MySQL Shell 8.0 Release Notes

Abstract

This document contains release notes for the changes in each release of MySQL Shell 8.0.

For additional MySQL Shell documentation, see http://dev.mysql.com/.

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 (http://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 either the MySQL Forums or MySQL Mailing Lists, where you can discuss your issues with other MySQL users.

For additional documentation on MySQL products, including translations of the documentation into other languages, and downloadable versions in variety of formats, including HTML and PDF formats, see the MySQL Documentation Library.

Document generated on: 2018-02-21 (revision: 14082)

Table of Contents

Preface and Legal Notices

This document contains release notes for the changes in each release of MySQL Shell 8.0.

Legal Notices

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

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 Shell (part of MySQL 8.0) 8.0.6 (Not yet released)

Version 8.0.6 has no changelog entries, or they have not been published because the product version has not been released.

Changes in MySQL Shell (part of MySQL 8.0) 8.0.5 (Not yet released)

Version 8.0.5 has no changelog entries, or they have not been published because the product version has not been released.

Changes in MySQL Shell (part of MySQL 8.0) 8.0.4 (2018-01-25, Release Candidate)

Functionality Added or Changed

- When you create clusters or add instances you can now override the default group name, local addresses, and group seeds. This makes it easier to customize your clusters. The following options were added to the `dba.createCluster()` and `cluster.addInstance()` commands:

  - use `groupName` with `dba.createCluster()` to set the name of the cluster
  - use `localAddress` to set the address which an instance provides to communicate with other instances
  - use `groupSeeds` to set the instances used as seeds when instances join the cluster

  For more information, see Customizing InnoDB clusters. (Bug #26485254, Bug #26838005)

- Connections to an InnoDB cluster have been simplified. Now, when you issue `dba.getCluster()` and the active Shell session is not a connection to the primary, the cluster is queried for the primary information and a connection to the primary is automatically opened. This ensures that configuration changes can be written to the metadata. As part of this improvement you can now configure the type of MySQL Shell connection to an InnoDB cluster with the following options which have been added:

  - `--redirect-primary`
  - `--redirect-secondary`
  - `--cluster`

  Additionally the following changes were made:

  - The `dba.resetSession()` method has been removed.
  - A new `disconnect()` method has been added to the `cluster` object, which closes all internal sessions opened by the cluster. InnoDB cluster operations on a disconnected cluster object result in an error.
  - The output of the `Cluster.status()` method now includes the `groupInformationSourceMember` field, which shows the URI of the internal connection used by the cluster object to obtain information about the cluster.

References: See also: Bug #25091586.

- MySQL Shell now supports autocompletion of text preceeding the cursor by pressing the Tab key. Autocompletion is available for:
  - Built-in MySQL Shell commands, for example typing **\con** followed by the Tab key completes to `\connect`.
  - SQL, JavaScript and Python language keywords depending on the current MySQL Shell mode.
  - Table names, column names, and active schema names in SQL mode, based on the current default schema.

