MySQL文档存储

本章介绍了将MySQL作为文件存储的另一种工作方式，有时被称为 "使用NoSQL"。如果你的目的是以传统的（SQL）方式使用MySQL，本章可能与你无关。

传统上，像MySQL这样的关系型数据库通常需要在存储文档之前定义一个模式。本节描述的功能使你能够将MySQL用作文档存储，这是一个无模式的，因此是模式灵活的文档存储系统。例如，当你创建描述产品的文档时，你不需要在存储和操作文档之前知道和定义任何产品的所有可能属性。这与使用关系型数据库和在表中存储产品不同，当时在向数据库添加任何产品之前，必须知道和定义表中的所有列。本章描述的功能使你能够选择如何配置MySQL，只使用文档存储模型，或将文档存储模型的灵活性与关系模型的力量结合起来。

为了将MySQL用作文档存储，你可以使用下列服务器特性。

- X插件使MySQL服务器能够使用X协议与客户进行通信，这是使用MySQL作为文档存储的先决条件。从MySQL 8.0开始，X插件在MySQL服务器中被默认启用。有关验证X Plugin安装以及配置和监控X Plugin的说明，请参见第20.5节 "X Plugin"。

- X协议支持CRUD和SQL操作，通过SASL认证，允许命令的流（管道化），并且在协议和消息层上是可扩展的。与X协议兼容的客户端包括MySQL Shell和MySQL 8.0连接器。

- 使用X协议与MySQL服务器通信的客户端可以使用X DevAPI来开发应用程序。X DevAPI提供了一个现代化的编程接口，设计简单而强大，为既定的行业标准概念提供支持。本章解释了如何开始在MySQL Shell中使用X DevAPI的JavaScript或Python实现作为客户端。请参阅《X DevAPI用户指南》，了解使用X DevAPI的深入教程。

## 20.1 Interfaces to a MySQL Document Store

To work with MySQL as a document store, you use dedicated components and a choice of clients that support communicating with the MySQL server to develop document based applications.

* The following MySQL products support X Protocol and enable you to use X DevAPI in your chosen language to develop applications that communicate with a MySQL Server functioning as a document store:
  + MySQL Shell (which provides implementations of X DevAPI in JavaScript and Python)
  + Connector/C++
  + Connector/J
  + Connector/Node.js
  + Connector/NET
  + Connector/Python
* MySQL Shell is an interactive interface to MySQL supporting JavaScript, Python, or SQL modes. You can use MySQL Shell to prototype applications, execute queries and update data. [Installing MySQL Shell](https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-install.html) has instructions to download and install MySQL Shell.
* The quick-start guides (tutorials) in this chapter help you to get started using MySQL Shell with MySQL as a document store.

The quick-start guide for JavaScript is here: [Section 20.3, “JavaScript Quick-Start Guide: MySQL Shell for Document Store”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript).

The quick-start guide for Python is here: [Section 20.4, “Python Quick-Start Guide: MySQL Shell for Document Store”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python).

* The MySQL Shell User Guide at [MySQL Shell 8.0](https://dev.mysql.com/doc/mysql-shell/8.0/en/) provides detailed information about configuring and using MySQL Shell.

## 20.2 Document Store Concepts

This section explains the concepts introduced as part of using MySQL as a document store.

* [JSON Document](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#document-store-concepts-json-document)
* [Collection](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#document-store-concepts-collection)
* [CRUD Operations](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#document-store-concepts-crud-operation)

### JSON Document

A JSON document is a data structure composed of key-value pairs and is the fundamental structure for using MySQL as document store. For example, the world\_x schema (installed later in this chapter) contains this document:

{

"GNP": 4834,

"\_id": "00005de917d80000000000000023",

"Code": "BWA",

"Name": "Botswana",

"IndepYear": 1966,

"geography": {

"Region": "Southern Africa",

"Continent": "Africa",

"SurfaceArea": 581730

},

"government": {

"HeadOfState": "Festus G. Mogae",

"GovernmentForm": "Republic"

},

"demographics": {

"Population": 1622000,

"LifeExpectancy": 39.29999923706055

}

}

This document shows that the values of keys can be simple data types, such as integers or strings, but can also contain other documents, arrays, and lists of documents. For example, the **geography** key's value consists of multiple key-value pairs. A JSON document is represented internally using the MySQL binary JSON object, through the [**JSON**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\data-types.html#json) MySQL datatype.

The most important differences between a document and the tables known from traditional relational databases are that the structure of a document does not have to be defined in advance, and a collection can contain multiple documents with different structures. Relational tables on the other hand require that their structure be defined, and all rows in the table must contain the same columns.

### Collection

A collection is a container that is used to store JSON documents in a MySQL database. Applications usually run operations against a collection of documents, for example to find a specific document.

### CRUD Operations

The four basic operations that can be issued against a collection are Create, Read, Update and Delete (CRUD). In terms of MySQL this means:

* Create a new document (insertion or addition)
* Read one or more documents (queries)
* Update one or more documents
* Delete one or more documents

## 20.3 JavaScript Quick-Start Guide: MySQL Shell for Document Store

[20.3.1 MySQL Shell](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-shell)

[20.3.2 Download and Import world\_x Database](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-download)

[20.3.3 Documents and Collections](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-documents-collections)

[20.3.4 Relational Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-relational-tables)

[20.3.5 Documents in Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-documents-in-tables)

This quick-start guide provides instructions to begin prototyping document store applications interactively with MySQL Shell. The guide includes the following topics:

* Introduction to MySQL functionality, MySQL Shell, and the **world\_x** example schema.
* Operations to manage collections and documents.
* Operations to manage relational tables.
* Operations that apply to documents within tables.

To follow this quick-start guide you need a MySQL server with X Plugin installed, the default in 8.0, and MySQL Shell to use as the client. [MySQL Shell 8.0](https://dev.mysql.com/doc/mysql-shell/8.0/en/) provides more in-depth information about MySQL Shell. The Document Store is accessed using X DevAPI, and MySQL Shell provides this API in both JavaScript and Python.

### Related Information

* [MySQL Shell 8.0](https://dev.mysql.com/doc/mysql-shell/8.0/en/) provides more in-depth information about MySQL Shell.
* See [Installing MySQL Shell](https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-install.html) and [Section 20.5, “X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin) for more information about the tools used in this quick-start guide.
* [X DevAPI User Guide](https://dev.mysql.com/doc/x-devapi-userguide/en/) provides more examples of using X DevAPI to develop applications which use Document Store.
* A [Python](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python) quick-start guide is also available.

### 20.3.1 MySQL Shell

This quick-start guide assumes a certain level of familiarity with MySQL Shell. The following section is a high level overview, see the MySQL Shell documentation for more information. MySQL Shell is a unified scripting interface to MySQL Server. It supports scripting in JavaScript and Python. JavaScript is the default processing mode.

#### Start MySQL Shell

After you have installed and started MySQL server, connect MySQL Shell to the server instance. You need to know the address of the MySQL server instance you plan to connect to. To be able to use the instance as a Document Store, the server instance must have X Plugin installed and you should connect to the server using X Protocol. For example to connect to the instance **ds1.example.com** on the default X Protocol port of 33060 use the network string ***user*@ds1.example.com:33060**.

Tip

If you connect to the instance using classic MySQL protocol, for example by using the default [**port**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_port) of 3306 instead of the **[mysqlx\_port](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_port)**, you cannot use the Document Store functionality shown in this tutorial. For example the **db** global object is not populated. To use the Document Store, always connect using X Protocol.

If MySQL Shell is not already running, open a terminal window and issue:

**mysqlsh *user*@ds1.example.com:33060/world\_x**

Alternatively, if MySQL Shell is already running use the **\connect** command by issuing:

**\connect *user*@ds1.example.com:33060/world\_x**

You need to specify the address of the MySQL server instance which you want to connect MySQL Shell to. For example in the previous example:

* ***user*** represents the user name of your MySQL account.
* **ds1.example.com** is the hostname of the server instance running MySQL. Replace this with the hostname of the MySQL server instance you are using as a Document Store.
* The default schema for this session is **world\_x**. For instructions on setting up the **world\_x** schema, see [Section 20.3.2, “Download and Import world\_x Database”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-download).

For more information, see [Section 4.2.5, “Connecting to the Server Using URI-Like Strings or Key-Value Pairs”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#connecting-using-uri-or-key-value-pairs).

Once MySQL Shell opens, the **mysql-js>** prompt indicates that the active language for this session is JavaScript.

mysql-js>

MySQL Shell supports input-line editing as follows:

* **left-arrow** and **right-arrow** keys move horizontally within the current input line.
* **up-arrow** and **down-arrow** keys move up and down through the set of previously entered lines.
* **Backspace** deletes the character before the cursor and typing new characters enters them at the cursor position.
* **Enter** sends the current input line to the server.

#### Get Help for MySQL Shell

Type **mysqlsh --help** at the prompt of your command interpreter for a list of command-line options.

**mysqlsh --help**

Type **\help** at the MySQL Shell prompt for a list of available commands and their descriptions.

mysql-js> **\help**

Type **\help** followed by a command name for detailed help about an individual MySQL Shell command. For example, to view help on the **\connect** command, issue:

mysql-js> **\help \connect**

#### Quit MySQL Shell

To quit MySQL Shell, issue the following command:

mysql-js> **\quit**

#### Related Information

* See [Interactive Code Execution](https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-interactive-code-execution.html) for an explanation of how interactive code execution works in MySQL Shell.
* See [Getting Started with MySQL Shell](https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-getting-started.html) to learn about session and connection alternatives.

### 20.3.2 Download and Import world\_x Database

As part of this quick-start guide, an example schema is provided which is referred to as the **world\_x** schema. Many of the examples demonstrate Document Store functionality using this schema. Start your MySQL server so that you can load the **world\_x** schema, then follow these steps:

1. Download [world\_x-db.zip](http://downloads.mysql.com/docs/world_x-db.zip).
2. Extract the installation archive to a temporary location such as /tmp/. Unpacking the archive results in a single file named world\_x.sql.
3. Import the world\_x.sql file to your server. You can either:
   * Start MySQL Shell in SQL mode and import the file by issuing:
   * **mysqlsh -u root --sql --file /tmp/world\_x-db/world\_x.sql**
   * Enter password: **\*\*\*\***
   * Set MySQL Shell to SQL mode while it is running and source the schema file by issuing:
   * **\sql**
   * Switching to SQL mode... Commands end with ;
   * **\source /tmp/world\_x-db/world\_x.sql**

Replace /tmp/ with the path to the world\_x.sql file on your system. Enter your password if prompted. A non-root account can be used as long as the account has privileges to create new schemas.

#### The world\_x Schema

The **world\_x** example schema contains the following JSON collection and relational tables:

* Collection
  + **countryinfo**: Information about countries in the world.
* Tables
  + **country**: Minimal information about countries of the world.
  + **city**: Information about some of the cities in those countries.
  + **countrylanguage**: Languages spoken in each country.

#### Related Information

* [MySQL Shell Sessions](https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-sessions.html) explains session types.

### 20.3.3 Documents and Collections

[20.3.3.1 Create, List, and Drop Collections](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-collections-operations)

[20.3.3.2 Working with Collections](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-collections-add)

[20.3.3.3 Find Documents](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-documents-find)

[20.3.3.4 Modify Documents](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-documents-modify)

[20.3.3.5 Remove Documents](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-documents-remove)

[20.3.3.6 Create and Drop Indexes](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-indexes-create)

When you are using MySQL as a Document Store, collections are containers within a schema that you can create, list, and drop. Collections contain JSON documents that you can add, find, update, and remove.

The examples in this section use the **countryinfo** collection in the **world\_x** schema. For instructions on setting up the **world\_x** schema, see [Section 20.3.2, “Download and Import world\_x Database”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-download).

#### Documents

In MySQL, documents are represented as JSON objects. Internally, they are stored in an efficient binary format that enables fast lookups and updates.

* Simple document format for JavaScript:
* {field1: "value", field2 : 10, "field 3": null}

An array of documents consists of a set of documents separated by commas and enclosed within **[** and **]** characters.

* Simple array of documents for JavaScript:
* [{"Name": "Aruba", "Code:": "ABW"}, {"Name": "Angola", "Code:": "AGO"}]

MySQL supports the following JavaScript value types in JSON documents:

* numbers (integer and floating point)
* strings
* boolean (False and True)
* null
* arrays of more JSON values
* nested (or embedded) objects of more JSON values

#### Collections

Collections are containers for documents that share a purpose and possibly share one or more indexes. Each collection has a unique name and exists within a single schema.

The term schema is equivalent to a database, which means a group of database objects as opposed to a relational schema, used to enforce structure and constraints over data. A schema does not enforce conformity on the documents in a collection.

In this quick-start guide:

* Basic objects include:

| **Object form** | **Description** |
| --- | --- |
| **db** | **db** is a global variable assigned to the current active schema. When you want to run operations against the schema, for example to retrieve a collection, you use methods available for the **db** variable. |
| **db.getCollections()** | [db.getCollections()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-collections-get) returns a list of collections in the schema. Use the list to get references to collection objects, iterate over them, and so on. |

* Basic operations scoped by collections include:

| **Operation form** | **Description** |
| --- | --- |
| **db.*name*.add()** | The [add()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-collections-add) method inserts one document or a list of documents into the named collection. |
| **db.*name*.find()** | The [find()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-documents-find) method returns some or all documents in the named collection. |
| **db.*name*.modify()** | The [modify()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-documents-modify) method updates documents in the named collection. |
| **db.*name*.remove()** | The [remove()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-documents-remove) method deletes one document or a list of documents from the named collection. |

#### Related Information

* See [Working with Collections](https://dev.mysql.com/doc/x-devapi-userguide/en/devapi-users-working-with-collections.html) for a general overview.
* [CRUD EBNF Definitions](https://dev.mysql.com/doc/x-devapi-userguide/en/mysql-x-crud-ebnf-definitions.html) provides a complete list of operations.

#### 20.3.3.1 Create, List, and Drop Collections

In MySQL Shell, you can create new collections, get a list of the existing collections in a schema, and remove an existing collection from a schema. Collection names are case-sensitive and each collection name must be unique.

##### Confirm the Schema

To show the value that is assigned to the schema variable, issue:

mysql-js> **db**

If the schema value is not **Schema:world\_x**, then set the **db** variable by issuing:

mysql-js> **\use world\_x**

##### Create a Collection

To create a new collection in an existing schema, use the **db** object's **createCollection()** method. The following example creates a collection called **flags** in the **world\_x** schema.

mysql-js> **db.createCollection("flags")**

The method returns a collection object.

<Collection:flags>

##### List Collections

To display all collections in the **world\_x** schema, use the **db** object's **getCollections()** method. Collections returned by the server you are currently connected to appear between brackets.

mysql-js> **db.getCollections()**

[

<Collection:countryinfo>,

<Collection:flags>

]

##### Drop a Collection

To drop an existing collection from a schema, use the **db** object's **dropCollection()** method. For example, to drop the **flags** collection from the current schema, issue:

mysql-js> **db.dropCollection("flags")**

The **dropCollection()** method is also used in MySQL Shell to drop a relational table from a schema.

##### Related Information

* See [Collection Objects](https://dev.mysql.com/doc/x-devapi-userguide/en/collection-objects.html) for more examples.

#### 20.3.3.2 Working with Collections

To work with the collections in a schema, use the **db** global object to access the current schema. In this example we are using the **world\_x** schema imported previously, and the **countryinfo** collection. Therefore, the format of the operations you issue is **db.*collection\_name*.operation**, where ***collection\_name*** is the name of the collection which the operation is executed against. In the following examples, the operations are executed against the **countryinfo** collection.

##### Add a Document

Use the **add()** method to insert one document or a list of documents into an existing collection. Insert the following document into the **countryinfo** collection. As this is multi-line content, press **Enter** twice to insert the document.

mysql-js> **db.countryinfo.add(**

**{**

**GNP: .6,**

**IndepYear: 1967,**

**Name: "Sealand",**

**Code: "SEA",**

**demographics: {**

**LifeExpectancy: 79,**

**Population: 27**

**},**

**geography: {**

**Continent: "Europe",**

**Region: "British Islands",**

**SurfaceArea: 193**

**},**

**government: {**

**GovernmentForm: "Monarchy",**

**HeadOfState: "Michael Bates"**

**}**

**}**

**)**

The method returns the status of the operation. You can verify the operation by searching for the document. For example:

mysql-js> **db.countryinfo.find("Name = 'Sealand'")**

{

"GNP": 0.6,

"\_id": "00005e2ff4af00000000000000f4",

"Name": "Sealand",

"Code:": "SEA",

"IndepYear": 1967,

"geography": {

"Region": "British Islands",

"Continent": "Europe",

"SurfaceArea": 193

},

"government": {

"HeadOfState": "Michael Bates",

"GovernmentForm": "Monarchy"

},

"demographics": {

"Population": 27,

"LifeExpectancy": 79

}

}

Note that in addition to the fields specified when the document was added, there is one more field, the **\_id**. Each document requires an identifier field called **\_id**. The value of the **\_id** field must be unique among all documents in the same collection. In MySQL 8.0.11 and higher, document IDs are generated by the server, not the client, so MySQL Shell does not automatically set an **\_id** value. A MySQL server at 8.0.11 or higher sets an **\_id** value if the document does not contain the **\_id** field. A MySQL server at an earlier 8.0 release or at 5.7 does not set an **\_id** value in this situation, so you must specify it explicitly. If you do not, MySQL Shell returns error 5115 Document is missing a required field. For more information see [Understanding Document IDs](https://dev.mysql.com/doc/x-devapi-userguide/en/understanding-automatic-document-ids.html).

