unread_rows to False.

  (Bug #22367904, Bug #27411275, Bug #27991948, Bug #27802700, Bug #28133321, Bug #27650437, Bug #79623, Bug #89305, Bug #90799, Bug #90585, Bug #90292, Bug #91107, WL #11951)

Changes in MySQL Connector/Python 8.0.11 (2018-04-19)

Functionality Added or Changed

- *X DevAPI:* Previously, when documents without an `_id` attribute were added to a collection, Connector/Python automatically generated _id for them. Now a MySQL 8 server generates the `_id` attribute unless

a document already contains one. The generated IDs resulting from a document-add operation can be obtained using the new `get_generated_ids()` method.

This capability requires a MySQL 8.0 server. Because MySQL 5.7 does not support document ID generation, the document-add operation returns an error if you do not define the _id's.

Incompatibility: The `get_generated_ids()` method replaces the now removed `get_document_ids()`. (Bug #27627366, WL #11448)

- Added *NOWAIT* and *SKIP_LOCKED* support to the ReadStatement.lock_shared() and ReadStatement.lock_exclusive() methods. Example usage: `lock_exclusive(mysqlx.LockContention.SKIP_LOCKED)`. (WL #11282)
- The C extension (cext) is now enabled by default, as the `use_pure` option changed from *True* to *False* by default.

  If the C extension is not available on the system then the Python implementation is used instead, and `use_pure` is set to *True*. (WL #11614)
- Added the X DevAPI SHA256_MEMORY authentication mechanism.

  Example mysqlx.get_session() usages: *?auth=SHA256_MEMORY* via a connection string, *"auth": mysqlx.Auth.SHA256_MEMORY* via a dictionary, or *auth=mysqlx.Auth.SHA256_MEMORY* via method parameters. (WL #11668)

Bugs Fixed

- Warnings are now stored as a list of dictionaries instead of a list of tuples. In other words, *get_warnings()* returns the likes of *[{"level": _level_, "code": _code_, "msg": _msg_}]* instead of *[(_level_, _code_, _msg_)]*. (Bug #27639119)
- The mapped MySQL Server error codes were synced with MySQL Server 8.0.11. (Bug #27634885)
- Removed *upsert* functionality from *InsertStatement* as it can only be used by collections, so *upsert* remains available to *AddStatement*. (Bug #27589450)
- *MySQLConverter.escape()* functionality was added to *create_schema()*'s count mechanism. (Bug #27528842)
- When using prepared statements, string columns were returned as bytearrays instead of strings. The returned value is now a string decoded using the connection's `charset` (defaults to 'utf8'), or as a bytearray if this conversion fails. (Bug #27364914)
- The result from JSON_TYPE() was returned as a bytearray instead of a string. The returned value is now a string decoded using the connection's `charset` (defaults to 'utf8'), or as a bytearray if this conversion fails. (Bug #24948205, Bug #83516)
- JSON integer values were cast to bytes in Python instead of integers. (Bug #24948186, Bug #83513)

Changes in MySQL Connector/Python 8.0.7 - 8.0.10 (Skipped version numbers)

There are no release notes for these skipped version numbers.

Changes in MySQL Connector/Python 8.0.6 (2018-02-01, Release Candidate)

- Functionality Added or Changed

- Bugs Fixed

Functionality Added or Changed

- A new `bdist_wheel` distutils command was added to build a Connector/Python wheel package.

  A new `--static` option was added that enables static linking for the C extension variant. (WL #11041)

- *X DevAPI:* In the process of refining the definition of the X DevAPI to cover the most relevant usage scenarios, the following API components have been removed from the X DevAPI implementation for Connector/Python:

  - **API components that support session configurations**.

    The `mysqlx.config` namespace and all members of the namespace.

  - The `create_table`, `drop_table`, `create_view`, `drop_view`, and `alter_view` methods from the `Schema` class.

  (WL #11303, WL #11372)

- A *Pylint* test was added for the *mysqlx* module. (WL #11149)

- A new `Modify.patch()` method was added to the X DevAPI as a way to change several document attributes in one operation; otherwise known as a JSON Merge Patch via RFC 7386. (WL #11142)

- The `create_index()` method was added to the Collection API. (WL #11278)

- The transaction API was extended to allow setting savepoints. The following methods have been added to the *Session* object:

  - `set_savepoint([name])`: executes the SAVEPOINT name SQL statement to generate a savepoint. If a name is not provided (or None), one is generated.

    The SAVEPOINT statement sets a named transaction savepoint with a name of identifier. If the current transaction has a savepoint with the same name, the old savepoint is deleted and a new one is set.

  - `release_savepoint(name)`: executes the RELEASE name SQL statement to release a savepoint.

    The RELEASE SAVEPOINT statement removes the named savepoint from the set of savepoints of the current transaction. No commit or rollback occurs. It returns an error if the savepoint does not exist.

  - `rollback_to(name)`: executes the ROLLBACK TO name SQL statement to rollback a savepoint.

    The ROLLBACK TO identifier command reverts the state of the transaction back to what was when executed the command SAVEPOINT identifier.