  Autocompletion can be configured using the following command options:
  - `--name-cache`
  - `--no-name-cache`
- A new `patch()` operation has been added to the `modify()` function which enables you to modify a document by merging in a set of changes. For more information see `JSON_MERGE_PATCH()`.
- You can now use Unix sockets for X Protocol connections. Socket file paths in URI type strings should be either percent encoded, such as `root@/home%2Fuser%2Fmysql-sandboxes%2F3310%2Fsandboxdata%2Fmysqlx.sock`, or surrounded by parenthesis such as `root@(/home/user/mysql-sandboxes/3310/sandboxdata/mysqlx.sock)`. The `--socket` option cannot be combined with the `--port` option.
- MySQL Shell performs automatic `_id` generation on collection add operations when no `_id` is specified on the documents being added. The autogenerated _id is created using the UUID() function. Now, the order of the tokens used has changed from `aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee`, where `eeeeeeeeeeee` is the MAC Address, to `eeeeeeeeeeee-dddd-cccc-bbbb-aaaaaaaa`.
- Sessions created by X DevAPI using either `mysql.getClassicSession(connection_data)` or `mysqlx.getSession(connection_data)` now use `ssl-mode=REQUIRED` as the default if no `ssl-mode` is provided, and neither `ssl-ca` nor `ssl-capath` is provided. If no `ssl-mode` is provided and any of `ssl-ca` or `ssl`-capath is provided, all Sessions created by MySQL Shell now default to `ssl-mode=VERIFY_CA`.
- In addition to the existing CRUD commands which work on documents in a collection by matching a filter, the following operations have been added to enable you to work with single documents:
  - `Collection.replaceOne(string id, Document doc)` updates (or replaces) the document identified by `id` with the provided one, if it exists.
  - `Collection.addOrReplaceOne(string id, Document doc)` add the given document. If the `id` or any other field that has a unique index on it already exists in the collection, the operations updates the matching document instead.
  - `Document Collection.getOne(string id)` returns the document with the given `id`. This is a shortcut for `Collection.find("_id = :id").bind("id", id).execute().fetchOne()`.
  - `Result Collection.removeOne(string id)` removes the document with the given `id`. This is a shortcut for `Collection.remove("_id = :id").bind("id", id).execute().`

  Using these operations you can reference documents by ID, making operations on single documents simpler by following a load, modify and save pattern such as:

```
doc = collection.getOne(id);
doc["address"] = "123 Long Street";
collection.replaceOne(id, doc);
```

- You can now use savepoints with MySQL Shell sessions. The following methods have been added to the Session object:

  - Use `setSavepoint()` to generate a savepoint. The server executes the `SAVEPOINT` SQL statement and returns the generated savepoint name.
  - Use `setSavepoint(name)` to specify the name used by the `SAVEPOINT` SQL statement.
  - Use `releaseSavepoint(name)` to execute the `RELEASE` SQL statement.
  - Use `rollbackTo(name)` to execute the `ROLLBACK TO name` SQL statement.

  Any 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 Syntax.

- X DevAPI now supports row-locking for CRUD operations on tables and collections. Use the `lockShared()` method for write locks and the `lockExclusive()` method for read locks, which have been added to `find()` and `select()`. Either method can be called any number of times, including none, in any combination. But the locking type to be used is always the last one called. Once you call a lock method you can only then execute the operation, calling `execute()` or `bind()`. For more information, see InnoDB Locking.

- The X DevAPI drop operations have been improved. Now, the `drop()` methods are at the same level as the `create()` methods and they all return nothing. There is now no need to use `execute()` after the drop method. If a drop method is called on an object which no longer exists in the database there is now no error.

  This modifies Session objects so that:

  - `dropSchema()` returns Nothing
  - `dropTable(String schema, String name)` is removed
  - `dropCollection(String schema, String name)` is removed
  - `dropView(String schema, String name)` is removed

  This modifies Schema objects so that:

  - Nothing `dropCollection(String name)` is added

  This modifies Collection objects so that:

  - `dropIndex(String name)` is a direct operation, meaning `.execute()` is no longer needed
  - `dropIndex(String name)` returns Nothing

- The `mysql` module, used with classic sessions for SQL connections to instances, has been streamlined. This has resulted in the following changes:

- The new `getSession()` function has been added to the `mysql` module to create a classic session. This function works in the same way as the existing `mysql.getClassicSession()` function.
- The `runSql()` method of `ClassicSession` has been extended to take a list of arguments to replace with placeholders in the query. For example, now you can issue:

  ```
  session.runSql("select * from tbl where name = ? and age > ?", ["Joe", "20"])
  ```

  Additionally a `query()` function has been added to `ClassicSession` as an alias to `runSql()`, making it consistent with `Session.query()`.
- The following functions have been removed from `ClassicSession`:
  - `createSchema()`
  - `getSchema()`
  - `getDefaultSchema()`
  - `getCurrentSchema()`
  - `setCurrentSchema()`
  - `ClassicSchema()`
- The `ClassicTable` object has been removed.
- The following `Table` and View DDL functions have been removed from `Schema` objects:
  - `dropTable()`
  - `dropView()`
- The behavior of the global `db` variable has been modified. When a global connection is created and the connection data includes a default schema, the global `db` variable is set to the corresponding `Schema` object. Now, this is not done for connections through the MySQL protocol, such as those created by `ClassicSession`.