##### Related Information

* See [CollectionAddFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-collection-crud-functions.html" \l "crud-ebnf-collectionaddfunction" \t "_top) for the full syntax definition.
* See [Understanding Document IDs](https://dev.mysql.com/doc/x-devapi-userguide/en/understanding-automatic-document-ids.html).

#### 20.3.3.3 Find Documents

You can use the **find()** method to query for and return documents from a collection in a schema. MySQL Shell provides additional methods to use with the **find()** method to filter and sort the returned documents.

MySQL provides the following operators to specify search conditions: **OR** (**||**), **AND** (**&&**), **XOR**, **IS**, **NOT**, **BETWEEN**, **IN**, **LIKE**, **!=**, **<>**, **>**, **>=**, **<**, **<=**, **&**, **|**, **<<**, **>>**, **+**, **-**, **\***, **/**, **~**, and **%**.

##### Find All Documents in a Collection

To return all documents in a collection, use the **find()** method without specifying search conditions. For example, the following operation returns all documents in the **countryinfo** collection.

mysql-js> **db.countryinfo.find()**

[

{

"GNP": 828,

"Code:": "ABW",

"Name": "Aruba",

"IndepYear": null,

"geography": {

"Continent": "North America",

"Region": "Caribbean",

"SurfaceArea": 193

},

"government": {

"GovernmentForm": "Nonmetropolitan Territory of The Netherlands",

"HeadOfState": "Beatrix"

}

"demographics": {

"LifeExpectancy": 78.4000015258789,

"Population": 103000

},

...

}

]

240 documents in set (0.00 sec)

The method produces results that contain operational information in addition to all documents in the collection.

An empty set (no matching documents) returns the following information:

Empty set (0.00 sec)

##### Filter Searches

You can include search conditions with the **find()** method. The syntax for expressions that form a search condition is the same as that of traditional MySQL [Chapter 12, *Functions and Operators*](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html). You must enclose all expressions in quotes. For the sake of brevity, some of the examples do not display output.

A simple search condition could consist of the **Name** field and a value we know is in a document. The following example returns a single document:

mysql-js> **db.countryinfo.find("Name = 'Australia'")**

[

{

"GNP": 351182,

"Code:": "AUS",

"Name": "Australia",

"IndepYear": 1901,

"geography": {

"Continent": "Oceania",

"Region": "Australia and New Zealand",

"SurfaceArea": 7741220

},

"government": {

"GovernmentForm": "Constitutional Monarchy, Federation",

"HeadOfState": "Elisabeth II"

}

"demographics": {

"LifeExpectancy": 79.80000305175781,

"Population": 18886000

},

}

]

The following example searches for all countries that have a GNP higher than $500 billion. The **countryinfo** collection measures GNP in units of million.

mysql-js> **db.countryinfo.find("GNP > 500000")**

...[output removed]

10 documents in set (0.00 sec)

The Population field in the following query is embedded within the demographics object. To access the embedded field, use a period between demographics and Population to identify the relationship. Document and field names are case-sensitive.

mysql-js> **db.countryinfo.find("GNP > 500000 and demographics.Population < 100000000")**

...[output removed]

6 documents in set (0.00 sec)

Arithmetic operators in the following expression are used to query for countries with a GNP per capita higher than $30000. Search conditions can include arithmetic operators and most MySQL functions.

**Note**

Seven documents in the **countryinfo** collection have a population value of zero. Therefore warning messages appear at the end of the output.

mysql-js> **db.countryinfo.find("GNP\*1000000/demographics.Population > 30000")**

...[output removed]

9 documents in set, 7 warnings (0.00 sec)

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

You can separate a value from the search condition by using the **bind()** method. For example, instead of specifying a hard-coded country name as the condition, substitute a named placeholder consisting of a colon followed by a name that begins with a letter, such as country. Then use the **bind(*placeholder*, *value*)** method as follows:

mysql-js> **db.countryinfo.find("Name = :country").bind("country", "Italy")**

{

"GNP": 1161755,

"\_id": "00005de917d8000000000000006a",

"Code": "ITA",

"Name": "Italy",

"Airports": [],

"IndepYear": 1861,

"geography": {

"Region": "Southern Europe",

"Continent": "Europe",

"SurfaceArea": 301316

},

"government": {

"HeadOfState": "Carlo Azeglio Ciampi",

"GovernmentForm": "Republic"

},

"demographics": {

"Population": 57680000,

"LifeExpectancy": 79

}

}

1 document in set (0.01 sec)

Tip

Within a program, binding enables you to specify placeholders in your expressions, which are filled in with values before execution and can benefit from automatic escaping, as appropriate.

Always use binding to sanitize input. Avoid introducing values in queries using string concatenation, which can produce invalid input and, in some cases, can cause security issues.

You can use placeholders and the **bind()** method to create saved searches which you can then call with different values. For example to create a saved search for a country:

mysql-js> **var myFind = db.countryinfo.find("Name = :country")**

mysql-js> **myFind.bind('country', 'France')**

{

"GNP": 1424285,

"\_id": "00005de917d80000000000000048",

"Code": "FRA",

"Name": "France",

"IndepYear": 843,

"geography": {

"Region": "Western Europe",

"Continent": "Europe",

"SurfaceArea": 551500

},

"government": {

"HeadOfState": "Jacques Chirac",

"GovernmentForm": "Republic"

},

"demographics": {

"Population": 59225700,

"LifeExpectancy": 78.80000305175781

}

}

1 document in set (0.0028 sec)

mysql-js> **myFind.bind('country', 'Germany')**

{

"GNP": 2133367,

"\_id": "00005de917d80000000000000038",

"Code": "DEU",

"Name": "Germany",

"IndepYear": 1955,

"geography": {

"Region": "Western Europe",

"Continent": "Europe",

"SurfaceArea": 357022

},

"government": {

"HeadOfState": "Johannes Rau",

"GovernmentForm": "Federal Republic"

},

"demographics": {

"Population": 82164700,

"LifeExpectancy": 77.4000015258789

}

}

1 document in set (0.0026 sec)

##### Project Results

You can return specific fields of a document, instead of returning all the fields. The following example returns the GNP and Name fields of all documents in the **countryinfo** collection matching the search conditions.

Use the **fields()** method to pass the list of fields to return.

mysql-js> **db.countryinfo.find("GNP > 5000000").fields(["GNP", "Name"])**

[

{

"GNP": 8510700,

"Name": "United States"

}

]

1 document in set (0.00 sec)

In addition, you can alter the returned documents—adding, renaming, nesting and even computing new field values—with an expression that describes the document to return. For example, alter the names of the fields with the following expression to return only two documents.

mysql-js> **db.countryinfo.find().fields(**

**mysqlx.expr('{"Name": upper(Name), "GNPPerCapita": GNP\*1000000/demographics.Population}')).limit(2)**

{

"Name": "ARUBA",

"GNPPerCapita": 8038.834951456311

}

{

"Name": "AFGHANISTAN",

"GNPPerCapita": 263.0281690140845

}

##### Limit, Sort, and Skip Results

You can apply the **limit()**, **sort()**, and **skip()** methods to manage the number and order of documents returned by the **find()** method.

To specify the number of documents included in a result set, append the **limit()** method with a value to the **find()** method. The following query returns the first five documents in the **countryinfo** collection.

mysql-js> **db.countryinfo.find().limit(5)**

... [output removed]

5 documents in set (0.00 sec)

To specify an order for the results, append the **sort()** method to the **find()** method. Pass to the **sort()** method a list of one or more fields to sort by and, optionally, the descending (**desc**) or ascending (**asc**) attribute as appropriate. Ascending order is the default order type.

For example, the following query sorts all documents by the IndepYear field and then returns the first eight documents in descending order.

mysql-js> **db.countryinfo.find().sort(["IndepYear desc"]).limit(8)**

... [output removed]

8 documents in set (0.00 sec)

By default, the **limit()** method starts from the first document in the collection. You can use the **skip()** method to change the starting document. For example, to ignore the first document and return the next eight documents matching the condition, pass to the **skip()** method a value of 1.

mysql-js> **db.countryinfo.find().sort(["IndepYear desc"]).limit(8).skip(1)**

... [output removed]

8 documents in set (0.00 sec)

##### Related Information

* The [MySQL Reference Manual](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html) provides detailed documentation on functions and operators.
* See [CollectionFindFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-collection-crud-functions.html" \l "crud-ebnf-collectionfindfunction" \t "_top) for the full syntax definition.

#### 20.3.3.4 Modify Documents

You can use the **modify()** method to update one or more documents in a collection. The X DevAPI provides additional methods for use with the **modify()** method to:

* Set and unset fields within documents.
* Append, insert, and delete arrays.
* Bind, limit, and sort the documents to be modified.

##### Set and Unset Document Fields

The **modify()** method works by filtering a collection to include only the documents to be modified and then applying the operations that you specify to those documents.

In the following example, the **modify()** method uses the search condition to identify the document to change and then the **set()** method replaces two values within the nested demographics object.

mysql-js> **db.countryinfo.modify("Code = 'SEA'").set(**

**"demographics", {"LifeExpectancy": 78, "Population": 28})**

After you modify a document, use the **find()** method to verify the change.

To remove content from a document, use the **modify()** and **unset()** methods. For example, the following query removes the GNP from a document that matches the search condition.

mysql-js> **db.countryinfo.modify("Name = 'Sealand'").unset("GNP")**

Use the **find()** method to verify the change.

mysql-js> **db.countryinfo.find("Name = 'Sealand'")**

{

"\_id": "00005e2ff4af00000000000000f4",

"Name": "Sealand",

"Code:": "SEA",

"IndepYear": 1967,

"geography": {

"Region": "British Islands",

"Continent": "Europe",

"SurfaceArea": 193

},

"government": {

"HeadOfState": "Michael Bates",

"GovernmentForm": "Monarchy"

},

"demographics": {

"Population": 27,

"LifeExpectancy": 79

}

}

##### Append, Insert, and Delete Arrays

To append an element to an array field, or insert, or delete elements in an array, use the **arrayAppend()**, **arrayInsert()**, or **arrayDelete()** methods. The following examples modify the **countryinfo** collection to enable tracking of international airports.

The first example uses the **modify()** and **set()** methods to create a new Airports field in all documents.

**Caution**

Use care when you modify documents without specifying a search condition; doing so modifies all documents in the collection.

mysql-js> **db.countryinfo.modify("true").set("Airports", [])**

With the Airports field added, the next example uses the **arrayAppend()** method to add a new airport to one of the documents. $.Airports in the following example represents the Airports field of the current document.

mysql-js> **db.countryinfo.modify("Name = 'France'").arrayAppend("$.Airports", "ORY")**

Use **find()** to see the change.

mysql-js> **db.countryinfo.find("Name = 'France'")**

{

"GNP": 1424285,

"\_id": "00005de917d80000000000000048",

"Code": "FRA",

"Name": "France",

"Airports": [

"ORY"

],

"IndepYear": 843,

"geography": {

"Region": "Western Europe",

"Continent": "Europe",

"SurfaceArea": 551500

},

"government": {

"HeadOfState": "Jacques Chirac",

"GovernmentForm": "Republic"

},

"demographics": {

"Population": 59225700,

"LifeExpectancy": 78.80000305175781

}

}

To insert an element at a different position in the array, use the **arrayInsert()** method to specify which index to insert in the path expression. In this case, the index is 0, or the first element in the array.

mysql-js> **db.countryinfo.modify("Name = 'France'").arrayInsert("$.Airports[0]", "CDG")**

To delete an element from the array, you must pass to the **arrayDelete()** method the index of the element to be deleted.

mysql-js> **db.countryinfo.modify("Name = 'France'").arrayDelete("$.Airports[1]")**

##### Related Information

* The [MySQL Reference Manual](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\data-types.html#json-paths) provides instructions to help you search for and modify JSON values.
* See [CollectionModifyFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-collection-crud-functions.html" \l "crud-ebnf-collectionmodifyfunction" \t "_top) for the full syntax definition.

#### 20.3.3.5 Remove Documents

You can use the **remove()** method to delete some or all documents from a collection in a schema. The X DevAPI provides additional methods for use with the **remove()** method to filter and sort the documents to be removed.

##### Remove Documents Using Conditions

The following example passes a search condition to the **remove()** method. All documents matching the condition are removed from the **countryinfo** collection. In this example, one document matches the condition.

mysql-js> **db.countryinfo.remove("Code = 'SEA'")**

##### Remove the First Document

To remove the first document in the **countryinfo** collection, use the **limit()** method with a value of 1.

mysql-js> **db.countryinfo.remove("true").limit(1)**

##### Remove the Last Document in an Order

The following example removes the last document in the **countryinfo** collection by country name.

mysql-js> **db.countryinfo.remove("true").sort(["Name desc"]).limit(1)**

##### Remove All Documents in a Collection

You can remove all documents in a collection. To do so, use the **remove("true")** method without specifying a search condition.

**Caution**

Use care when you remove documents without specifying a search condition. This action deletes all documents from the collection.

Alternatively, use the **db.drop\_collection('countryinfo')** operation to delete the **countryinfo** collection.

##### Related Information

* See [CollectionRemoveFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-collection-crud-functions.html" \l "crud-ebnf-collectionremovefunction" \t "_top) for the full syntax definition.
* See [Section 20.3.2, “Download and Import world\_x Database”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-download) for instructions to recreate the **world\_x** schema.

#### 20.3.3.6 Create and Drop Indexes

Indexes are used to find documents with specific field values quickly. Without an index, MySQL must begin with the first document and then read through the entire collection to find the relevant fields. The larger the collection, the more this costs. If a collection is large and queries on a specific field are common, then consider creating an index on a specific field inside a document.

For example, the following query performs better with an index on the Population field:

mysql-js> **db.countryinfo.find("demographics.Population < 100")**

...[output removed]

8 documents in set (0.00 sec)

The **createIndex()** method creates an index that you can define with a JSON document that specifies which fields to use. This section is a high level overview of indexing. For more information see [Indexing Collections](https://dev.mysql.com/doc/x-devapi-userguide/en/collection-indexing.html).

##### Add a Nonunique Index

To create a nonunique index, pass an index name and the index information to the **createIndex()** method. Duplicate index names are prohibited.

The following example specifies an index named **popul**, defined against the **Population** field from the **demographics** object, indexed as an **Integer** numeric value. The final parameter indicates whether the field should require the **NOT NULL** constraint. If the value is **false**, the field can contain **NULL** values. The index information is a JSON document with details of one or more fields to include in the index. Each field definition must include the full document path to the field, and specify the type of the field.

mysql-js> **db.countryinfo.createIndex("popul", {fields:**

**[{field: '$.demographics.Population', type: 'INTEGER'}]})**

Here, the index is created using an integer numeric value. Further options are available, including options for use with GeoJSON data. You can also specify the type of index, which has been omitted here because the default type “index” is appropriate.

##### Add a Unique Index

To create a unique index, pass an index name, the index definition, and the index type “unique” to the **createIndex()** method. This example shows a unique index created on the country name (**"Name"**), which is another common field in the **countryinfo** collection to index. In the index field description, **"TEXT(40)"** represents the number of characters to index, and **"required": True** specifies that the field is required to exist in the document.

mysql-js> **db.countryinfo.createIndex("name",**

**{"fields": [{"field": "$.Name", "type": "TEXT(40)", "required": true}], "unique": true})**

##### Drop an Index

To drop an index, pass the name of the index to drop to the **dropIndex()** method. For example, you can drop the “popul” index as follows:

mysql-js> **db.countryinfo.dropIndex("popul")**

##### Related Information

* See [Indexing Collections](https://dev.mysql.com/doc/x-devapi-userguide/en/collection-indexing.html) for more information.
* See [Defining an Index](https://dev.mysql.com/doc/x-devapi-userguide/en/collection-indexing.html#collection-index-definitions) for more information on the JSON document that defines an index.
* See [Collection Index Management Functions](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-collection-index-management-functions.html) for the full syntax definition.

### 20.3.4 Relational Tables

[20.3.4.1 Insert Records into Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-table-insert)

[20.3.4.2 Select Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-table-select)

[20.3.4.3 Update Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-table-update)

[20.3.4.4 Delete Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-table-delete)

You can also use X DevAPI to work with relational tables. In MySQL, each relational table is associated with a particular storage engine. The examples in this section use **[InnoDB](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\innodb-storage-engine.html" \o "Chapter 15 The InnoDB Storage Engine)** tables in the **world\_x** schema.

#### Confirm the Schema

To show the schema that is assigned to the **db** global variable, issue **db**.

mysql-js> **db**

<Schema:world\_x>

If the returned value is not **Schema:world\_x**, set the **db** variable as follows:

mysql-js> **\use world\_x**

Schema `world\_x` accessible through db.

#### Show All Tables

To display all relational tables in the **world\_x** schema, use the **getTables()** method on the **db** object.

mysql-js> **db.getTables()**

{

"city": <Table:city>,

"country": <Table:country>,

"countrylanguage": <Table:countrylanguage>

}

#### Basic Table Operations

Basic operations scoped by tables include:

| **Operation form** | **Description** |
| --- | --- |
| **db.*name*.insert()** | The [insert()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-table-insert) method inserts one or more records into the named table. |
| **db.*name*.select()** | The [select()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-table-select) method returns some or all records in the named table. |
| **db.*name*.update()** | The [update()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-table-update) method updates records in the named table. |
| **db.*name*.delete()** | The [delete()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-table-delete) method deletes one or more records from the named table. |

#### Related Information

* See [Working with Relational Tables](https://dev.mysql.com/doc/x-devapi-userguide/en/devapi-users-working-with-relational-tables.html) for more information.
* [CRUD EBNF Definitions](https://dev.mysql.com/doc/x-devapi-userguide/en/mysql-x-crud-ebnf-definitions.html) provides a complete list of operations.
* See [Section 20.3.2, “Download and Import world\_x Database”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-download) for instructions on setting up the **world\_x** schema sample.