  Names passed to these functions are checked to make sure that the name is not null or an empty string. Names such as '', "", ``, and so on, are not allowed even though they are allowed by the server. For more information, see SAVEPOINT, ROLLBACK TO SAVEPOINT, and RELEASE SAVEPOINT Statements. (WL #11281)

Bugs Fixed

- On Enterprise Linux 7, SSL connections could fail due to the Python 2.7.9 or higher requirement. Since EL7 backported the SSL module from Python 3 (PEP466) into its default Python 2.7.5, SSL connections are now enabled on EL7. (Bug #27368032)
- MySQL Server 8.0 utf8mb4 collations were missing from Connector/Python. (Bug #27277964)
- The LICENSE and README files were missing from the C extension (“cext”) builds. (Bug #26912787)
- On Linux, commercial packages included source (.py) files in the package instead of only .pyc/.pyo files. (Bug #26821756)
- Python 3.6 is now officially supported and tested. (WL #11079)

Changes in MySQL Connector/Python 8.0.5 (2017-09-28, Development Milestone)

Packaging Notes

- MySQL Connector/Python packages are now available in two formats: Pure Python packages that contain only Python files, and packages that contain the Python files plus the C Extension and C Protobuf extension. Exception platforms are Solaris, macOS, and Windows, for which packages containing the Python files and C extensions are available but not pure Python packages. (Bug #26648417)

Functionality Added or Changed

- MySQL Connector/Python now supports connections to MySQL accounts that use the `caching_sha2_password` authentication plugin (see Caching SHA-2 Pluggable Authentication). This requires MySQL server version 8.0.3 or higher. It also requires use of a secure connection because Connector/Python does not support RSA encryption for password exchange. (WL #11073)
- MySQL Connector/Python now supports an `auth` connection option to specify the authentication methanism. Permitted values are `plain`, `mysql41`, and `external`. The option name and value are not case sensitive.

  If the authentication mechanism is not specified, it defaults to `plain` for secure (TLS) or Unix socket connections, or `mysql41` for insecure connections. (WL #10771)
- MySQL Connector/Python now supports a pure Python implementation of Protobuf. Consequently, the Protobuf C extension has become optional. Connector/Python will use the Python implementation if the C extension is not available. The Protobuf Python package is required if it is desired not to use the C extension.

  The version requirements are Protobuf C++ 2.6.0 or higher, Protobuf Python 3.0.0 or higher. (WL #10899)
- A `mysqlx.sessions` variable is now exposed to scripts that can be used for session-related tasks such as saving or loading session configuration information. (WL #10053)
- These methods have been added for `Collection`: `add_or_replace_one()`, `get_one()`, `replace_one()`, and `remove_one()`. (WL #10975)

- These methods have been added for `FindStatement` and `SelectStatement`, to enable shared and exclusive locks to be acquired: `lock_shared()` and `lock_exclusive()`. (WL #10974)

- There is support for new forms of comparisons that use the `IN` operator:

```
item IN list
item IN document path
dict IN dict
```

  The left-hand-side value must be castable to the `JSON` type. (WL #10973)

Changes in MySQL Connector/Python 8.0.4 (2017-07-17, Development Milestone)

MySQL Connectors and other MySQL client tools and applications now synchronize the first digit of their version number with the (latest) MySQL server version they support. For example, MySQL Connector/Python 8.0.12 would be designed to support all features of MySQL server version 8 (or earlier). This change makes it easy and intuitive to decide which client version to use for which server version.

Connector/Python 8.0.4 is the first release to use the new numbering. It is the successor to Connector/Python 2.2.3.

Character Set Support

- Connector/Python now supports MySQL servers configured to use `utf8mb4` as the default character set. (WL #10659)

Functionality Added or Changed

- To avoid unintentional changes to all items in a collection, the `Collection.modify()` and `Collection.remove()` methods now require a nonempty selection expression as argument. To intentionally apply an operation to an entire collection, pass a condition that always evaluates to true, such as `True`. A similar change was made to `Table.update()` and `Table.delete()`. (Bug #25991574, WL #10754)

- For MSI and Solaris packages, the pure Python Protobuf support implementation was replaced by a C ++ extension. This enables Connector/Python to support Python 2 and 3 as well Protobuf 2 and 3. (WL #10198)

- The `NodeSession` class has been renamed to `Session`, and the `XSession` class has been removed. (WL #10658)

- Connections created using `Session` objects now are encrypted by default. Also, the `ssl-enabled` connection option has been replaced by `ssl-mode`. Permitted `ssl-mode` values are `disabled`, `required` (the default), `verify_ca` and `verify_identity`. (WL #10770)

- The format of document ID values generated when adding documents to a collection has changed. It is still a string of 32 hexadecimal digits based on UUID, but the order of digits was changed to match the requirement of a stable ID prefix. (WL #10004)

Bugs Fixed

- The C Extension was not installed by some Connector/Python installers, such as Solaris `.pkg` and macOS `.dmg` installer packages. (Bug #24422244)
- `Collection.drop_index("name")` incorrectly returned an instance of `DropCollectionIndexStatement`. (WL #10772)

Index

R

W