- Use `util.checkForServerUpgrade([uri])` to check if a server can be upgraded from the version it is currently running to the next major series release. For example, you can verify that a server instance running MySQL 5.7 satisfies the upgrade prerequisites for MySQL 8.0, see Verifying Upgrade Prerequisites for Your MySQL 5.7 Installation. To verify a server instance at a URI type string such as `user@example.com:3306` issue:

  ```
  util.checkForServerUpgrade(['user@example.com:3306'])
  ```

  MySQL Shell provides a report on anything in the table space which would cause a problem when upgrading the instance to MySQL 8.0.
- MySQL Shell builds against MySQL server version 8.0.4 and OpenSSL.
- MySQL 8.0.4 uses the `caching_sha2_password` authentication plugin by default. This means any new accounts created in MySQL use this new plugin. This new authentication method is completely transparent if the connection is made with SSL enabled, which is the default. Connecting with SSL enabled is also the preferred mode of connection. If SSL is explicitly disabled, the following limitations apply:

- If the caching_sha2_password plugin reports an error such as `Authentication requires a secure connection`, it is not possible to connect to the account with MySQL Shell, until either a SSL connection is made or the mysql client is used to clear the error.
- X Protocol sessions require SSL and do not authenticate with SSL disabled.

Bugs Fixed

- Attempting a server connection without specifying the port or socket, when using an expired or temporary password, failed with an error requiring a password reset. The default connection data is now set at an appropriate point in connection processing, so the connection succeeds. (Bug #27266648)
- MySQL Shell required the option name `ssl-ciphers` instead of the standard MySQL option name `--ssl-cipher` to specify the list of permitted ciphers for connection encryption. The standard option name `--ssl-cipher` is now required in MySQL Shell. (Bug #27185275)
- The `--show-warnings` option was not working in MySQL Shell. (Bug #27036716)
- The command line options `--classic`, `--node`, `--sqln`, and `--ssl` were stated as deprecated, but actually had been removed. MySQL Shell now handles the options again, but prints a warning message when they are used. The deprecated options are processed as their replacement options, as follows:
  - `--classic` is processed as `--mysql`
  - `--node` is processed as `--mysqlx`
  - `--sqln` is processed as `--sqlx`
  - `--ssl` is processed as `--ssl-mode`

  (Bug #27012385)
- The output of `\status` when using a Unix socket for connections was showing a relative path. Now the absolute path of the socket is shown. (Bug #26983193)
- Account validation did not work correctly unless the session account existed. Now, validation is done using the account that was authenticated by the server. (Bug #26979375)
- The AdminAPI in MySQL Shell for working with InnoDB cluster only supports TCP connections to server instances. The AdminAPI now checks that a TCP connection is in use before starting an operation that requires database access, instead of attempting the operation with another connection type and not succeeding. (Bug #26970629)
- If you issued `STOP GROUP REPLICATION` on an instance that belonged to a cluster, attempting to rejoin the instance to the cluster failed because the wrong Group Replication seeds were being used. Now, `Cluster.rejoinInstance()` correctly sets `group_replication_group_seeds` based on the `group_replication_local_address` of all currently active instances in the cluster. (Bug #26861636)
- Sometimes the `dba.addInstance()` command failed with an error indicating that the server was in `RECOVERING` state despite being `ONLINE`. The fix ensures the correct instance state is returned. (Bug #26834542)
- If the user running MySQL Shell did not have write permissions to the option file configured by AdminAPI, no error was displayed. (Bug #26830224)

- With the addition of WL#10470, the default value of `server_id` has changed to 1. As the server ID has to be unique for each instance, this caused issues with AdminAPI. Now, server instances with `server_id=1` are correctly identified as incorrectly configured for InnoDB cluster use. (Bug #26818744)

- AdminAPI now supports Python 2.6 in addition to Python 2.7, removing the need to manually install on Oracle Linux 6. (Bug #26809748)