#### 20.3.4.1 Insert Records into Tables

You can use the **insert()** method with the **values()** method to insert records into an existing relational table. The **insert()** method accepts individual columns or all columns in the table. Use one or more **values()** methods to specify the values to be inserted.

##### Insert a Complete Record

To insert a complete record, pass to the **insert()** method all columns in the table. Then pass to the **values()** method one value for each column in the table. For example, to add a new record to the city table in the **world\_x** schema, insert the following record and press **Enter** twice.

mysql-js> **db.city.insert("ID", "Name", "CountryCode", "District", "Info").values(**

**None, "Olympia", "USA", "Washington", '{"Population": 5000}')**

The city table has five columns: ID, Name, CountryCode, District, and Info. Each value must match the data type of the column it represents.

##### Insert a Partial Record

The following example inserts values into the ID, Name, and CountryCode columns of the city table.

mysql-js> **db.city.insert("ID", "Name", "CountryCode").values(**

**None, "Little Falls", "USA").values(None, "Happy Valley", "USA")**

When you specify columns using the **insert()** method, the number of values must match the number of columns. In the previous example, you must supply three values to match the three columns specified.

##### Related Information

* See [TableInsertFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-table-crud-functions.html" \l "crud-ebnf-tableinsertfunction" \t "_top) for the full syntax definition.

#### 20.3.4.2 Select Tables

You can use the **select()** method to query for and return records from a table in a database. The X DevAPI provides additional methods to use with the **select()** method to filter and sort the returned records.

MySQL provides the following operators to specify search conditions: **OR** (**||**), **AND** (**&&**), **XOR**, **IS**, **NOT**, **BETWEEN**, **IN**, **LIKE**, **!=**, **<>**, **>**, **>=**, **<**, **<=**, **&**, **|**, **<<**, **>>**, **+**, **-**, **\***, **/**, **~**, and **%**.

##### Select All Records

To issue a query that returns all records from an existing table, use the **select()** method without specifying search conditions. The following example selects all records from the city table in the **world\_x** database.

**Note**

Limit the use of the empty **select()** method to interactive statements. Always use explicit column-name selections in your application code.

mysql-js> **db.city.select()**

+------+------------+-------------+------------+-------------------------+

| ID | Name | CountryCode | District | Info |

+------+------------+-------------+------------+-------------------------+

| 1 | Kabul | AFG | Kabol |{"Population": 1780000} |

| 2 | Qandahar | AFG | Qandahar |{"Population": 237500} |

| 3 | Herat | AFG | Herat |{"Population": 186800} |

... ... ... ... ...

| 4079 | Rafah | PSE | Rafah |{"Population": 92020} |

+------+------- ----+-------------+------------+-------------------------+

4082 rows in set (0.01 sec)

An empty set (no matching records) returns the following information:

Empty set (0.00 sec)

##### Filter Searches

To issue a query that returns a set of table columns, use the **select()** method and specify the columns to return between square brackets. This query returns the Name and CountryCode columns from the city table.

mysql-js> **db.city.select(["Name", "CountryCode"])**

+-------------------+-------------+

| Name | CountryCode |

+-------------------+-------------+

| Kabul | AFG |

| Qandahar | AFG |

| Herat | AFG |

| Mazar-e-Sharif | AFG |

| Amsterdam | NLD |

... ...

| Rafah | PSE |

| Olympia | USA |

| Little Falls | USA |

| Happy Valley | USA |

+-------------------+-------------+

4082 rows in set (0.00 sec)

To issue a query that returns rows matching specific search conditions, use the **where()** method to include those conditions. For example, the following example returns the names and country codes of the cities that start with the letter Z.

mysql-js> **db.city.select(["Name", "CountryCode"]).where("Name like 'Z%'")**

+-------------------+-------------+

| Name | CountryCode |

+-------------------+-------------+

| Zaanstad | NLD |

| Zoetermeer | NLD |

| Zwolle | NLD |

| Zenica | BIH |

| Zagazig | EGY |

| Zaragoza | ESP |

| Zamboanga | PHL |

| Zahedan | IRN |

| Zanjan | IRN |

| Zabol | IRN |

| Zama | JPN |

| Zhezqazghan | KAZ |

| Zhengzhou | CHN |

... ...

| Zeleznogorsk | RUS |

+-------------------+-------------+

59 rows in set (0.00 sec)

You can separate a value from the search condition by using the **bind()** method. For example, instead of using "Name = 'Z%' " as the condition, substitute a named placeholder consisting of a colon followed by a name that begins with a letter, such as name. Then include the placeholder and value in the **bind()** method as follows:

mysql-js> **db.city.select(["Name", "CountryCode"]).**

**where("Name like :name").bind("name", "Z%")**

Tip

Within a program, binding enables you to specify placeholders in your expressions, which are filled in with values before execution and can benefit from automatic escaping, as appropriate.

Always use binding to sanitize input. Avoid introducing values in queries using string concatenation, which can produce invalid input and, in some cases, can cause security issues.

##### Project Results

To issue a query using the [**AND**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html#operator_and) operator, add the operator between search conditions in the **where()** method.

mysql-js> **db.city.select(["Name", "CountryCode"]).where(**

**"Name like 'Z%' and CountryCode = 'CHN'")**

+----------------+-------------+

| Name | CountryCode |

+----------------+-------------+

| Zhengzhou | CHN |

| Zibo | CHN |

| Zhangjiakou | CHN |

| Zhuzhou | CHN |

| Zhangjiang | CHN |

| Zigong | CHN |

| Zaozhuang | CHN |

... ...

| Zhangjiagang | CHN |

+----------------+-------------+

22 rows in set (0.01 sec)

To specify multiple conditional operators, you can enclose the search conditions in parenthesis to change the operator precedence. The following example demonstrates the placement of [**AND**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html#operator_and) and [**OR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html#operator_or) operators.

mysql-js> **db.city.select(["Name", "CountryCode"]).**

**where("Name like 'Z%' and (CountryCode = 'CHN' or CountryCode = 'RUS')")**

+-------------------+-------------+

| Name | CountryCode |

+-------------------+-------------+

| Zhengzhou | CHN |

| Zibo | CHN |

| Zhangjiakou | CHN |

| Zhuzhou | CHN |

... ...

| Zeleznogorsk | RUS |

+-------------------+-------------+

29 rows in set (0.01 sec)

##### Limit, Order, and Offset Results

You can apply the **limit()**, **orderBy()**, and **offSet()** methods to manage the number and order of records returned by the **select()** method.

To specify the number of records included in a result set, append the **limit()** method with a value to the **select()** method. For example, the following query returns the first five records in the country table.

mysql-js> **db.country.select(["Code", "Name"]).limit(5)**

+------+-------------+

| Code | Name |

+------+-------------+

| ABW | Aruba |

| AFG | Afghanistan |

| AGO | Angola |

| AIA | Anguilla |

| ALB | Albania |

+------+-------------+

5 rows in set (0.00 sec)

To specify an order for the results, append the **orderBy()** method to the **select()** method. Pass to the **orderBy()** method a list of one or more columns to sort by and, optionally, the descending (**desc**) or ascending (**asc**) attribute as appropriate. Ascending order is the default order type.

For example, the following query sorts all records by the Name column and then returns the first three records in descending order .

mysql-js> **db.country.select(["Code", "Name"]).orderBy(["Name desc"]).limit(3)**

+------+------------+

| Code | Name |

+------+------------+

| ZWE | Zimbabwe |

| ZMB | Zambia |

| YUG | Yugoslavia |

+------+------------+

3 rows in set (0.00 sec)

By default, the **limit()** method starts from the first record in the table. You can use the **offset()** method to change the starting record. For example, to ignore the first record and return the next three records matching the condition, pass to the **offset()** method a value of 1.

mysql-js> **db.country.select(["Code", "Name"]).orderBy(["Name desc"]).limit(3).offset(1)**

+------+------------+

| Code | Name |

+------+------------+

| ZMB | Zambia |

| YUG | Yugoslavia |

| YEM | Yemen |

+------+------------+

3 rows in set (0.00 sec)

##### Related Information

* The [MySQL Reference Manual](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html) provides detailed documentation on functions and operators.
* See [TableSelectFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-table-crud-functions.html" \l "crud-ebnf-tableselectfunction" \t "_top) for the full syntax definition.

#### 20.3.4.3 Update Tables

You can use the **update()** method to modify one or more records in a table. The **update()** method works by filtering a query to include only the records to be updated and then applying the operations you specify to those records.

To replace a city name in the city table, pass to the **set()** method the new city name. Then, pass to the **where()** method the city name to locate and replace. The following example replaces the city Peking with Beijing.

mysql-js> **db.city.update().set("Name", "Beijing").where("Name = 'Peking'")**

Use the **select()** method to verify the change.

mysql-js> **db.city.select(["ID", "Name", "CountryCode", "District", "Info"]).where("Name = 'Beijing'")**

+------+-----------+-------------+----------+-----------------------------+

| ID | Name | CountryCode | District | Info |

+------+-----------+-------------+----------+-----------------------------+

| 1891 | Beijing | CHN | Peking | {"Population": 7472000} |

+------+-----------+-------------+----------+-----------------------------+

1 row in set (0.00 sec)

##### Related Information

* See [TableUpdateFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-table-crud-functions.html" \l "crud-ebnf-tableupdatefunction" \t "_top) for the full syntax definition.

#### 20.3.4.4 Delete Tables

You can use the **delete()** method to remove some or all records from a table in a database. The X DevAPI provides additional methods to use with the **delete()** method to filter and order the records to be deleted.

##### Delete Records Using Conditions

The following example passes search conditions to the **delete()** method. All records matching the condition are deleted from the city table. In this example, one record matches the condition.

mysql-js> **db.city.delete().where("Name = 'Olympia'")**

##### Delete the First Record

To delete the first record in the city table, use the **limit()** method with a value of 1.

mysql-js> **db.city.delete().limit(1)**

##### Delete All Records in a Table

You can delete all records in a table. To do so, use the **delete()** method without specifying a search condition.

**Caution**

Use care when you delete records without specifying a search condition; doing so deletes all records from the table.

##### Drop a Table

The **dropCollection()** method is also used in MySQL Shell to drop a relational table from a database. For example, to drop the **citytest** table from the **world\_x** database, issue:

mysql-js> **session.dropCollection("world\_x", "citytest")**

##### Related Information

* See [TableDeleteFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-table-crud-functions.html" \l "crud-ebnf-tabledeletefunction" \t "_top) for the full syntax definition.
* See [Section 20.3.2, “Download and Import world\_x Database”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript-download) for instructions to recreate the **world\_x** database.

### 20.3.5 Documents in Tables

In MySQL, a table may contain traditional relational data, JSON values, or both. You can combine traditional data with JSON documents by storing the documents in columns having a native [**JSON**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\data-types.html#json) data type.

Examples in this section use the city table in the **world\_x** schema.

#### city Table Description

The city table has five columns (or fields).

+---------------+------------+-------+-------+---------+------------------+

| Field | Type | Null | Key | Default | Extra |

+---------------+------------+-------+-------+---------+------------------+

| ID | int(11) | NO | PRI | null | auto\_increment |

| Name | char(35) | NO | | | |

| CountryCode | char(3) | NO | | | |

| District | char(20) | NO | | | |

| Info | json | YES | | null | |

+---------------+------------+-------+-------+---------+------------------+

#### Insert a Record

To insert a document into the column of a table, pass to the **values()** method a well-formed JSON document in the correct order. In the following example, a document is passed as the final value to be inserted into the Info column.

mysql-js> **db.city.insert().values(**

**None, "San Francisco", "USA", "California", '{"Population":830000}')**

#### Select a Record

You can issue a query with a search condition that evaluates document values in the expression.

mysql-js> **db.city.select(["ID", "Name", "CountryCode", "District", "Info"]).where(**

**"CountryCode = :country and Info->'$.Population' > 1000000").bind(**

**'country', 'USA')**

+------+----------------+-------------+----------------+-----------------------------+

| ID | Name | CountryCode | District | Info |

+------+----------------+-------------+----------------+-----------------------------+

| 3793 | New York | USA | New York | {"Population": 8008278} |

| 3794 | Los Angeles | USA | California | {"Population": 3694820} |

| 3795 | Chicago | USA | Illinois | {"Population": 2896016} |

| 3796 | Houston | USA | Texas | {"Population": 1953631} |

| 3797 | Philadelphia | USA | Pennsylvania | {"Population": 1517550} |

| 3798 | Phoenix | USA | Arizona | {"Population": 1321045} |

| 3799 | San Diego | USA | California | {"Population": 1223400} |

| 3800 | Dallas | USA | Texas | {"Population": 1188580} |

| 3801 | San Antonio | USA | Texas | {"Population": 1144646} |

+------+----------------+-------------+----------------+-----------------------------+

9 rows in set (0.01 sec)

#### Related Information

* See [Working with Relational Tables and Documents](https://dev.mysql.com/doc/x-devapi-userguide/en/devapi-users-working-with-relational-tables-and-documents.html) for more information.
* See [Section 11.5, “The JSON Data Type”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\data-types.html#json) for a detailed description of the data type.

## 20.4 Python Quick-Start Guide: MySQL Shell for Document Store

[20.4.1 MySQL Shell](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-shell)

[20.4.2 Download and Import world\_x Database](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-download)

[20.4.3 Documents and Collections](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-collections)

[20.4.4 Relational Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-relational-tables)

[20.4.5 Documents in Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-in-tables)

This quick-start guide provides instructions to begin prototyping document store applications interactively with MySQL Shell. The guide includes the following topics:

* Introduction to MySQL functionality, MySQL Shell, and the **world\_x** example schema.
* Operations to manage collections and documents.
* Operations to manage relational tables.
* Operations that apply to documents within tables.

To follow this quick-start guide you need a MySQL server with X Plugin installed, the default in 8.0, and MySQL Shell to use as the client. MySQL Shell includes X DevAPI, implemented in both JavaScript and Python, which enables you to connect to the MySQL server instance using X Protocol and use the server as a Document Store.

### Related Information

* [MySQL Shell 8.0](https://dev.mysql.com/doc/mysql-shell/8.0/en/) provides more in-depth information about MySQL Shell.
* See [Installing MySQL Shell](https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-install.html) and [Section 20.5, “X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin) for more information about the tools used in this quick-start guide.
* See [Supported Languages](https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-features.html#shell-supported-languages) for more information about the languages MySQL Shell supports.
* [X DevAPI User Guide](https://dev.mysql.com/doc/x-devapi-userguide/en/) provides more examples of using X DevAPI to develop applications which use MySQL as a Document Store.
* A [JavaScript](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-javascript) quick-start guide is also available.

### 20.4.1 MySQL Shell

This quick-start guide assumes a certain level of familiarity with MySQL Shell. The following section is a high level overview, see the MySQL Shell documentation for more information. MySQL Shell is a unified scripting interface to MySQL Server. It supports scripting in JavaScript and Python. JavaScript is the default processing mode.

#### Start MySQL Shell

After you have installed and started MySQL server, connect MySQL Shell to the server instance. You need to know the address of the MySQL server instance you plan to connect to. To be able to use the instance as a Document Store, the server instance must have X Plugin installed and you should connect to the server using X Protocol. For example to connect to the instance **ds1.example.com** on the default X Protocol port of 33060 use the network string ***user*@ds1.example.com:33060**.

Tip

If you connect to the instance using classic MySQL protocol, for example by using the default [**port**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_port) of 3306 instead of the **[mysqlx\_port](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_port)**, you cannot use the Document Store functionality shown in this tutorial. For example the **db** global object is not populated. To use the Document Store, always connect using X Protocol.

If MySQL Shell is not already running, open a terminal window and issue:

**mysqlsh *user*@ds1.example.com:33060/world\_x**

Alternatively, if MySQL Shell is already running use the **\connect** command by issuing:

**\connect *user*@ds1.example.com:33060/world\_x**

You need to specify the address of the MySQL server instance which you want to connect MySQL Shell to. For example in the previous example:

* ***user*** represents the user name of your MySQL account.
* **ds1.example.com** is the hostname of the server instance running MySQL. Replace this with the hostname of the MySQL server instance you are using as a Document Store.
* The default schema for this session is **world\_x**. For instructions on setting up the **world\_x** schema, see [Section 20.4.2, “Download and Import world\_x Database”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-download).

For more information, see [Section 4.2.5, “Connecting to the Server Using URI-Like Strings or Key-Value Pairs”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#connecting-using-uri-or-key-value-pairs).

Once MySQL Shell opens, the **mysql-js>** prompt indicates that the active language for this session is JavaScript. To switch MySQL Shell to Python mode, use the **\py** command.

mysql-js> **\py**

Switching to Python mode...

mysql-py>

MySQL Shell supports input-line editing as follows:

* **left-arrow** and **right-arrow** keys move horizontally within the current input line.
* **up-arrow** and **down-arrow** keys move up and down through the set of previously entered lines.
* **Backspace** deletes the character before the cursor and typing new characters enters them at the cursor position.
* **Enter** sends the current input line to the server.

#### Get Help for MySQL Shell

Type **mysqlsh --help** at the prompt of your command interpreter for a list of command-line options.