- After removing an instance from a cluster using `Cluster.removeInstance()`, the instance silently rejoined the Group Replication group after it restarted. This happened because `group_replication_start_on_boot` was set to `ON` by default. Now, for instances running MySQL version 8.0.4 and later, the fix sets `group_replication_start_on_boot` to `OFF` in the option file. For instances running a MySQL version earlier than 8.0.4, a warning is issued to tell you to manually edit `group_replication_start_on_boot` in the instance's option file to avoid the issue. (Bug #26796118)

- Using AdminAPI commands on Windows that required SSL resulted in an error due to the Python version being used. (Bug #26636911)

- Creating an InnoDB cluster from an existing Group Replication deployment, by using the `adoptFromGR` option with the `dba.createCluster()` command, would fail with an error stating that the instance was already part of a replication group. The issue was only present in the MySQL Shell default wizard mode. The fix ensures that the interactive layer of the `dba.createCluster()` command allows the use of the `adoptFromGR` option. (Bug #26485316)

- Support for microseconds has been added to the `mysqlx.dateValue()` function and the `Date` object. (Bug #26429495)

- The warnings generated when creating and adding sandbox instances have been improved. (Bug #26393614)

- When working with instances that had `require_secure_transport=ON`, AdminAPI commands that required a connection to the instance failed. (Bug #26248116)

- MySQL Shell now automatically pre-loads the built-in `mysql` and `mysqlx` API modules when it is invoked in batch mode, as well as when it is used in interactive mode. (Bug #26174373)

- The user configuration path for MySQL Shell defaults to `%AppData%\MySQL\mysqlsh\` on Windows, and `~/.mysqlsh/` on Unix. This directory is used for the MySQL Shell history file (`history`), log file (`mysqlsh.log`), and theme file (`prompt.json`). It is also the final location that MySQL Shell searches for startup scripts (`mysqlshrc.js` or `mysqlshrc.py`).

  You can now override the user configuration path by specifying an alternative path using the `MYSQLSH_USER_CONFIG_HOME` environment variable. A directory specified by that environment variable is used by MySQL Shell for the user configuration data in place of `%AppData%\MySQL\mysqlsh\` on Windows, and `~/.mysqlsh/` on Unix. (Bug #26025157)

- The `Cluster.dissolve()` command was trying to stop Group Replication on all of the instances registered in the metadata which lead to connection errors if any of those instances were not contactable, in other words with the state `(MISSING)`. The fix ensures that only instances which can be contacted, in other words with the state `ONLINE`, are stopped. (Bug #26001653)

- When adding instances to an InnoDB Cluster using the appropriate AdminAPI operations, checks are performed to verify the compatibility of any existing tables. If incompatible tables (for example using `MyISAM`) are detected then an error is issued. However the error message was referring to an option not available for the AdminAPI operations: `--allow-non-compatible-tables`. (Bug #25966731)

- For file processing, MySQL Shell now expands a leading tilde in a file path to the appropriate home directory. MySQL Shell identifies the home directory using a relevant environment variable, or looks it up for the logged-in user. (Bug #25676405)
- MySQL Shell now provides a `program_name` connection attribute to the server at connect time, with the value `mysqlsh`. Connection attributes are displayed in the Performance Schema connection attribute tables. (Bug #24735491, Bug #82771)
- Running `help()` in MySQL Shell in Python mode caused an AttributeError. (Bug #24554329, Bug #82767)
- The `cluster.rejoinInstance()` command attempted to rejoin an instance even if was already part of the cluster. Now, only instances in the `MISSING` state are accepted by `cluster.rejoinInstance()`. Attempting to rejoin an instance in any other state fails with an error. (Bug #87873, Bug #26870329)
- On Unix, if Python 3 was installed AdminAPI commands failed. (Bug #87731, Bug #26785584)
- When using the `dba.checkInstanceConfiguration()` and `dba.configurelocalinstance()` commands, the account being used was not being checked if it had enough privileges to actually execute the command. The fix ensures that account has the required privileges before proceeding. This also required a change of the privileges given to `clusterAdmin` users. (Bug #87300, Bug #26609909)
- Arrays and Objects now accept the `IN` operator. For example:

```
collection.find("'Fred' IN username")
```

Changes in MySQL Shell (part of MySQL 8.0) 8.0.3 (2017-09-29, Development Milestone)

MySQL Shell now synchronizes the first digit of its version number with the (highest) MySQL server version it supports. This change makes it easy and intuitive to decide which client version to use for which server version. MySQL Shell now uses the same version number as MySQL Server.

MySQL Shell 8.0.3 is the first release to use the new numbering. It is the successor to MySQL Shell 8.0.0.

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- With the addition of and WL#10611 and WL#10960 , it was not possible to add or rejoin instances that belonged to a cluster (or a replication group) because `super_read_only=ON` was being set by Group Replication when stopping. To ensure that AdminAPI supports instances running MySQL 8.0.2 and later, the following functions have been modified:
  - `dba.configureLocalInstance()`
  - `dba.createCluster()`
  - `dba.rebootClusterFromCompleteOutage()`
  - `dba.dropMetadataSchema()`

  Now, if any of these functions is issued against an instance which has `super_read_only=ON`, in interactive mode you are given the choice to set `super_read_only=OFF`. To force the function to set `super_read_only=OFF` in a script, pass the `clearReadOnly` option set to `true`. For example

`dba.configureLocalInstance({clearReadOnly: true}).` For more information see Super Read-only and Instances. (Bug #26422638)

- MySQL Shell now handles user interrupts, such as SIGINT, correctly. For example on Linux pressing Control-C when MySQL Shell is not executing anything exits the application. In SQL mode, interruption sends a `KILL QUERY` statement to the active MySQL Shell session from a new temporary session, resulting in the server interrupting the query and returning an error (or in an early return with no error in some cases, like the `sleep()` function). In JavaScript or Python scripting modes, how interruption behaves depends on the specific function being executed. If what is being executed is language code (such as a while loop and other normal script code), an exception is generated in the active language, which causes the code to stop executing. The exception may be caught by the script, but if not, the execution control returns to MySQL Shell. (Bug #24757361)

- MySQL Shell now includes a history function that stores the code which you issue. The history can be saved, searched, and filtered. A new mechanism to customize the MySQL Shell prompt has been added. Information such as the current mode (SQL, JavaScript, or Python), session information (host, uri, port and so on), the current active schema and others can be included in the prompt through variables. The customization information is self-contained in JSON theme files, which can be shared between users. MySQL Shell now supports unicode if the terminal used to run MySQL Shell supports it. Similarly if the terminal supports color, MySQL Shell can be configured to use colors in the theme.

- The connection options passed to MySQL Shell, such as `sslMode` and so on, have been changed to use dashes and no longer be case sensitive. The options are now:

  - `sslMode` is now `ssl-mode`
  - `sslCa` is now `ssl-ca`
  - `sslCaPath` is now `ssl-capath`
  - `sslCert` is now `ssl-cert`
  - `sslKey` is now `ssl-key`
  - `sslCrl` is now `ssl-crl`
  - `sslCrlPath` is now `ssl-crlpath`
  - `sslCiphers` is now `ssl-ciphers`
  - `sslTlsVersion` is now `tls-version`
  - `authMethod` is now `auth-method`

- The types of session available have been simplified. `XSession` and `NodeSession` have been consolidated into `Session`. This has caused the following changes:

  - The following command options have been deprecated: `--node`, `--sqln`, `--classic`
  - The following command options have been introduced to replace the deprecated ones: `-ma`, `--mysqlx` (`-mx`), `--mysql` (`-mc`), `--sqlx`
  - The `\connect` MySQL Shell command no longer supports the arguments `-c`, and `-n`. Now the `\connect` command supports the argument `--mysqlx`(`-mx`) for creating X Protocol connections, and `--mysql`(`-mc`) for creating MySQL protocol connections.

- The interpretation of the `document_path` field in operations such as `modify()` has been changed. Now, when the `document_path` is not set, operations apply to the whole document. All operations always preserve a document's `_id` field.