**mysqlsh --help**

Type **\help** at the MySQL Shell prompt for a list of available commands and their descriptions.

mysql-py> **\help**

Type **\help** followed by a command name for detailed help about an individual MySQL Shell command. For example, to view help on the **\connect** command, issue:

mysql-py> **\help \connect**

#### Quit MySQL Shell

To quit MySQL Shell, issue the following command:

mysql-py> **\quit**

#### Related Information

* See [Interactive Code Execution](https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-interactive-code-execution.html) for an explanation of how interactive code execution works in MySQL Shell.
* See [Getting Started with MySQL Shell](https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-getting-started.html) to learn about session and connection alternatives.

### 20.4.2 Download and Import world\_x Database

As part of this quick-start guide, an example schema is provided which is referred to as the **world\_x** schema. Many of the examples demonstrate Document Store functionality using this schema. Start your MySQL server so that you can load the **world\_x** schema, then follow these steps:

1. Download [world\_x-db.zip](http://downloads.mysql.com/docs/world_x-db.zip).
2. Extract the installation archive to a temporary location such as /tmp/. Unpacking the archive results in a single file named world\_x.sql.
3. Import the world\_x.sql file to your server. You can either:
   * Start MySQL Shell in SQL mode and import the file by issuing:
   * **mysqlsh -u root --sql --file /tmp/world\_x-db/world\_x.sql**
   * Enter password: **\*\*\*\***
   * Set MySQL Shell to SQL mode while it is running and source the schema file by issuing:
   * **\sql**
   * Switching to SQL mode... Commands end with ;
   * **\source /tmp/world\_x-db/world\_x.sql**

Replace /tmp/ with the path to the world\_x.sql file on your system. Enter your password if prompted. A non-root account can be used as long as the account has privileges to create new schemas.

#### The world\_x Schema

The **world\_x** example schema contains the following JSON collection and relational tables:

* Collection
  + **countryinfo**: Information about countries in the world.
* Tables
  + **country**: Minimal information about countries of the world.
  + **city**: Information about some of the cities in those countries.
  + **countrylanguage**: Languages spoken in each country.

#### Related Information

* [MySQL Shell Sessions](https://dev.mysql.com/doc/mysql-shell/8.0/en/mysql-shell-sessions.html) explains session types.

### 20.4.3 Documents and Collections

[20.4.3.1 Create, List, and Drop Collections](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-collections-operations)

[20.4.3.2 Working with Collections](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-add)

[20.4.3.3 Find Documents](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-find)

[20.4.3.4 Modify Documents](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-modify)

[20.4.3.5 Remove Documents](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-remove)

[20.4.3.6 Create and Drop Indexes](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-index)

When you are using MySQL as a Document Store, collections are containers within a schema that you can create, list, and drop. Collections contain JSON documents that you can add, find, update, and remove.

The examples in this section use the **countryinfo** collection in the **world\_x** schema. For instructions on setting up the **world\_x** schema, see [Section 20.4.2, “Download and Import world\_x Database”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-download).

#### Documents

In MySQL, documents are represented as JSON objects. Internally, they are stored in an efficient binary format that enables fast lookups and updates.

* Simple document format for Python:
* {"field1": "value", "field2" : 10, "field 3": null}

An array of documents consists of a set of documents separated by commas and enclosed within **[** and **]** characters.

* Simple array of documents for Python:
* [{"Name": "Aruba", "Code:": "ABW"}, {"Name": "Angola", "Code:": "AGO"}]

MySQL supports the following Python value types in JSON documents:

* numbers (integer and floating point)
* strings
* boolean (False and True)
* None
* arrays of more JSON values
* nested (or embedded) objects of more JSON values

#### Collections

Collections are containers for documents that share a purpose and possibly share one or more indexes. Each collection has a unique name and exists within a single schema.

The term schema is equivalent to a database, which means a group of database objects as opposed to a relational schema, used to enforce structure and constraints over data. A schema does not enforce conformity on the documents in a collection.

In this quick-start guide:

* Basic objects include:

| **Object form** | **Description** |
| --- | --- |
| **db** | **db** is a global variable assigned to the current active schema. When you want to run operations against the schema, for example to retrieve a collection, you use methods available for the **db** variable. |
| **db.get\_collections()** | [db.get\_collections()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-collections-get) returns a list of collections in the schema. Use the list to get references to collection objects, iterate over them, and so on. |

* Basic operations scoped by collections include:

| **Operation form** | **Description** |
| --- | --- |
| **db.*name*.add()** | The [add()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-add) method inserts one document or a list of documents into the named collection. |
| **db.*name*.find()** | The [find()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-find) method returns some or all documents in the named collection. |
| **db.*name*.modify()** | The [modify()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-modify) method updates documents in the named collection. |
| **db.*name*.remove()** | The [remove()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-documents-remove) method deletes one document or a list of documents from the named collection. |

#### Related Information

* See [Working with Collections](https://dev.mysql.com/doc/x-devapi-userguide/en/devapi-users-working-with-collections.html) for a general overview.
* [CRUD EBNF Definitions](https://dev.mysql.com/doc/x-devapi-userguide/en/mysql-x-crud-ebnf-definitions.html) provides a complete list of operations.

#### 20.4.3.1 Create, List, and Drop Collections

In MySQL Shell, you can create new collections, get a list of the existing collections in a schema, and remove an existing collection from a schema. Collection names are case-sensitive and each collection name must be unique.

##### Confirm the Schema

To show the value that is assigned to the schema variable, issue:

mysql-py> **db**

If the schema value is not **Schema:world\_x**, then set the **db** variable by issuing:

mysql-py> **\use world\_x**

##### Create a Collection

To create a new collection in an existing schema, use the **db** object's **createCollection()** method. The following example creates a collection called **flags** in the **world\_x** schema.

mysql-py> **db.create\_collection("flags")**

The method returns a collection object.

<Collection:flags>

##### List Collections

To display all collections in the **world\_x** schema, use the **db** object's **get\_collections()** method. Collections returned by the server you are currently connected to appear between brackets.

mysql-py> **db.get\_collections()**

[

<Collection:countryinfo>,

<Collection:flags>

]

##### Drop a Collection

To drop an existing collection from a schema, use the **db** object's **drop\_collection()** method. For example, to drop the **flags** collection from the current schema, issue:

mysql-py> **db.drop\_collection("flags")**

The **drop\_collection()** method is also used in MySQL Shell to drop a relational table from a schema.

##### Related Information

* See [Collection Objects](https://dev.mysql.com/doc/x-devapi-userguide/en/collection-objects.html) for more examples.

#### 20.4.3.2 Working with Collections

To work with the collections in a schema, use the **db** global object to access the current schema. In this example we are using the **world\_x** schema imported previously, and the **countryinfo** collection. Therefore, the format of the operations you issue is **db.*collection\_name*.operation**, where ***collection\_name*** is the name of the collection which the operation is executed against. In the following examples, the operations are executed against the **countryinfo** collection.

##### Add a Document

Use the **add()** method to insert one document or a list of documents into an existing collection. Insert the following document into the **countryinfo** collection. As this is multi-line content, press **Enter** twice to insert the document.

mysql-py> **db.countryinfo.add(**

**{**

**"GNP": .6,**

**"IndepYear": 1967,**

**"Name": "Sealand",**

**"Code:": "SEA",**

**"demographics": {**

**"LifeExpectancy": 79,**

**"Population": 27**

**},**

**"geography": {**

**"Continent": "Europe",**

**"Region": "British Islands",**

**"SurfaceArea": 193**

**},**

**"government": {**

**"GovernmentForm": "Monarchy",**

**"HeadOfState": "Michael Bates"**

**}**

**}**

**)**

The method returns the status of the operation. You can verify the operation by searching for the document. For example:

mysql-py> **db.countryinfo.find("Name = 'Sealand'")**

{

"GNP": 0.6,

"\_id": "00005e2ff4af00000000000000f4",

"Name": "Sealand",

"Code:": "SEA",

"IndepYear": 1967,

"geography": {

"Region": "British Islands",

"Continent": "Europe",

"SurfaceArea": 193

},

"government": {

"HeadOfState": "Michael Bates",

"GovernmentForm": "Monarchy"

},

"demographics": {

"Population": 27,

"LifeExpectancy": 79

}

}

Note that in addition to the fields specified when the document was added, there is one more field, the **\_id**. Each document requires an identifier field called **\_id**. The value of the **\_id** field must be unique among all documents in the same collection. In MySQL 8.0.11 and higher, document IDs are generated by the server, not the client, so MySQL Shell does not automatically set an **\_id** value. A MySQL server at 8.0.11 or higher sets an **\_id** value if the document does not contain the **\_id** field. A MySQL server at an earlier 8.0 release or at 5.7 does not set an **\_id** value in this situation, so you must specify it explicitly. If you do not, MySQL Shell returns error 5115 Document is missing a required field. For more information see [Understanding Document IDs](https://dev.mysql.com/doc/x-devapi-userguide/en/understanding-automatic-document-ids.html).

##### Related Information

* See [CollectionAddFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-collection-crud-functions.html" \l "crud-ebnf-collectionaddfunction" \t "_top) for the full syntax definition.
* See [Understanding Document IDs](https://dev.mysql.com/doc/x-devapi-userguide/en/understanding-automatic-document-ids.html).

#### 20.4.3.3 Find Documents

You can use the **find()** method to query for and return documents from a collection in a schema. MySQL Shell provides additional methods to use with the **find()** method to filter and sort the returned documents.

MySQL provides the following operators to specify search conditions: **OR** (**||**), **AND** (**&&**), **XOR**, **IS**, **NOT**, **BETWEEN**, **IN**, **LIKE**, **!=**, **<>**, **>**, **>=**, **<**, **<=**, **&**, **|**, **<<**, **>>**, **+**, **-**, **\***, **/**, **~**, and **%**.

##### Find All Documents in a Collection

To return all documents in a collection, use the **find()** method without specifying search conditions. For example, the following operation returns all documents in the **countryinfo** collection.

mysql-py> **db.countryinfo.find()**

[

{

"GNP": 828,

"Code:": "ABW",

"Name": "Aruba",

"IndepYear": null,

"geography": {

"Continent": "North America",

"Region": "Caribbean",

"SurfaceArea": 193

},

"government": {

"GovernmentForm": "Nonmetropolitan Territory of The Netherlands",

"HeadOfState": "Beatrix"

}

"demographics": {

"LifeExpectancy": 78.4000015258789,

"Population": 103000

},

...

}

]

240 documents in set (0.00 sec)

The method produces results that contain operational information in addition to all documents in the collection.

An empty set (no matching documents) returns the following information:

Empty set (0.00 sec)

##### Filter Searches

You can include search conditions with the **find()** method. The syntax for expressions that form a search condition is the same as that of traditional MySQL [Chapter 12, *Functions and Operators*](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html). You must enclose all expressions in quotes. For the sake of brevity, some of the examples do not display output.

A simple search condition could consist of the **Name** field and a value we know is in a document. The following example returns a single document:

mysql-py> **db.countryinfo.find("Name = 'Australia'")**

[

{

"GNP": 351182,

"Code:": "AUS",

"Name": "Australia",

"IndepYear": 1901,

"geography": {

"Continent": "Oceania",

"Region": "Australia and New Zealand",

"SurfaceArea": 7741220

},

"government": {

"GovernmentForm": "Constitutional Monarchy, Federation",

"HeadOfState": "Elisabeth II"

}

"demographics": {

"LifeExpectancy": 79.80000305175781,

"Population": 18886000

},

}

]

The following example searches for all countries that have a GNP higher than $500 billion. The **countryinfo** collection measures GNP in units of million.

mysql-py> **db.countryinfo.find("GNP > 500000")**

...[output removed]

10 documents in set (0.00 sec)

The Population field in the following query is embedded within the demographics object. To access the embedded field, use a period between demographics and Population to identify the relationship. Document and field names are case-sensitive.

mysql-py> **db.countryinfo.find("GNP > 500000 and demographics.Population < 100000000")**

...[output removed]

6 documents in set (0.00 sec)

Arithmetic operators in the following expression are used to query for countries with a GNP per capita higher than $30000. Search conditions can include arithmetic operators and most MySQL functions.

**Note**

Seven documents in the **countryinfo** collection have a population value of zero. Therefore warning messages appear at the end of the output.

mysql-py> **db.countryinfo.find("GNP\*1000000/demographics.Population > 30000")**

...[output removed]

9 documents in set, 7 warnings (0.00 sec)

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

Warning (Code 1365): Division by 0

You can separate a value from the search condition by using the **bind()** method. For example, instead of specifying a hard-coded country name as the condition, substitute a named placeholder consisting of a colon followed by a name that begins with a letter, such as country. Then use the **bind(*placeholder*, *value*)** method as follows:

mysql-py> **db.countryinfo.find("Name = :country").bind("country", "Italy")**

{

"GNP": 1161755,

"\_id": "00005de917d8000000000000006a",

"Code": "ITA",

"Name": "Italy",

"Airports": [],

"IndepYear": 1861,

"geography": {

"Region": "Southern Europe",

"Continent": "Europe",

"SurfaceArea": 301316

},

"government": {

"HeadOfState": "Carlo Azeglio Ciampi",

"GovernmentForm": "Republic"

},

"demographics": {

"Population": 57680000,

"LifeExpectancy": 79

}

}

1 document in set (0.01 sec)

Tip

Within a program, binding enables you to specify placeholders in your expressions, which are filled in with values before execution and can benefit from automatic escaping, as appropriate.

Always use binding to sanitize input. Avoid introducing values in queries using string concatenation, which can produce invalid input and, in some cases, can cause security issues.

You can use placeholders and the **bind()** method to create saved searches which you can then call with different values. For example to create a saved search for a country:

mysql-py> **myFind = db.countryinfo.find("Name = :country")**

mysql-py> **myFind.bind('country', 'France')**

{

"GNP": 1424285,

"\_id": "00005de917d80000000000000048",

"Code": "FRA",

"Name": "France",

"IndepYear": 843,

"geography": {

"Region": "Western Europe",

"Continent": "Europe",

"SurfaceArea": 551500

},

"government": {

"HeadOfState": "Jacques Chirac",

"GovernmentForm": "Republic"

},

"demographics": {

"Population": 59225700,

"LifeExpectancy": 78.80000305175781

}

}

1 document in set (0.0028 sec)

mysql-py> **myFind.bind('country', 'Germany')**

{

"GNP": 2133367,

"\_id": "00005de917d80000000000000038",

"Code": "DEU",

"Name": "Germany",

"IndepYear": 1955,

"geography": {

"Region": "Western Europe",

"Continent": "Europe",

"SurfaceArea": 357022

},

"government": {

"HeadOfState": "Johannes Rau",

"GovernmentForm": "Federal Republic"

},

"demographics": {

"Population": 82164700,

"LifeExpectancy": 77.4000015258789

}

}

1 document in set (0.0026 sec)

##### Project Results

You can return specific fields of a document, instead of returning all the fields. The following example returns the GNP and Name fields of all documents in the **countryinfo** collection matching the search conditions.

Use the **fields()** method to pass the list of fields to return.

mysql-py> **db.countryinfo.find("GNP > 5000000").fields(["GNP", "Name"])**

[

{

"GNP": 8510700,

"Name": "United States"

}

]

1 document in set (0.00 sec)

In addition, you can alter the returned documents—adding, renaming, nesting and even computing new field values—with an expression that describes the document to return. For example, alter the names of the fields with the following expression to return only two documents.

mysql-py> **db.countryinfo.find().fields(**

**mysqlx.expr('{"Name": upper(Name), "GNPPerCapita": GNP\*1000000/demographics.Population}')).limit(2)**

{

"Name": "ARUBA",

"GNPPerCapita": 8038.834951456311

}

{

"Name": "AFGHANISTAN",

"GNPPerCapita": 263.0281690140845

}

##### Limit, Sort, and Skip Results

You can apply the **limit()**, **sort()**, and **skip()** methods to manage the number and order of documents returned by the **find()** method.

To specify the number of documents included in a result set, append the **limit()** method with a value to the **find()** method. The following query returns the first five documents in the **countryinfo** collection.

mysql-py> **db.countryinfo.find().limit(5)**

... [output removed]

5 documents in set (0.00 sec)

To specify an order for the results, append the **sort()** method to the **find()** method. Pass to the **sort()** method a list of one or more fields to sort by and, optionally, the descending (**desc**) or ascending (**asc**) attribute as appropriate. Ascending order is the default order type.

For example, the following query sorts all documents by the IndepYear field and then returns the first eight documents in descending order.

mysql-py> **db.countryinfo.find().sort(["IndepYear desc"]).limit(8)**

... [output removed]

8 documents in set (0.00 sec)

By default, the **limit()** method starts from the first document in the collection. You can use the **skip()** method to change the starting document. For example, to ignore the first document and return the next eight documents matching the condition, pass to the **skip()** method a value of 1.

mysql-py> **db.countryinfo.find().sort(["IndepYear desc"]).limit(8).skip(1)**

... [output removed]

8 documents in set (0.00 sec)

##### Related Information

* The [MySQL Reference Manual](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html) provides detailed documentation on functions and operators.
* See [CollectionFindFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-collection-crud-functions.html" \l "crud-ebnf-collectionfindfunction" \t "_top) for the full syntax definition.

#### 20.4.3.4 Modify Documents

You can use the **modify()** method to update one or more documents in a collection. The X DevAPI provides additional methods for use with the **modify()** method to:

* Set and unset fields within documents.
* Append, insert, and delete arrays.
* Bind, limit, and sort the documents to be modified.