Bugs Fixed

- In MySQL Shell, the `Schema.getCollectionAsTable()` function and the `select()` method could not be used in the same Python statement. (Bug #26552804)
- When using the `clusterAdmin` option, the created account did not have all of the correct privileges. (Bug #26523629)
- MySQL Shell returned some elements of DATE and DATETIME values incorrectly, including month values and fractional seconds. (Bug #26428636)
- The month was incorrectly incremented on insertion of a timestamp in a table using MySQL Shell. (Bug #26423177)
- For columns with the `ZEROFILL` attribute, `NULL` was also returned padded with zeroes. (Bug #26406214)
- The output of the MySQL Shell `\status` command was enhanced with additional information. (Bug #26403909)
- The MySQL Shell help for the `\connect` command indicated that a connection name could be used instead of a URI string, which was incorrect. (Bug #26392676)
- When using the `multiMaster` option with `dba.createCluster()`, the warning displayed in interactive mode was not being logged. (Bug #26385634)
- When making cluster topology or membership changes, AdminAPI was not taking into consideration the value of `group_replication_group_name`, which could lead to incorrect, non-deterministic results in scenarios such as a split brain. Now, the following commands validate the InnoDB cluster Metadata and the corresponding instance's `group_replication_group_name` value:

  - `dba.getCluster()`
  - `Cluster.rejoinInstance()`
  - `Cluster.forceQuorumUsingPartitionOf()`

  If the values of `group_replication_group_name` do not match, the commands abort with an error.

  `dba.rebootClusterFromCompleteOutage()` was also updated to ensure that the `group_replication_group_name` variable has not been changed before rejoining the instance. (Bug #26159339)
- AdminAPI now always uses the active user value for the current `mysqlsh` session, whether the value was explicitly specified by the user or is the result of an implicit default used by `mysqlsh`. (Bug #26132527)
- The checks performed by the AdminAPI upon issuing `dba.rebootClusterFromCompleteOutage()` were more strict than those required by Group Replication. Now, the AdminAPI considers tables with a Primary Key Equivalent (such as a Non Null Unique Key) as compatible, matching the current requirement for Group Replication. (Bug #25974689)
- The MySQL Shell command `\use` did not attempt to reconnect if the connection to the global session was lost. (Bug #25974014, Bug #86118)

- The short form `-?` can now be used as an alias for the `--help` command-line option in MySQL Shell. (Bug #25813228)
- The MySQL Shell command history displayed the commands that were used to automatically import the `mysql` and `mysqlx` API modules when MySQL Shell started. (Bug #25739185)
- The randomly generated passwords used by internal users were not compatible with instances running the Password Validation plugin. (Bug #25714751)
- The MySQL Shell command history displayed the contents of scripts that were run using the `\source` MySQL Shell command. (Bug #25676495)
- It is no longer possible to use the `adoptFromGr` option with the `multiMaster` option. When adopting an existing group to an InnoDB cluster, the group is adopted based on whether it is running as multi-primary or single-primary. Therefore there is no use for the `multiMaster` option when adapting a group. (Bug #25664700)
- Issuing `configureLocalInstance()` when using a URI that contained a user without the correct privileges resulted in an incorrect new user being created. Now, if the user in `configureLocalInstance()` URI does not have enough privileges to grant all the necessary privileges for the new user chosen during the interactive wizard configuration the user is not created. (Bug #25614855)
- The `mysqlx.getNodeSession()` function in MySQL Shell now returns an error if an unrecognized connection option is provided. (Bug #25552033)
- Issuing `Cluster.rescan()` resulted in non-determinstic behavior which could produce incorrect JSON output, showing an instance that was already part of the cluster as belonging to the `newlyDiscoveredInstances[]` list and to the `unavailableInstances`[] list. This also resulted in MySQL Shell prompting to add or remove the instance from the cluster. (Bug #25534693)
- AdminAPI functions now accept the standard connection parameters as used by `shell.connect`. New validations have been added for when `require_secure_transport` is `ON`, now it is not possible to create a cluster with `memberSslMode:DISABLED` or to add an instance with `require_secure_transport=ON` to a cluster where `memberSslMode:DISABLED`. (Bug #25532298)
- The parsing of account names, for example when passing the `clusterAdmin` option to `dba.configureLocalInstance()` has been improved. (Bug #25528695)
- The file permissions of option files created by AdminAPI did not match those of options files created by MySQL install. (Bug #25526248)
- Issuing `configureLocalInstance()` twice could fail. (Bug #25519190)
- When passing the `rejoinInstances[]` option to `dba.rebootClusterFromCompleteOutage()`, if no `rejoinInstances[]` option was specified then members were being incorrectly handled during the rebuild. Now, instances that are eligible to be added to the `rejoinInstances[]` list but that are specified in the `removeInstances[]` list are skipped by the interactive wizard that tries to automatically build a `rejoinInstances[]` list if one was not provided. This fix also ensures that both interactive and non-interactive use of MySQL Shell correctly verify the `rejoinInstances[]` list does not contain a unreachable instance. (Bug #25516390)
- The error messages issued when the SSL mode used by the cluster and the one specified when issuing `addInstance()` command do not match have been improved. (Bug #25495056)