##### Set and Unset Document Fields

The **modify()** method works by filtering a collection to include only the documents to be modified and then applying the operations that you specify to those documents.

In the following example, the **modify()** method uses the search condition to identify the document to change and then the **set()** method replaces two values within the nested demographics object.

mysql-py> **db.countryinfo.modify("Code = 'SEA'").set(**

**"demographics", {"LifeExpectancy": 78, "Population": 28})**

After you modify a document, use the **find()** method to verify the change.

To remove content from a document, use the **modify()** and **unset()** methods. For example, the following query removes the GNP from a document that matches the search condition.

mysql-py> **db.countryinfo.modify("Name = 'Sealand'").unset("GNP")**

Use the **find()** method to verify the change.

mysql-py> **db.countryinfo.find("Name = 'Sealand'")**

{

"\_id": "00005e2ff4af00000000000000f4",

"Name": "Sealand",

"Code:": "SEA",

"IndepYear": 1967,

"geography": {

"Region": "British Islands",

"Continent": "Europe",

"SurfaceArea": 193

},

"government": {

"HeadOfState": "Michael Bates",

"GovernmentForm": "Monarchy"

},

"demographics": {

"Population": 27,

"LifeExpectancy": 79

}

}

##### Append, Insert, and Delete Arrays

To append an element to an array field, or insert, or delete elements in an array, use the **array\_append()**, **array\_insert()**, or **array\_delete()** methods. The following examples modify the **countryinfo** collection to enable tracking of international airports.

The first example uses the **modify()** and **set()** methods to create a new Airports field in all documents.

**Caution**

Use care when you modify documents without specifying a search condition; doing so modifies all documents in the collection.

mysql-py> **db.countryinfo.modify("true").set("Airports", [])**

With the Airports field added, the next example uses the **array\_append()** method to add a new airport to one of the documents. $.Airports in the following example represents the Airports field of the current document.

mysql-py> **db.countryinfo.modify("Name = 'France'").array\_append("$.Airports", "ORY")**

Use **find()** to see the change.

mysql-py> **db.countryinfo.find("Name = 'France'")**

{

"GNP": 1424285,

"\_id": "00005de917d80000000000000048",

"Code": "FRA",

"Name": "France",

"Airports": [

"ORY"

],

"IndepYear": 843,

"geography": {

"Region": "Western Europe",

"Continent": "Europe",

"SurfaceArea": 551500

},

"government": {

"HeadOfState": "Jacques Chirac",

"GovernmentForm": "Republic"

},

"demographics": {

"Population": 59225700,

"LifeExpectancy": 78.80000305175781

}

}

To insert an element at a different position in the array, use the **array\_insert()** method to specify which index to insert in the path expression. In this case, the index is 0, or the first element in the array.

mysql-py> **db.countryinfo.modify("Name = 'France'").array\_insert("$.Airports[0]", "CDG")**

To delete an element from the array, you must pass to the **array\_delete()** method the index of the element to be deleted.

mysql-py> **db.countryinfo.modify("Name = 'France'").array\_delete("$.Airports[1]")**

##### Related Information

* The [MySQL Reference Manual](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\data-types.html#json-paths) provides instructions to help you search for and modify JSON values.
* See [CollectionModifyFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-collection-crud-functions.html" \l "crud-ebnf-collectionmodifyfunction" \t "_top) for the full syntax definition.

#### 20.4.3.5 Remove Documents

You can use the **remove()** method to delete some or all documents from a collection in a schema. The X DevAPI provides additional methods for use with the **remove()** method to filter and sort the documents to be removed.

##### Remove Documents Using Conditions

The following example passes a search condition to the **remove()** method. All documents matching the condition are removed from the **countryinfo** collection. In this example, one document matches the condition.

mysql-py> **db.countryinfo.remove("Code = 'SEA'")**

##### Remove the First Document

To remove the first document in the **countryinfo** collection, use the **limit()** method with a value of 1.

mysql-py> **db.countryinfo.remove("true").limit(1)**

##### Remove the Last Document in an Order

The following example removes the last document in the **countryinfo** collection by country name.

mysql-py> **db.countryinfo.remove("true").sort(["Name desc"]).limit(1)**

##### Remove All Documents in a Collection

You can remove all documents in a collection. To do so, use the **remove("true")** method without specifying a search condition.

**Caution**

Use care when you remove documents without specifying a search condition. This action deletes all documents from the collection.

Alternatively, use the **db.drop\_collection('countryinfo')** operation to delete the **countryinfo** collection.

##### Related Information

* See [CollectionRemoveFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-collection-crud-functions.html" \l "crud-ebnf-collectionremovefunction" \t "_top) for the full syntax definition.
* See [Section 20.4.2, “Download and Import world\_x Database”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-download) for instructions to recreate the **world\_x** schema.

#### 20.4.3.6 Create and Drop Indexes

Indexes are used to find documents with specific field values quickly. Without an index, MySQL must begin with the first document and then read through the entire collection to find the relevant fields. The larger the collection, the more this costs. If a collection is large and queries on a specific field are common, then consider creating an index on a specific field inside a document.

For example, the following query performs better with an index on the Population field:

mysql-py> **db.countryinfo.find("demographics.Population < 100")**

...[output removed]

8 documents in set (0.00 sec)

The **create\_index()** method creates an index that you can define with a JSON document that specifies which fields to use. This section is a high level overview of indexing. For more information see [Indexing Collections](https://dev.mysql.com/doc/x-devapi-userguide/en/collection-indexing.html).

##### Add a Nonunique Index

To create a nonunique index, pass an index name and the index information to the **create\_index()** method. Duplicate index names are prohibited.

The following example specifies an index named **popul**, defined against the **Population** field from the **demographics** object, indexed as an **Integer** numeric value. The final parameter indicates whether the field should require the **NOT NULL** constraint. If the value is **false**, the field can contain **NULL** values. The index information is a JSON document with details of one or more fields to include in the index. Each field definition must include the full document path to the field, and specify the type of the field.

mysql-py> **db.countryinfo.createIndex("popul", {fields:**

**[{field: '$.demographics.Population', type: 'INTEGER'}]})**

Here, the index is created using an integer numeric value. Further options are available, including options for use with GeoJSON data. You can also specify the type of index, which has been omitted here because the default type “index” is appropriate.

##### Add a Unique Index

To create a unique index, pass an index name, the index definition, and the index type “unique” to the **create\_index()** method. This example shows a unique index created on the country name (**"Name"**), which is another common field in the **countryinfo** collection to index. In the index field description, **"TEXT(40)"** represents the number of characters to index, and **"required": True** specifies that the field is required to exist in the document.

mysql-py> **db.countryinfo.create\_index("name",**

**{"fields": [{"field": "$.Name", "type": "TEXT(40)", "required": True}], "unique": True})**

##### Drop an Index

To drop an index, pass the name of the index to drop to the **drop\_index()** method. For example, you can drop the “popul” index as follows:

mysql-py> **db.countryinfo.drop\_index("popul")**

##### Related Information

* See [Indexing Collections](https://dev.mysql.com/doc/x-devapi-userguide/en/collection-indexing.html) for more information.
* See [Defining an Index](https://dev.mysql.com/doc/x-devapi-userguide/en/collection-indexing.html#collection-index-definitions) for more information on the JSON document that defines an index.
* See [Collection Index Management Functions](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-collection-index-management-functions.html) for the full syntax definition.

### 20.4.4 Relational Tables

[20.4.4.1 Insert Records into Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-table-insert)

[20.4.4.2 Select Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-table-select)

[20.4.4.3 Update Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-table-update)

[20.4.4.4 Delete Tables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-table-delete)

You can also use X DevAPI to work with relational tables. In MySQL, each relational table is associated with a particular storage engine. The examples in this section use **[InnoDB](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\innodb-storage-engine.html" \o "Chapter 15 The InnoDB Storage Engine)** tables in the **world\_x** schema.

#### Confirm the Schema

To show the schema that is assigned to the **db** global variable, issue **db**.

mysql-py> **db**

<Schema:world\_x>

If the returned value is not **Schema:world\_x**, set the **db** variable as follows:

mysql-py> **\use world\_x**

Schema `world\_x` accessible through db.

#### Show All Tables

To display all relational tables in the **world\_x** schema, use the **get\_tables()** method on the **db** object.

mysql-py> **db.get\_tables()**

[

<Table:city>,

<Table:country>,

<Table:countrylanguage>

]

#### Basic Table Operations

Basic operations scoped by tables include:

| **Operation form** | **Description** |
| --- | --- |
| **db.*name*.insert()** | The [insert()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-table-insert) method inserts one or more records into the named table. |
| **db.*name*.select()** | The [select()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-table-select) method returns some or all records in the named table. |
| **db.*name*.update()** | The [update()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-table-update) method updates records in the named table. |
| **db.*name*.delete()** | The [delete()](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-table-delete) method deletes one or more records from the named table. |

#### Related Information

* See [Working with Relational Tables](https://dev.mysql.com/doc/x-devapi-userguide/en/devapi-users-working-with-relational-tables.html) for more information.
* [CRUD EBNF Definitions](https://dev.mysql.com/doc/x-devapi-userguide/en/mysql-x-crud-ebnf-definitions.html) provides a complete list of operations.
* See [Section 20.4.2, “Download and Import world\_x Database”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-download) for instructions on setting up the **world\_x** schema sample.

#### 20.4.4.1 Insert Records into Tables

You can use the **insert()** method with the **values()** method to insert records into an existing relational table. The **insert()** method accepts individual columns or all columns in the table. Use one or more **values()** methods to specify the values to be inserted.

##### Insert a Complete Record

To insert a complete record, pass to the **insert()** method all columns in the table. Then pass to the **values()** method one value for each column. For example, to add a new record to the city table in the **world\_x** database, insert the following record and press **Enter** twice.

mysql-py> **db.city.insert("ID", "Name", "CountryCode", "District", "Info").values(**

**None, "Olympia", "USA", "Washington", '{"Population": 5000}')**

The city table has five columns: ID, Name, CountryCode, District, and Info. Each value must match the data type of the column it represents.

##### Insert a Partial Record

The following example inserts values into the ID, Name, and CountryCode columns of the city table.

mysql-py> **db.city.insert("ID", "Name", "CountryCode").values(**

**None, "Little Falls", "USA").values(None, "Happy Valley", "USA")**

When you specify columns using the **insert()** method, the number of values must match the number of columns. In the previous example, you must supply three values to match the three columns specified.

##### Related Information

* See [TableInsertFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-table-crud-functions.html" \l "crud-ebnf-tableinsertfunction" \t "_top) for the full syntax definition.

#### 20.4.4.2 Select Tables

You can use the **select()** method to query for and return records from a table in a database. The X DevAPI provides additional methods to use with the **select()** method to filter and sort the returned records.

MySQL provides the following operators to specify search conditions: **OR** (**||**), **AND** (**&&**), **XOR**, **IS**, **NOT**, **BETWEEN**, **IN**, **LIKE**, **!=**, **<>**, **>**, **>=**, **<**, **<=**, **&**, **|**, **<<**, **>>**, **+**, **-**, **\***, **/**, **~**, and **%**.

##### Select All Records

To issue a query that returns all records from an existing table, use the **select()** method without specifying search conditions. The following example selects all records from the city table in the **world\_x** database.

**Note**

Limit the use of the empty **select()** method to interactive statements. Always use explicit column-name selections in your application code.

mysql-py> **db.city.select()**

+------+------------+-------------+------------+-------------------------+

| ID | Name | CountryCode | District | Info |

+------+------------+-------------+------------+-------------------------+

| 1 | Kabul | AFG | Kabol |{"Population": 1780000} |

| 2 | Qandahar | AFG | Qandahar |{"Population": 237500} |

| 3 | Herat | AFG | Herat |{"Population": 186800} |

... ... ... ... ...

| 4079 | Rafah | PSE | Rafah |{"Population": 92020} |

+------+------- ----+-------------+------------+-------------------------+

4082 rows in set (0.01 sec)

An empty set (no matching records) returns the following information:

Empty set (0.00 sec)

##### Filter Searches

To issue a query that returns a set of table columns, use the **select()** method and specify the columns to return between square brackets. This query returns the Name and CountryCode columns from the city table.

mysql-py> **db.city.select(["Name", "CountryCode"])**

+-------------------+-------------+

| Name | CountryCode |

+-------------------+-------------+

| Kabul | AFG |

| Qandahar | AFG |

| Herat | AFG |

| Mazar-e-Sharif | AFG |

| Amsterdam | NLD |

... ...

| Rafah | PSE |

| Olympia | USA |

| Little Falls | USA |

| Happy Valley | USA |

+-------------------+-------------+

4082 rows in set (0.00 sec)

To issue a query that returns rows matching specific search conditions, use the **where()** method to include those conditions. For example, the following example returns the names and country codes of the cities that start with the letter Z.

mysql-py> **db.city.select(["Name", "CountryCode"]).where("Name like 'Z%'")**

+-------------------+-------------+

| Name | CountryCode |

+-------------------+-------------+

| Zaanstad | NLD |

| Zoetermeer | NLD |

| Zwolle | NLD |

| Zenica | BIH |

| Zagazig | EGY |

| Zaragoza | ESP |

| Zamboanga | PHL |

| Zahedan | IRN |

| Zanjan | IRN |

| Zabol | IRN |

| Zama | JPN |

| Zhezqazghan | KAZ |

| Zhengzhou | CHN |

... ...

| Zeleznogorsk | RUS |

+-------------------+-------------+

59 rows in set (0.00 sec)

You can separate a value from the search condition by using the **bind()** method. For example, instead of using "Name = 'Z%' " as the condition, substitute a named placeholder consisting of a colon followed by a name that begins with a letter, such as name. Then include the placeholder and value in the **bind()** method as follows:

mysql-py> **db.city.select(["Name", "CountryCode"]).where(**

**"Name like :name").bind("name", "Z%")**

Tip

Within a program, binding enables you to specify placeholders in your expressions, which are filled in with values before execution and can benefit from automatic escaping, as appropriate.

Always use binding to sanitize input. Avoid introducing values in queries using string concatenation, which can produce invalid input and, in some cases, can cause security issues.

##### Project Results

To issue a query using the [**AND**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html#operator_and) operator, add the operator between search conditions in the **where()** method.

mysql-py> **db.city.select(["Name", "CountryCode"]).where(**

**"Name like 'Z%' and CountryCode = 'CHN'")**

+----------------+-------------+

| Name | CountryCode |

+----------------+-------------+

| Zhengzhou | CHN |

| Zibo | CHN |

| Zhangjiakou | CHN |

| Zhuzhou | CHN |

| Zhangjiang | CHN |

| Zigong | CHN |

| Zaozhuang | CHN |

... ...

| Zhangjiagang | CHN |

+----------------+-------------+

22 rows in set (0.01 sec)

To specify multiple conditional operators, you can enclose the search conditions in parenthesis to change the operator precedence. The following example demonstrates the placement of [**AND**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html#operator_and) and [**OR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html#operator_or) operators.

mysql-py> **db.city.select(["Name", "CountryCode"]).where(**

**"Name like 'Z%' and (CountryCode = 'CHN' or CountryCode = 'RUS')")**

+-------------------+-------------+

| Name | CountryCode |

+-------------------+-------------+

| Zhengzhou | CHN |

| Zibo | CHN |

| Zhangjiakou | CHN |

| Zhuzhou | CHN |

... ...

| Zeleznogorsk | RUS |

+-------------------+-------------+

29 rows in set (0.01 sec)

##### Limit, Order, and Offset Results

You can apply the **limit()**, **order\_by()**, and **offset()** methods to manage the number and order of records returned by the **select()** method.

To specify the number of records included in a result set, append the **limit()** method with a value to the **select()** method. For example, the following query returns the first five records in the country table.

mysql-py> **db.country.select(["Code", "Name"]).limit(5)**

+------+-------------+

| Code | Name |

+------+-------------+

| ABW | Aruba |

| AFG | Afghanistan |

| AGO | Angola |

| AIA | Anguilla |

| ALB | Albania |

+------+-------------+

5 rows in set (0.00 sec)

To specify an order for the results, append the **order\_by()** method to the **select()** method. Pass to the **order\_by()** method a list of one or more columns to sort by and, optionally, the descending (**desc**) or ascending (**asc**) attribute as appropriate. Ascending order is the default order type.

For example, the following query sorts all records by the Name column and then returns the first three records in descending order .

mysql-py> **db.country.select(["Code", "Name"]).order\_by(["Name desc"]).limit(3)**

+------+------------+

| Code | Name |

+------+------------+

| ZWE | Zimbabwe |

| ZMB | Zambia |

| YUG | Yugoslavia |

+------+------------+

3 rows in set (0.00 sec)

By default, the **limit()** method starts from the first record in the table. You can use the **offset()** method to change the starting record. For example, to ignore the first record and return the next three records matching the condition, pass to the **offset()** method a value of 1.

mysql-py> **db.country.select(["Code", "Name"]).order\_by(["Name desc"]).limit(3).offset(1)**

+------+------------+

| Code | Name |

+------+------------+

| ZMB | Zambia |

| YUG | Yugoslavia |

| YEM | Yemen |

+------+------------+

3 rows in set (0.00 sec)

##### Related Information

* The [MySQL Reference Manual](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\functions.html) provides detailed documentation on functions and operators.
* See [TableSelectFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-table-crud-functions.html" \l "crud-ebnf-tableselectfunction" \t "_top) for the full syntax definition.

#### 20.4.4.3 Update Tables

You can use the **update()** method to modify one or more records in a table. The **update()** method works by filtering a query to include only the records to be updated and then applying the operations you specify to those records.