- The `--ssl` option has been deprecated, use the `--ssl-mode` option. Now, if you use the `--ssl` option a deprecation warning is generated and the `--ssl-mode` option is set to either `DISABLED` or `REQUIRED` based on the value used with the `--ssl` option. (Bug #25403945)
- When creating a sandbox instance using the `dba.deploySandboxInstance()` function in MySQL Shell, pressing Ctrl + C at the prompt for a MySQL root password for the instance did not cancel the deployment. (Bug #25316811)
- MySQL Shell did not exit gracefully when the user did not have a valid and accessible home directory. (Bug #25298480)
- MySQL Shell created a logger but did not deallocate it on exiting the shell. (Bug #25238576)
- Issuing `removeInstance`() on the last member of a cluster, and particularly the seed member, was resulting in a cluster that could not be dissolved. Now, issuing `removeInstance()` on the last member of a cluster results in an error, and you must use `dissolve()` on that instance to ensure the cluster is correctly dissolved. (Bug #25226130)
- The output of `cluster.status()` now includes the `ssl` parameter, which shows whether secure connections are required by the cluster or disabled. (Bug #25226117)
- MySQL Shell could hang when Ctrl + C was used to exit the shell. (Bug #25180850, Bug #84022)
- Attempting to create a multi-primary cluster in interactive mode failed unless you passed in the `{force: true}` option. Now when you confirm that you understand the impact of using multi-primary mode the command correctly creates a multi-primary cluster. (Bug #25034951)
- The `removeInstance()` was not working on stopped instances and it was not possible to remove an unavailable instance from the cluster. The fix adds a new option `force` to the removeInstance() command to enable you to remove instances from the metadata that are permanently not available, avoiding obsolete data from being kept in the metadata of the cluster. In addition the error message provided when not using the force option has been improved and the online help for the `removeInstance()` was also updated accordingly. (Bug #24916064)
- Unsigned data could be incorrectly read from the database. (Bug #24912358)
- The parsing of Unix sockets provided as part of a URI has been improved. (Bug #24905066)
- The error messages generated by issuing `dba.deployLocalInstance()` against an unsuitable or incompatible instance have been improved. (Bug #24598272)
- The `dba.createCluster()`, `dba.getCluster()`, and `dba.rebootClusterFromCompleteOutage()` functions have been updated to validate the cluster name, using the following rules:
  - Name must start with a letter or the _ character
  - Name can only contain alphanumeric characters and the _ character
  - Cannot be longer than 40 characters
  - Cannot be empty

  The `Cluster.addInstance()` function has been updated to validate the label used on an instance in the cluster, using the following rules:
  - Label can only contain alphanumerics or the _ character

- Cannot be longer than 256 characters
- Cannot be empty

(Bug #24565242)

- MySQL Shell now accepts Unicode characters as input. (Bug #23151666, Bug #81176)

Changes in MySQL Shell (part of MySQL 8.0) 8.0.0 (2017-07-14, Development Milestone)

- Functionality Added or Changed
- Bugs Fixed

Functionality Added or Changed

- Calling the `modify()` or `remove()` function without a parameter caused the function to be executed against the whole collection, which could cause unexpected results such as deleting all rows in a table. To avoid this and make the behavior consistent with `update()` and `delete()`, a client-side exception is now thrown if the `modify()` or `remove()` function is called without a parameter. Now, to execute the `modify()` or `remove()` function against a collection call them with an expression that evaluates to `true`, for example `remove('true')` or `modify('true')`.

Bugs Fixed

- Executing AdminAPI commands on a server with a version of Python lower than 2.7 was failing without the correct error message. (Bug #25975317)
- When using MySQL Shell on Windows any files created or opened, for example those used during `dba.createSandboxInstance()`, could not be deleted. (Bug #25789094)
- The help for `dba.configureLocalInstance(instance[, options])` has been improved to describe the returned JSON object. (Bug #25703028)
- The options in the MySQL Shell options dictionary are now fully documented. (Bug #25701345)
- When using `dba.deploySandboxInstance()` and passing in `sandboxDir`, the specified path must not exceed 89 characters. (Bug #25485035)
- `shell.connect()` did not report an error if an invalid argument was used. An `ArgumentError` is now issued for any invalid argument.

  The following mutually exclusive pairs of options are now checked, and an error is issued if both are specified:

  - `--password` and `--dbPassword`
  - `--user` and `--dbUser`
  - `--port` and `--socket`

  (Bug #25268670)

  References: See also: Bug #24911173.

- `removeInstance()` resulted in unexpected behavior in some cases, for example when an empty password was passed as part of the URI to the instance. (Bug #25111911)

- A number of issues with the output of `shell.help("prompt")` have been corrected. (Bug #25026855, Bug #25242638, Bug #25676343, Bug #25176769)
- MySQL Shell now displays an invalid year as `0000`, matching the behavior of the MySQL prompt, rather than as `0`. (Bug #24912061)
- MySQL Shell did not display fractional seconds for values in DATETIME columns. (Bug #24911885)
- Creating Classic sessions that connect using Unix sockets now uses the correct defaults such as hostname. This resolves the previous limitation of using Unix sockets to connect to InnoDB cluster instances. See MySQL Shell Connections for information on how the defaults are applied to socket connections. (Bug #24848763, Bug #26036466)

  References: See also: Bug #24911068.
- Some issues with the MySQL Shell command line help output were fixed. (Bug #24841749, Bug #24841493, Bug #24910540)
- URIs were incorrectly parsed in MySQL Shell when passwords were hidden. (Bug #24793956)
- `mysqlsh` stopped responding if the `\source` command was given a directory (rather than file) argument. (Bug #23097932, Bug #81060)
- On an instance configured as a multi-threaded slave, in other words `slave_parallel_workers` set to greater than 0, and with `slave_parallel_type=DATABASE`, `dba.checkInstanceConfiguration()` was not detecting that the instance was not correctly configured for InnoDB cluster usage.
- If `removeInstance()` failed due to a connection error, an error was reported but the instance was incorrectly removed from the InnoDB cluster metadata, and remained part of the replication group. The fix ensures the metadata is correctly updated according to the result of `removeInstance`().
- In a situation where a new primary instance was elected, adding a new instance to the cluster resulted in an error due to a failed connection to the previous primary instance.
- The functions that modify server variables, such as `dba.createCluster()` and `dba.validateInstance()` now provide more information in interactive mode output and log output about server variables which are changed when executed.
- Deploying instances to paths with directories that contained spaces was failing without error. Use double backslash to specify such paths, for example `D:\\Cluster\\foo bar`.
- The Cluster object obtained from functions such as `dba.createCluster()` or `dba.getCluster()` became unusable once the Shell session in which the object was created is was connected to a different server. The fix modifies the Cluster object so that:
  - The Cluster object holds an internal reference to the Session from which it was created or retrieved.
  - AdminAPI functions that modify the Cluster are made using the session referenced by the object.