To replace a city name in the city table, pass to the **set()** method the new city name. Then, pass to the **where()** method the city name to locate and replace. The following example replaces the city Peking with Beijing.

mysql-py> **db.city.update().set("Name", "Beijing").where("Name = 'Peking'")**

Use the **select()** method to verify the change.

mysql-py> **db.city.select(["ID", "Name", "CountryCode", "District", "Info"]).where("Name = 'Beijing'")**

+------+-----------+-------------+----------+-----------------------------+

| ID | Name | CountryCode | District | Info |

+------+-----------+-------------+----------+-----------------------------+

| 1891 | Beijing | CHN | Peking | {"Population": 7472000} |

+------+-----------+-------------+----------+-----------------------------+

1 row in set (0.00 sec)

##### Related Information

* See [TableUpdateFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-table-crud-functions.html" \l "crud-ebnf-tableupdatefunction" \t "_top) for the full syntax definition.

#### 20.4.4.4 Delete Tables

You can use the **delete()** method to remove some or all records from a table in a database. The X DevAPI provides additional methods to use with the **delete()** method to filter and order the records to be deleted.

##### Delete Records Using Conditions

The example that follows passes search conditions to the **delete()** method. All records matching the condition are deleted from the **city** table. In this example, one record matches the condition.

mysql-py> **db.city.delete().where("Name = 'Olympia'")**

##### Delete the First Record

To delete the first record in the city table, use the **limit()** method with a value of 1.

mysql-py> **db.city.delete().limit(1)**

##### Delete All Records in a Table

You can delete all records in a table. To do so, use the **delete()** method without specifying a search condition.

**Caution**

Use care when you delete records without specifying a search condition; doing so deletes all records from the table.

##### Drop a Table

The **drop\_collection()** method is also used in MySQL Shell to drop a relational table from a database. For example, to drop the **citytest** table from the **world\_x** database, issue:

mysql-py> **db.drop\_collection("citytest")**

##### Related Information

* See [TableDeleteFunction](https://dev.mysql.com/doc/x-devapi-userguide/en/crud-ebnf-table-crud-functions.html" \l "crud-ebnf-tabledeletefunction" \t "_top) for the full syntax definition.
* See [Section 20.4.2, “Download and Import world\_x Database”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#mysql-shell-tutorial-python-download) for instructions to recreate the **world\_x** database.

### 20.4.5 Documents in Tables

In MySQL, a table may contain traditional relational data, JSON values, or both. You can combine traditional data with JSON documents by storing the documents in columns having a native [**JSON**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\data-types.html#json) data type.

Examples in this section use the city table in the **world\_x** schema.

#### city Table Description

The city table has five columns (or fields).

+---------------+------------+-------+-------+---------+------------------+

| Field | Type | Null | Key | Default | Extra |

+---------------+------------+-------+-------+---------+------------------+

| ID | int(11) | NO | PRI | null | auto\_increment |

| Name | char(35) | NO | | | |

| CountryCode | char(3) | NO | | | |

| District | char(20) | NO | | | |

| Info | json | YES | | null | |

+---------------+------------+-------+-------+---------+------------------+

#### Insert a Record

To insert a document into the column of a table, pass to the **values()** method a well-formed JSON document in the correct order. In the following example, a document is passed as the final value to be inserted into the Info column.

mysql-py> **db.city.insert().values(**

**None, "San Francisco", "USA", "California", '{"Population":830000}')**

#### Select a Record

You can issue a query with a search condition that evaluates document values in the expression.

mysql-py> **db.city.select(["ID", "Name", "CountryCode", "District", "Info"]).where(**

**"CountryCode = :country and Info->'$.Population' > 1000000").bind(**

**'country', 'USA')**

+------+----------------+-------------+----------------+-----------------------------+

| ID | Name | CountryCode | District | Info |

+------+----------------+-------------+----------------+-----------------------------+

| 3793 | New York | USA | New York | {"Population": 8008278} |

| 3794 | Los Angeles | USA | California | {"Population": 3694820} |

| 3795 | Chicago | USA | Illinois | {"Population": 2896016} |

| 3796 | Houston | USA | Texas | {"Population": 1953631} |

| 3797 | Philadelphia | USA | Pennsylvania | {"Population": 1517550} |

| 3798 | Phoenix | USA | Arizona | {"Population": 1321045} |

| 3799 | San Diego | USA | California | {"Population": 1223400} |

| 3800 | Dallas | USA | Texas | {"Population": 1188580} |

| 3801 | San Antonio | USA | Texas | {"Population": 1144646} |

+------+----------------+-------------+----------------+-----------------------------+

9 rows in set (0.01 sec)

#### Related Information

* See [Working with Relational Tables and Documents](https://dev.mysql.com/doc/x-devapi-userguide/en/devapi-users-working-with-relational-tables-and-documents.html) for more information.
* See [Section 11.5, “The JSON Data Type”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\data-types.html#json) for a detailed description of the data type.

## 20.5 X Plugin

[20.5.1 Checking X Plugin Installation](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-checking-installation)

[20.5.2 Disabling X Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-disabling)

[20.5.3 Using Encrypted Connections with X Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-encrypted-connections)

[20.5.4 Using X Plugin with the Caching SHA-2 Authentication Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-sha2-cache-plugin)

[20.5.5 Connection Compression with X Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression)

[20.5.6 X Plugin Options and Variables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-options-variables)

[20.5.7 Monitoring X Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-system-monitoring)

This section explains how to use, configure and monitor X Plugin.

### 20.5.1 Checking X Plugin Installation

X Plugin is enabled by default in MySQL 8, therefore installing or upgrading to MySQL 8 makes the plugin available. You can verify X Plugin is installed on an instance of MySQL server by using the [**SHOW plugins**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\sql-statements.html#show-plugins) statement to view the plugins list.

To use MySQL Shell to verify X Plugin is installed, issue:

shell> mysqlsh -u ***user*** --sqlc -P 3306 -e "SHOW plugins"

To use MySQL Client to verify X Plugin is installed, issue:

shell> mysql -u ***user*** -p -e "SHOW plugins"

An example result if X Plugin is installed is highlighted here:

+----------------------------+----------+--------------------+---------+---------+

| Name | Status | Type | Library | License |

+----------------------------+----------+--------------------+---------+---------+

...

| mysqlx | ACTIVE | DAEMON | NULL | GPL |

...

+----------------------------+----------+--------------------+---------+---------+

### 20.5.2 Disabling X Plugin

The X Plugin can be disabled at startup by either setting [mysqlx=0](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "option_mysqld_mysqlx) in your MySQL configuration file, or by passing in either [--mysqlx=0](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#option_mysqld_mysqlx) or [--skip-mysqlx](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#option_mysqld_mysqlx) when starting the MySQL server.

Alternatively, use the [-DWITH\_MYSQLX=OFF](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_with_mysqlx) CMake option to compile MySQL Server without X Plugin.

### 20.5.3 Using Encrypted Connections with X Plugin

This section explains how to configure X Plugin to use encrypted connections. For more background information, see [Section 6.3, “Using Encrypted Connections”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\security.html#encrypted-connections).

To enable configuring support for encrypted connections, X Plugin has **mysqlx\_ssl\_*xxx*** system variables, which can have different values from the **ssl\_*xxx*** system variables used with MySQL Server. For example, X Plugin can have SSL key, certificate, and certificate authority files that differ from those used for MySQL Server. These variables are described at [Section 20.5.6.2, “X Plugin Options and System Variables”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-options-system-variables). Similarly, X Plugin has its own **Mysqlx\_ssl\_*xxx*** status variables that correspond to the MySQL Server encrypted-connection **Ssl\_*xxx*** status variables. See [Section 20.5.6.3, “X Plugin Status Variables”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-status-variables).

At initialization, X Plugin determines its TLS context for encrypted connections as follows:

* If all **mysqlx\_ssl\_*xxx*** system variables have their default values, X Plugin uses the same TLS context as the MySQL Server main connection interface, which is determined by the values of the **ssl\_*xxx*** system variables.
* If any **mysqlx\_ssl\_*xxx*** variable has a nondefault value, X Plugin uses the TLS context defined by the values of its own system variables. (This is the case if any **mysqlx\_ssl\_*xxx*** system variable is set to a value different from its default.)

This means that, on a server with X Plugin enabled, you can choose to have MySQL Protocol and X Protocol connections share the same encryption configuration by setting only the **ssl\_*xxx*** variables, or have separate encryption configurations for MySQL Protocol and X Protocol connections by configuring the **ssl\_*xxx*** and **mysqlx\_ssl\_*xxx*** variables separately.

To have MySQL Protocol and X Protocol connections use the same encryption configuration, set only the **ssl\_*xxx*** system variables in my.cnf:

[mysqld]

ssl\_ca=ca.pem

ssl\_cert=server-cert.pem

ssl\_key=server-key.pem

To configure encryption separately for MySQL Protocol and X Protocol connections, set both the **ssl\_*xxx*** and **mysqlx\_ssl\_*xxx*** system variables in my.cnf:

[mysqld]

ssl\_ca=ca1.pem

ssl\_cert=server-cert1.pem

ssl\_key=server-key1.pem

mysqlx\_ssl\_ca=ca2.pem

mysqlx\_ssl\_cert=server-cert2.pem

mysqlx\_ssl\_key=server-key2.pem

For general information about configuring connection-encryption support, see [Section 6.3.1, “Configuring MySQL to Use Encrypted Connections”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\security.html#using-encrypted-connections). That discussion is written for MySQL Server, but the parameter names are similar for X Plugin. (The X Plugin **mysqlx\_ssl\_*xxx*** system variable names correspond to the MySQL Server **ssl\_*xxx*** system variable names.)

The **[tls\_version](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_tls_version)** system variable that determines the permitted TLS versions for MySQL Protocol connections also applies to X Protocol connections. The permitted TLS versions for both types of connections are therefore the same.

Encryption per connection is optional, but a specific user can be required to use encryption for X Protocol and MySQL Protocol connections by including an appropriate **REQUIRE** clause in the [**CREATE USER**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\sql-statements.html#create-user) statement that creates the user. For details, see [Section 13.7.1.3, “CREATE USER Statement”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\sql-statements.html#create-user). Alternatively, to require all users to use encryption for X Protocol and MySQL Protocol connections, enable the **[require\_secure\_transport](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_require_secure_transport)** system variable. For additional information, see [Configuring Encrypted Connections as Mandatory](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\security.html#mandatory-encrypted-connections).

### 20.5.4 Using X Plugin with the Caching SHA-2 Authentication Plugin

X Plugin supports MySQL user accounts created with the **caching\_sha2\_password** authentication plugin. For more information on this plugin, see [Section 6.4.1.2, “Caching SHA-2 Pluggable Authentication”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\security.html#caching-sha2-pluggable-authentication). You can use X Plugin to authenticate against such accounts using non-SSL connections with **SHA256\_MEMORY** authentication and SSL connections with **PLAIN** authentication.

Although the **caching\_sha2\_password** authentication plugin holds an authentication cache, this cache is not shared with X Plugin, so X Plugin uses its own authentication cache for **SHA256\_MEMORY** authentication. The X Plugin authentication cache stores hashes of user account passwords, and cannot be accessed using SQL. If a user account is modified or removed, the relevant entries are removed from the cache. The X Plugin authentication cache is maintained by the **mysqlx\_cache\_cleaner** plugin, which is enabled by default, and has no related system variables or status variables.

Before you can use non-SSL X Protocol connections to authenticate an account that uses the **caching\_sha2\_password** authentication plugin, the account must have authenticated at least once over an X Protocol connection with SSL, to supply the password to the X Plugin authentication cache. Once this initial authentication over SSL has succeeded, non-SSL X Protocol connections can be used.

It is possible to disable the **mysqlx\_cache\_cleaner** plugin by starting the MySQL server with the option **--mysqlx\_cache\_cleaner=0**. If you do this, the X Plugin authentication cache is disabled, and therefore SSL must always be used for X Protocol connections when authenticating with **SHA256\_MEMORY** authentication.

### 20.5.5 Connection Compression with X Plugin

From MySQL 8.0.19, X Plugin supports compression of messages sent over X Protocol connections. Connections can be compressed if the server and the client agree on a mutually supported compression algorithm. Enabling compression reduces the number of bytes sent over the network, but adds to the server and client an additional CPU cost for compression and decompression operations. The benefits of compression therefore occur primarily when there is low network bandwidth, network transfer time dominates the cost of compression and decompression operations, and result sets are large.

**Note**

Different MySQL clients implement support for connection compression differently; consult your client documentation for details. For example, for classic MySQL protocol connections, see [Section 4.2.8, “Connection Compression Control”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#connection-compression-control).

* [Configuring Connection Compression for X Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression-configuration)
* [Compressed Connection Characteristics for X Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression-characteristics)
* [Monitoring Connection Compression for X Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression-monitoring)

#### Configuring Connection Compression for X Plugin

By default, X Plugin supports the zstd, LZ4, and Deflate compression algorithms. Compression with the Deflate algorithm is carried out using the zlib software library, so the **deflate\_stream** compression algorithm setting for X Protocol connections is equivalent to the **zlib** setting for classic MySQL protocol connections.

On the server side, you can disallow any of the compression algorithms by setting the **[mysqlx\_compression\_algorithms](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_compression_algorithms)** system variable to include only those permittted. The algorithm names **zstd\_stream**, **lz4\_message**, and **deflate\_stream** can be specified in any combination, and the order and lettercase are not important. If the system variable value is the empty string, no compression algorithms are permitted and connections are uncompressed.

The following table compares the characteristics of the different compression algorithms and shows their assigned priorities. By default, the server chooses the highest-priority algorithm permitted in common by the server and the client; clients may change the priorities as described later. The short form alias for the algorithms can be used by clients when specifying them.

**Table 20.1 X Protocol Compression Algorithm Characteristics**

| **Algorithm** | **Alias** | **Compression Ratio** | **Throughput** | **CPU Cost** | **Default Priority** |
| --- | --- | --- | --- | --- | --- |
| **zsth\_stream** | **zstd** | High | High | Medium | First |
| **lz4\_message** | **lz4** | Low | High | Lowest | Second |
| **deflate\_stream** | **deflate** | High | Low | Highest | Third |

The X Protocol set of permitted compression algorithms (whether user-specified or default) is independent of the set of compression algorithms permitted by MySQL Server for classic MySQL protocol connections, which is specified by the **[protocol\_compression\_algorithms](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_protocol_compression_algorithms)** server system variable. If you do not specify the **[mysqlx\_compression\_algorithms](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_compression_algorithms)** system variable, X Plugin does not fall back to using compression settings for classic MySQL protocol connections. Instead, its default is to permit all algorithms shown in [Table 20.1, “X Protocol Compression Algorithm Characteristics”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression-algorithms). This is unlike the situation for the TLS context, where MySQL Server settings are used if the X Plugin system variables are not set, as described in [Section 20.5.3, “Using Encrypted Connections with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-encrypted-connections). For information about compression for classic MySQL protocol connections, see [Section 4.2.8, “Connection Compression Control”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#connection-compression-control).

On the client side, an X Protocol connection request can specify several parameters for compression control:

* The compression mode.
* The compression level (from MySQL 8.0.20).
* The list of permitted compression algorithms in priority order (from MySQL 8.0.22).

**Note**

Some clients or Connectors might not support a given compression-control feature. For example, specifying compression level for X Protocol connections is supported only by MySQL Shell, not by other MySQL clients or Connectors. See the documentation for specific products for details about supported features and how to use them.

The connection mode has these permitted values:

* **disabled**: The connection is uncompressed.
* **preferred**: The server and client negotiate to find a compression algorithm they both permit. If no common algorithm is available, the connection is uncompressed. This is the default mode if not specified explicitly.
* **required**: Compression algorithm negotiation occurs as for **preferred** mode, but if no common algorithm is available, the connection request terminates with an error.

In addition to agreeing on a compression algorithm for each connection, the server and client can agree on a compression level from the numeric range that applies to the agreed algorithm. As the compression level for an algorithm increases, the data compression ratio increases, which reduces the network bandwidth and transfer time needed to send the message to the client. However, the effort required for data compression also increases, taking up time and CPU and memory resources on the server. Increases in the compression effort do not have a linear relationship to increases in the compression ratio.

In MySQL 8.0.19, X Plugin always uses the library default compression level for each algorithm (3 for zstd, 0 for LZ4, and 6 for Deflate), and the client cannot negotiate this. From MySQL 8.0.20, the client can request a specific compression level during capability negotiations with the server for an X Protocol connection.

The default compression levels used by X Plugin from MySQL 8.0.20 have been selected through performance testing as being a good trade-off between compression time and network transit time. These defaults are not necessarily the same as the library default for each algorithm. They apply if the client does not request a compression level for the algorithm. The default compression levels are initially set to 3 for zstd, 2 for LZ4, and 3 for Deflate. You can adjust these settings using the [**mysqlx\_zstd\_default\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_zstd_default_compression_level), [**mysqlx\_lz4\_default\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_lz4_default_compression_level), and **[mysqlx\_deflate\_default\_compression\_level](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_deflate_default_compression_level)** system variables.

To prevent excessive resource consumption on the server, X Plugin sets a maximum compression level that the server permits for each algorithm. If a client requests a compression level that exceeds this setting, the server uses its maximum permitted compression level (compression level requests by a client are supported only by MySQL Shell). The maximum compression levels are initially set to 11 for zstd, 8 for LZ4, and 5 for Deflate. You can adjust these settings using the [**mysqlx\_zstd\_max\_client\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_zstd_max_client_compression_level), [**mysqlx\_lz4\_max\_client\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_lz4_max_client_compression_level), and **[mysqlx\_deflate\_max\_client\_compression\_level](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_deflate_max_client_compression_level)** system variables.

If the server and client permit more than one algorithm in common, the default priority order for choosing an algorithm during negotiation is shown in [Table 20.1, “X Protocol Compression Algorithm Characteristics”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression-algorithms). From MySQL 8.0.22, for clients that support specifying compression algorithms, the connection request can include a list of algorithms permitted by the client, specified using the algorithm name or its alias. The order of these algorithms in the list is taken as a priority order by the server. The algorithm used in this case is the first of those in the client list that is also permitted on the server side. However, the option for compression algorithms is subject to the compression mode:

* If the compression mode is **disabled**, the compression algorithms option is ignored.
* If the compression mode is **preferred** but no algorithm permitted on the client side is permitted on the server side, the connection is uncompressed.
* If the compression mode is **required** but no algorithm permitted on the client side is permitted on the server side, an error occurs.

To monitor the effects of message compression, use the X Plugin status variables described in [Monitoring Connection Compression for X Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression-monitoring). You can use these status variables to calculate the benefit of message compression with your current settings, and use that information to tune your settings.

#### Compressed Connection Characteristics for X Plugin

X Protocol connection compression operates with the following behaviors and boundaries:

* The **\_stream** and **\_message** suffixes in algorithm names refer to two different operational modes: In stream mode, all X Protocol messages in a single connection are compressed into a continuous stream and must be decompressed in the same manner—following the order they were compressed and without skipping any messages. In message mode, each message is compressed individually and independently, and need not be decompressed in the order in which they were compressed. Also, message mode does not require all compressed messages to be decompressed.
* Compression is not applied to any messages that are sent before authentication succeeds.
* Compression is not applied to control flow messages such as **Mysqlx.Ok**, **Mysqlx.Error**, and **Mysqlx.Sql.StmtExecuteOk** messages.
* All other X Protocol messages can be compressed if the server and client agree on a mutually permitted compression algorithm during capability negotiation. If the client does not request compression at that stage, neither the client nor the server applies compression to messages.
* When messages sent over X Protocol connections are compressed, the limit specified by the **[mysqlx\_max\_allowed\_packet](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_max_allowed_packet)** system variable still applies. The network packet must be smaller than this limit after the message payload has been decompressed. If the limit is exceeded, X Plugin returns a decompression error and closes the connection.
* The following points pertain to compression level requests by clients, which is supported only by MySQL Shell:
  + Compression levels must be specified by the client as an integer. If any other type of value is supplied, the connection closes with an error.
  + If a client specifies an algorithm but not a compression level, the server uses its default compression level for the algorithm.
  + If a client requests an algorithm compression level that exceeds the server maximum permitted level, the server uses the maximum permitted level.
  + If a client requests an algorithm compression level that is less than the server minimum permitted level, the server uses the minimum permitted level.

#### Monitoring Connection Compression for X Plugin

You can monitor the effects of message compression using the X Plugin status variables. When message compression is in use, the session **[Mysqlx\_compression\_algorithm](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_compression_algorithm)** status variable shows which compression algorithm is in use for the current X Protocol connection, and **[Mysqlx\_compression\_level](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_compression_level)** shows the compression level that was selected. These session status variables are available from MySQL 8.0.20.

From MySQL 8.0.19, X Plugin status variables can be used to calculate the efficiency of the compression algorithms that are selected (the data compression ratio), and the overall effect of using message compression. Use the session value of the status variables in the following calculations to see what the benefit of message compression was for a specific session with a known compression algorithm. Or use the global value of the status variables to check the overall benefit of message compression for your server across all sessions using X Protocol connections, including all the compression algorithms that have been used for those sessions, and all sessions that did not use message compression. You can then tune message compression by adjusting the permitted compression algorithms, maximum compression level, and default compression level, as described in [Configuring Connection Compression for X Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression-configuration).

When message compression is in use, the **[Mysqlx\_bytes\_sent](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_sent)** status variable shows the total number of bytes sent out from the server, including compressed message payloads measured after compression, any items in compressed messages that were not compressed such as X Protocol headers, and any uncompressed messages. The **[Mysqlx\_bytes\_sent\_compressed\_payload](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_sent_compressed_payload)** status variable shows the total number of bytes sent as compressed message payloads, measured after compression, and the **[Mysqlx\_bytes\_sent\_uncompressed\_frame](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_sent_uncompressed_frame)** status variable shows the total number of bytes for those same message payloads but measured before compression. The compression ratio, which shows the efficiency of the compression algorithm, can therefore be calculated using the following expression:

mysqlx\_bytes\_sent\_uncompressed\_frame / mysqlx\_bytes\_sent\_compressed\_payload

The effectiveness of compression for X Protocol messages sent by the server can be calculated using the following expression:

(mysqlx\_bytes\_sent - mysqlx\_bytes\_sent\_compressed\_payload + mysqlx\_bytes\_sent\_uncompressed\_frame) / mysqlx\_bytes\_sent

For messages received by the server from clients, the **[Mysqlx\_bytes\_received\_compressed\_payload](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_received_compressed_payload)** status variable shows the total number of bytes received as compressed message payloads, measured before decompression, and the **[Mysqlx\_bytes\_received\_uncompressed\_frame](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_received_uncompressed_frame)** status variable shows the total number of bytes for those same message payloads but measured after decompression. The **[Mysqlx\_bytes\_received](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_received)** status variable includes compressed message payloads measured before decompression, any uncompressed items in compressed messages, and any uncompressed messages.

### 20.5.6 X Plugin Options and Variables

[20.5.6.1 X Plugin Option and Variable Reference](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-option-variable-reference)

[20.5.6.2 X Plugin Options and System Variables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-options-system-variables)

[20.5.6.3 X Plugin Status Variables](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-status-variables)

This section describes the command options and system variables that configure X Plugin, as well as the status variables available for monitoring purposes. If configuration values specified at startup time are incorrect, X Plugin could fail to initialize properly and the server does not load it. In this case, the server could also produce error messages for other X Plugin settings because it cannot recognize them.

#### 20.5.6.1 X Plugin Option and Variable Reference

This table provides an overview of the command options, system variables, and status variables provided by X Plugin.

**Table 20.2 X Plugin Option and Variable Reference**

| **Name** | **Cmd-Line** | **Option File** | **System Var** | **Status Var** | **Var Scope** | **Dynamic** |
| --- | --- | --- | --- | --- | --- | --- |
| [**mysqlx**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#option_mysqld_mysqlx) | Yes | Yes |  |  |  |  |
| [**Mysqlx\_aborted\_clients**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_aborted_clients) |  |  |  | Yes | Global | No |
| [**Mysqlx\_address**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_address) |  |  |  | Yes | Global | No |
| [**mysqlx\_bind\_address**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_bind_address) | Yes | Yes | Yes |  | Global | No |
| [**Mysqlx\_bytes\_received**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_bytes_received) |  |  |  | Yes | Both | No |
| [**Mysqlx\_bytes\_received\_compressed\_payload**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_bytes_received_compressed_payload) |  |  |  | Yes | Both | No |
| [**Mysqlx\_bytes\_received\_uncompressed\_frame**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_bytes_received_uncompressed_frame) |  |  |  | Yes | Both | No |
| [**Mysqlx\_bytes\_sent**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_bytes_sent) |  |  |  | Yes | Both | No |
| [**Mysqlx\_bytes\_sent\_compressed\_payload**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_bytes_sent_compressed_payload) |  |  |  | Yes | Both | No |
| [**Mysqlx\_bytes\_sent\_uncompressed\_frame**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_bytes_sent_uncompressed_frame) |  |  |  | Yes | Both | No |
| [**Mysqlx\_compression\_algorithm**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_compression_algorithm) |  |  |  | Yes | Session | No |
| [**mysqlx\_compression\_algorithms**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_compression_algorithms) | Yes | Yes | Yes |  | Global | Yes |
| [**Mysqlx\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_compression_level) |  |  |  | Yes | Session | No |
| [**mysqlx\_connect\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_connect_timeout) | Yes | Yes | Yes |  | Global | Yes |
| [**Mysqlx\_connection\_accept\_errors**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_connection_accept_errors) |  |  |  | Yes | Both | No |
| [**Mysqlx\_connection\_errors**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_connection_errors) |  |  |  | Yes | Both | No |
| [**Mysqlx\_connections\_accepted**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_connections_accepted) |  |  |  | Yes | Global | No |
| [**Mysqlx\_connections\_closed**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_connections_closed) |  |  |  | Yes | Global | No |
| [**Mysqlx\_connections\_rejected**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_connections_rejected) |  |  |  | Yes | Global | No |
| [**Mysqlx\_crud\_create\_view**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_crud_create_view) |  |  |  | Yes | Both | No |
| [**Mysqlx\_crud\_delete**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_crud_delete) |  |  |  | Yes | Both | No |
| [**Mysqlx\_crud\_drop\_view**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_crud_drop_view) |  |  |  | Yes | Both | No |
| [**Mysqlx\_crud\_find**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_crud_find) |  |  |  | Yes | Both | No |
| [**Mysqlx\_crud\_insert**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_crud_insert) |  |  |  | Yes | Both | No |
| [**Mysqlx\_crud\_modify\_view**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_crud_modify_view) |  |  |  | Yes | Both | No |
| [**Mysqlx\_crud\_update**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_crud_update) |  |  |  | Yes | Both | No |
| [**mysqlx\_deflate\_default\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_deflate_default_compression_level) | Yes | Yes | Yes |  | Global | Yes |
| [**mysqlx\_deflate\_max\_client\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_deflate_max_client_compression_level) | Yes | Yes | Yes |  | Global | Yes |
| [**mysqlx\_document\_id\_unique\_prefix**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_document_id_unique_prefix) | Yes | Yes | Yes |  | Global | Yes |
| [**mysqlx\_enable\_hello\_notice**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_enable_hello_notice) | Yes | Yes | Yes |  | Global | Yes |
| [**Mysqlx\_errors\_sent**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_errors_sent) |  |  |  | Yes | Both | No |
| **Mysqlx\_errors\_unknown\_message\_type** |  |  |  | Yes | Both | No |
| [**Mysqlx\_expect\_close**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_expect_close) |  |  |  | Yes | Both | No |
| [**Mysqlx\_expect\_open**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_expect_open) |  |  |  | Yes | Both | No |
| [**mysqlx\_idle\_worker\_thread\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_idle_worker_thread_timeout) | Yes | Yes | Yes |  | Global | Yes |
| [**Mysqlx\_init\_error**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_init_error) |  |  |  | Yes | Both | No |
| [**mysqlx\_interactive\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_interactive_timeout) | Yes | Yes | Yes |  | Global | Yes |
| [**mysqlx\_lz4\_default\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_lz4_default_compression_level) | Yes | Yes | Yes |  | Global | Yes |
| [**mysqlx\_lz4\_max\_client\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_lz4_max_client_compression_level) | Yes | Yes | Yes |  | Global | Yes |
| [**mysqlx\_max\_allowed\_packet**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_max_allowed_packet) | Yes | Yes | Yes |  | Global | Yes |
| [**mysqlx\_max\_connections**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_max_connections) | Yes | Yes | Yes |  | Global | Yes |
| [**Mysqlx\_messages\_sent**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_messages_sent) |  |  |  | Yes | Both | No |
| [**mysqlx\_min\_worker\_threads**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_min_worker_threads) | Yes | Yes | Yes |  | Global | Yes |
| [**Mysqlx\_notice\_global\_sent**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_notice_global_sent) |  |  |  | Yes | Both | No |
| [**Mysqlx\_notice\_other\_sent**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_notice_other_sent) |  |  |  | Yes | Both | No |
| [**Mysqlx\_notice\_warning\_sent**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_notice_warning_sent) |  |  |  | Yes | Both | No |
| [**Mysqlx\_notified\_by\_group\_replication**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_notified_by_group_replication) |  |  |  | Yes | Both | No |
| [**Mysqlx\_port**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_port) |  |  |  | Yes | Global | No |
| [**mysqlx\_port**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_port) | Yes | Yes | Yes |  | Global | No |
| [**mysqlx\_port\_open\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_port_open_timeout) | Yes | Yes | Yes |  | Global | No |
| [**mysqlx\_read\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_read_timeout) | Yes | Yes | Yes |  | Session | Yes |
| [**Mysqlx\_rows\_sent**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_rows_sent) |  |  |  | Yes | Both | No |
| [**Mysqlx\_sessions**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_sessions) |  |  |  | Yes | Global | No |
| [**Mysqlx\_sessions\_accepted**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_sessions_accepted) |  |  |  | Yes | Global | No |
| [**Mysqlx\_sessions\_closed**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_sessions_closed) |  |  |  | Yes | Global | No |
| [**Mysqlx\_sessions\_fatal\_error**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_sessions_fatal_error) |  |  |  | Yes | Global | No |
| [**Mysqlx\_sessions\_killed**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_sessions_killed) |  |  |  | Yes | Global | No |
| [**Mysqlx\_sessions\_rejected**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_sessions_rejected) |  |  |  | Yes | Global | No |
| [**Mysqlx\_socket**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_socket) |  |  |  | Yes | Global | No |
| [**mysqlx\_socket**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_socket) | Yes | Yes | Yes |  | Global | No |
| [**Mysqlx\_ssl\_accept\_renegotiates**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_accept_renegotiates) |  |  |  | Yes | Global | No |
| [**Mysqlx\_ssl\_accepts**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_accepts) |  |  |  | Yes | Global | No |
| [**Mysqlx\_ssl\_active**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_active) |  |  |  | Yes | Both | No |
| [**mysqlx\_ssl\_ca**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_ca) | Yes | Yes | Yes |  | Global | No |
| [**mysqlx\_ssl\_capath**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_capath) | Yes | Yes | Yes |  | Global | No |
| [**mysqlx\_ssl\_cert**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_cert) | Yes | Yes | Yes |  | Global | No |
| [**Mysqlx\_ssl\_cipher**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_cipher) |  |  |  | Yes | Both | No |
| [**mysqlx\_ssl\_cipher**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_cipher) | Yes | Yes | Yes |  | Global | No |
| [**Mysqlx\_ssl\_cipher\_list**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_cipher_list) |  |  |  | Yes | Both | No |
| [**mysqlx\_ssl\_crl**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_crl) | Yes | Yes | Yes |  | Global | No |
| [**mysqlx\_ssl\_crlpath**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_crlpath) | Yes | Yes | Yes |  | Global | No |
| [**Mysqlx\_ssl\_ctx\_verify\_depth**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_ctx_verify_depth) |  |  |  | Yes | Both | No |
| [**Mysqlx\_ssl\_ctx\_verify\_mode**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_ctx_verify_mode) |  |  |  | Yes | Both | No |
| [**Mysqlx\_ssl\_finished\_accepts**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_finished_accepts) |  |  |  | Yes | Global | No |
| [**mysqlx\_ssl\_key**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_key) | Yes | Yes | Yes |  | Global | No |
| [**Mysqlx\_ssl\_server\_not\_after**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_server_not_after) |  |  |  | Yes | Global | No |
| [**Mysqlx\_ssl\_server\_not\_before**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_server_not_before) |  |  |  | Yes | Global | No |
| [**Mysqlx\_ssl\_verify\_depth**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_verify_depth) |  |  |  | Yes | Global | No |
| [**Mysqlx\_ssl\_verify\_mode**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_verify_mode) |  |  |  | Yes | Global | No |
| [**Mysqlx\_ssl\_version**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_ssl_version) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_create\_collection**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_create_collection) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_create\_collection\_index**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_create_collection_index) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_disable\_notices**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_disable_notices) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_drop\_collection**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_drop_collection) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_drop\_collection\_index**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_drop_collection_index) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_enable\_notices**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_enable_notices) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_ensure\_collection**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_ensure_collection) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_execute\_mysqlx**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_execute_mysqlx) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_execute\_sql**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_execute_sql) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_execute\_xplugin**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_execute_xplugin) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_get\_collection\_options**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_get_collection_options) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_kill\_client**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_kill_client) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_list\_clients**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_list_clients) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_list\_notices**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_list_notices) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_list\_objects**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_list_objects) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_modify\_collection\_options**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_modify_collection_options) |  |  |  | Yes | Both | No |
| [**Mysqlx\_stmt\_ping**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_stmt_ping) |  |  |  | Yes | Both | No |
| [**mysqlx\_wait\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_wait_timeout) | Yes | Yes | Yes |  | Session | Yes |
| [**Mysqlx\_worker\_threads**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_worker_threads) |  |  |  | Yes | Global | No |
| [**Mysqlx\_worker\_threads\_active**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#statvar_Mysqlx_worker_threads_active) |  |  |  | Yes | Global | No |
| [**mysqlx\_write\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_write_timeout) | Yes | Yes | Yes |  | Session | Yes |
| [**mysqlx\_zstd\_default\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_zstd_default_compression_level) | Yes | Yes | Yes |  | Global | Yes |
| [**mysqlx\_zstd\_max\_client\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_zstd_max_client_compression_level) | Yes | Yes | Yes |  | Global | Yes |

#### 20.5.6.2 X Plugin Options and System Variables

To control activation of X Plugin, use this option:

* [--mysqlx[=value]](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "option_mysqld_mysqlx)

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx[=value]** |
| **Type** | Enumeration |
| **Default Value** | **ON** |
| **Valid Values** | **ON**  **OFF**  **FORCE**  **FORCE\_PLUS\_PERMANENT** |

* This option controls how the server loads X Plugin at startup. In MySQL 8.0, X Plugin is enabled by default, but this option may be used to control its activation state.
* The option value should be one of those available for plugin-loading options, as described in [Section 5.6.1, “Installing and Uninstalling Plugins”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#plugin-loading).

If X Plugin is enabled, it exposes several system variables that permit control over its operation:

* **[mysqlx\_bind\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_bind_address)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-bind-address=addr** |
| **System Variable** | [**mysqlx\_bind\_address**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_bind_address) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | String |
| **Default Value** | **\*** |

* The network address on which X Plugin listens for TCP/IP connections. This variable is not dynamic and can be configured only at startup. This is the X Plugin equivalent of the **[bind\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_bind_address)** system variable; see that variable description for more information.
* By default, X Plugin accepts TCP/IP connections on all server host IPv4 interfaces, and, if the server host supports IPv6, on all IPv6 interfaces. If **[mysqlx\_bind\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_bind_address)** is specified, its value must satisfy these requirements:
  + Prior to MySQL 8.0.21, **[mysqlx\_bind\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_bind_address)** accepts a single address value, which may specify a single non-wildcard IP address (either IPv4 or IPv6), or a host name, or one of the wildcard address formats that permit listening on multiple network interfaces (**\***, **0.0.0.0**, or **::**).
  + As of MySQL 8.0.21, **[mysqlx\_bind\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_bind_address)** accepts either a single value as just described, or a list of comma-separated values. When the variable names a list of multiple values, each value must specify a single non-wildcard IP address (either IPv4 or IPv6) or a host name. Wildcard address formats (**\***, **0.0.0.0**, or **::**) are not allowed in a list of values.
  + As of MySQL 8.0.22, the value may include a network namespace specifier.

IP addresses can be specified as IPv4 or IPv6 addresses. For any value that is a host name, X Plugin resolves the name to an IP address and binds to that address. If a host name resolves to multiple IP addresses, X Plugin uses the first IPv4 address if there are any, or the first IPv6 address otherwise.

X Plugin treats different types of addresses as follows:

* + If the address is **\***, X Plugin accepts TCP/IP connections on all server host IPv4 interfaces, and, if the server host supports IPv6, on all IPv6 interfaces. Use this address to permit both IPv4 and IPv6 connections for X Plugin. This value is the default. If the variable specifies a list of multiple values, this value is not permitted.
  + If the address is **0.0.0.0**, X Plugin accepts TCP/IP connections on all server host IPv4 interfaces. If the variable specifies a list of multiple values, this value is not permitted.
  + If the address is **::**, X Plugin accepts TCP/IP connections on all server host IPv4 and IPv6 interfaces. If the variable specifies a list of multiple values, this value is not permitted.
  + If the address is an IPv4-mapped address, X Plugin accepts TCP/IP connections for that address, in either IPv4 or IPv6 format. For example, if X Plugin is bound to **::ffff:127.0.0.1**, a client such as MySQL Shell can connect using --host=127.0.0.1 or --host=::ffff:127.0.0.1.
  + If the address is a “regular” IPv4 or IPv6 address (such as **127.0.0.1** or **::1**), X Plugin accepts TCP/IP connections only for that IPv4 or IPv6 address.

These rules apply to specifying a network namespace for an address:

* + A network namespace can be specified for an IP address or a host name.
  + A network namespace cannot be specified for a wildcard IP address.
  + For a given address, the network namespace is optional. If given, it must be specified as a **/*ns*** suffix immediately following the address.
  + An address with no **/*ns*** suffix uses the host system global namespace. The global namespace is therefore the default.
  + An address with a **/*ns*** suffix uses the namespace named ***ns***.
  + The host system must support network namespaces and each named namespace must previously have been set up. Naming a nonexistent namespace produces an error.
  + If the variable value specifies multiple addresses, it can include addresses in the global namespace, in named namespaces, or a mix.

For additional information about network namespaces, see [Section 5.1.14, “Network Namespace Support”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#network-namespace-support).

**Important**

Because X Plugin is not a mandatory plugin, it does not prevent server startup if there is an error in the specified address or list of addresses (as MySQL Server does for **[bind\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_bind_address)** errors). With X Plugin, if one of the listed addresses cannot be parsed or if X Plugin cannot bind to it, the address is skipped, an error message is logged, and X Plugin attempts to bind to each of the remaining addresses. X Plugin's **[Mysqlx\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_address)** status variable displays only those addresses from the list for which the bind succeeded. If none of the listed addresses results in a successful bind, or if a single specified address fails, X Plugin logs the error message **ER\_XPLUGIN\_FAILED\_TO\_PREPARE\_IO\_INTERFACES** stating that X Protocol cannot be used. **[mysqlx\_bind\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_bind_address)** is not dynamic, so to fix any issues you must stop the server, correct the system variable value, and restart the server.

* **[mysqlx\_compression\_algorithms](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_compression_algorithms)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-compression-algorithms=value** |
| **Introduced** | 8.0.19 |
| **System Variable** | [**mysqlx\_compression\_algorithms**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_compression_algorithms) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Set |
| **Default Value** | **deflate\_stream,lz4\_message,zstd\_stream** |
| **Valid Values** | **deflate\_stream**  **lz4\_message**  **zstd\_stream** |

* The compression algorithms that are permitted for use on X Protocol connections. By default, the Deflate, LZ4, and zstd algorithms are all permitted. To disallow any of the algorithms, set **[mysqlx\_compression\_algorithms](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_compression_algorithms)** to include only the ones you permit. The algorithm names **deflate\_stream**, **lz4\_message**, and **zstd\_stream** can be specified in any combination, and the order and case are not important. If you set the system variable to the empty string, no compression algorithms are permitted and only uncompressed connections are used. Use the algorithm-specific system variables to adjust the default and maximum compression level for each permitted algorithm. For more details, and information on how connection compression for X Protocol relates to the equivalent settings for MySQL Server, see [Section 20.5.5, “Connection Compression with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression).

* **[mysqlx\_connect\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_connect_timeout)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-connect-timeout=#** |
| **System Variable** | [**mysqlx\_connect\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_connect_timeout) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **30** |
| **Minimum Value** | **1** |
| **Maximum Value** | **1000000000** |

* The number of seconds X Plugin waits for the first packet to be received from newly connected clients. This is the X Plugin equivalent of **[connect\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_connect_timeout)**; see that variable description for more information.

* **[mysqlx\_deflate\_default\_compression\_level](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_deflate_default_compression_level)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx\_deflate\_default\_compression\_level=#** |
| **Introduced** | 8.0.20 |
| **System Variable** | [**mysqlx\_deflate\_default\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_deflate_default_compression_level) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **3** |
| **Minimum Value** | **1** |
| **Maximum Value** | **9** |

* The default compression level that the server uses for the Deflate algorithm on X Protocol connections. Specify the level as an integer from 1 (the lowest compression effort) to 9 (the highest effort). This level is used if the client does not request a compression level during capability negotiation. If you do not specify this system variable, the server uses level 3 as the default. For more information, see [Section 20.5.5, “Connection Compression with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression).

* **[mysqlx\_deflate\_max\_client\_compression\_level](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_deflate_max_client_compression_level)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx\_deflate\_max\_client\_compression\_level=#** |
| **Introduced** | 8.0.20 |
| **System Variable** | [**mysqlx\_deflate\_max\_client\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_deflate_max_client_compression_level) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **5** |
| **Minimum Value** | **1** |
| **Maximum Value** | **9** |

* The maximum compression level that the server permits for the Deflate algorithm on X Protocol connections. The range is the same as for the default compression level for this algorithm. If the client requests a higher compression level than this, the server uses the level you set here. If you do not specify this system variable, the server sets a maximum compression level of 5.

* **[mysqlx\_document\_id\_unique\_prefix](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_document_id_unique_prefix)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-document-id-unique-prefix=#** |
| **System Variable** | [**mysqlx\_document\_id\_unique\_prefix**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_document_id_unique_prefix) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **0** |
| **Minimum Value** | **0** |
| **Maximum Value** | **65535** |

* Sets the first 4 bytes of document IDs generated by the server when documents are added to a collection. By setting this variable to a unique value per instance, you can ensure document IDs are unique across instances. See [Understanding Document IDs](https://dev.mysql.com/doc/x-devapi-userguide/en/understanding-automatic-document-ids.html).

* **[mysqlx\_enable\_hello\_notice](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_enable_hello_notice)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-enable-hello-notice[={OFF|ON}]** |
| **System Variable** | [**mysqlx\_enable\_hello\_notice**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_enable_hello_notice) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Boolean |
| **Default Value** | **ON** |

* Controls messages sent to classic MySQL protocol clients that try to connect over X Protocol. When enabled, clients which do not support X Protocol that attempt to connect to the server X Protocol port receive an error explaining they are using the wrong protocol.

* **[mysqlx\_idle\_worker\_thread\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_idle_worker_thread_timeout)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-idle-worker-thread-timeout=#** |
| **System Variable** | [**mysqlx\_idle\_worker\_thread\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_idle_worker_thread_timeout) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **60** |
| **Minimum Value** | **0** |
| **Maximum Value** | **3600** |

* The number of seconds after which idle worker threads are terminated.

* **[mysqlx\_interactive\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_interactive_timeout)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-interactive-timeout=#** |
| **System Variable** | [**mysqlx\_interactive\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_interactive_timeout) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **28800** |
| **Minimum Value** | **1** |
| **Maximum Value** | **2147483** |

* The default value of the **[mysqlx\_wait\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_wait_timeout)** session variable for interactive clients. (The number of seconds to wait for interactive clients to timeout.)

* **[mysqlx\_lz4\_default\_compression\_level](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_lz4_default_compression_level)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx\_lz4\_default\_compression\_level=#** |
| **Introduced** | 8.0.20 |
| **System Variable** | [**mysqlx\_lz4\_default\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_lz4_default_compression_level) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **2** |
| **Minimum Value** | **0** |
| **Maximum Value** | **16** |

* The default compression level that the server uses for the LZ4 algorithm on X Protocol connections. Specify the level as an integer from 0 (the lowest compression effort) to 16 (the highest effort). This level is used if the client does not request a compression level during capability negotiation. If you do not specify this system variable, the server uses level 2 as the default. For more information, see [Section 20.5.5, “Connection Compression with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression).

* **[mysqlx\_lz4\_max\_client\_compression\_level](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_lz4_max_client_compression_level)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx\_lz4\_max\_client\_compression\_level=#** |
| **Introduced** | 8.0.20 |
| **System Variable** | [**mysqlx\_lz4\_max\_client\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_lz4_max_client_compression_level) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **8** |
| **Minimum Value** | **0** |
| **Maximum Value** | **16** |

* The maximum compression level that the server permits for the LZ4 algorithm on X Protocol connections. The range is the same as for the default compression level for this algorithm. If the client requests a higher compression level than this, the server uses the level you set here. If you do not specify this system variable, the server sets a maximum compression level of 8.

* **[mysqlx\_max\_allowed\_packet](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_max_allowed_packet)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-max-allowed-packet=#** |
| **System Variable** | [**mysqlx\_max\_allowed\_packet**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_max_allowed_packet) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **67108864** |
| **Minimum Value** | **512** |
| **Maximum Value** | **1073741824** |

* The maximum size of network packets that can be received by X Plugin. This limit also applies when compression is used for the connection, so the network packet must be smaller than this size after the message has been decompressed. This is the X Plugin equivalent of **[max\_allowed\_packet](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_max_allowed_packet)**; see that variable description for more information.

* **[mysqlx\_max\_connections](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_max_connections)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-max-connections=#** |
| **System Variable** | [**mysqlx\_max\_connections**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_max_connections) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **100** |
| **Minimum Value** | **1** |
| **Maximum Value** | **65535** |

* The maximum number of concurrent client connections X Plugin can accept. This is the X Plugin equivalent of **[max\_connections](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_max_connections)**; see that variable description for more information.
* For modifications to this variable, if the new value is smaller than the current number of connections, the new limit is taken into account only for new connections.

* **[mysqlx\_min\_worker\_threads](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_min_worker_threads)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-min-worker-threads=#** |
| **System Variable** | [**mysqlx\_min\_worker\_threads**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_min_worker_threads) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **2** |
| **Minimum Value** | **1** |
| **Maximum Value** | **100** |

* The minimum number of worker threads used by X Plugin for handling client requests.

* **[mysqlx\_port](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_port)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-port=port\_num** |
| **System Variable** | [**mysqlx\_port**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_port) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **33060** |
| **Minimum Value** | **1** |
| **Maximum Value** | **65535** |

* The network port on which X Plugin listens for TCP/IP connections. This is the X Plugin equivalent of [**port**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_port); see that variable description for more information.

* **[mysqlx\_port\_open\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_port_open_timeout)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-port-open-timeout=#** |
| **System Variable** | [**mysqlx\_port\_open\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_port_open_timeout) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **0** |
| **Minimum Value** | **0** |
| **Maximum Value** | **120** |

* The number of seconds X Plugin waits for a TCP/IP port to become free.

* **[mysqlx\_read\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_read_timeout)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-read-timeout=#** |
| **System Variable** | [**mysqlx\_read\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_read_timeout) |
| **Scope** | Session |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **28800** |
| **Minimum Value** | **30** |
| **Maximum Value** | **2147483** |

* The number of seconds that X Plugin waits for blocking read operations to complete. After this time, if the read operation is not successful, X Plugin closes the connection and returns a warning notice with the error code ER\_IO\_READ\_ERROR to the client application.

* **[mysqlx\_socket](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_socket)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-socket=file\_name** |
| **System Variable** | [**mysqlx\_socket**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_socket) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | String |
| **Default Value** | **/tmp/mysqlx.sock** |

* The path to a Unix socket file which X Plugin uses for connections. This setting is only used by MySQL Server when running on Unix operating systems. Clients can use this socket to connect to MySQL Server using X Plugin.
* The default **[mysqlx\_socket](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_socket)** path and file name is based on the default path and file name for the main socket file for MySQL Server, with the addition of an **x** appended to the file name. The default path and file name for the main socket file is **/tmp/mysql.sock**, therefore the default path and file name for the X Plugin socket file is **/tmp/mysqlx.sock**.
* If you specify an alternative path and file name for the main socket file at server startup using the [**socket**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\server-administration.html#sysvar_socket) system variable, this does not affect the default for the X Plugin socket file. In this situation, if you want to store both sockets at a single path, you must set the **[mysqlx\_socket](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_socket)** system variable as well. For example in a configuration file:
* socket=/home/sockets/mysqld/mysql.sock
* mysqlx\_socket=/home/sockets/xplugin/xplugin.sock
* If you change the default path and file name for the main socket file at compile time using the [MYSQL\_UNIX\_ADDR](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysql_unix_addr) compile option, this does affect the default for the X Plugin socket file, which is formed by appending an **x** to the [MYSQL\_UNIX\_ADDR](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysql_unix_addr) file name. If you want to set a different default for the X Plugin socket file at compile time, use the [MYSQLX\_UNIX\_ADDR](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysqlx_unix_addr) compile option.
* The **MYSQLX\_UNIX\_PORT** environment variable can also be used to set a default for the X Plugin socket file at server startup (see [Section 4.9, “Environment Variables”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\programs.html#environment-variables)). If you set this environment variable, it overrides the compiled [MYSQLX\_UNIX\_ADDR](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\installing.html#option_cmake_mysqlx_unix_addr) value, but is overridden by the **[mysqlx\_socket](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_socket)** value.

* **[mysqlx\_ssl\_ca](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_ca)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-ssl-ca=file\_name** |
| **System Variable** | [**mysqlx\_ssl\_ca**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_ca) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | File name |
| **Default Value** | **NULL** |

* The **[mysqlx\_ssl\_ca](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_ca)** system variable is like **[ssl\_ca](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_ssl_ca)**, except that it applies to X Plugin rather than the MySQL Server main connection interface. For information about configuring encryption support for X Plugin, see [Section 20.5.3, “Using Encrypted Connections with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-encrypted-connections).

* **[mysqlx\_ssl\_capath](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_capath)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-ssl-capath=dir\_name** |
| **System Variable** | [**mysqlx\_ssl\_capath**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_capath) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Directory name |
| **Default Value** | **NULL** |

* The **[mysqlx\_ssl\_capath](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_capath)** system variable is like **[ssl\_capath](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_ssl_capath)**, except that it applies to X Plugin rather than the MySQL Server main connection interface. For information about configuring encryption support for X Plugin, see [Section 20.5.3, “Using Encrypted Connections with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-encrypted-connections).

* **[mysqlx\_ssl\_cert](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_cert)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-ssl-cert=file\_name** |
| **System Variable** | [**mysqlx\_ssl\_cert**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_cert) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | File name |
| **Default Value** | **NULL** |

* The **[mysqlx\_ssl\_cert](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_cert)** system variable is like **[ssl\_cert](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_ssl_cert)**, except that it applies to X Plugin rather than the MySQL Server main connection interface. For information about configuring encryption support for X Plugin, see [Section 20.5.3, “Using Encrypted Connections with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-encrypted-connections).

* **[mysqlx\_ssl\_cipher](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_cipher)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-ssl-cipher=name** |
| **System Variable** | [**mysqlx\_ssl\_cipher**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_cipher) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | String |
| **Default Value** | **NULL** |

* The **[mysqlx\_ssl\_cipher](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_cipher)** system variable is like **[ssl\_cipher](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_ssl_cipher)**, except that it applies to X Plugin rather than the MySQL Server main connection interface. For information about configuring encryption support for X Plugin, see [Section 20.5.3, “Using Encrypted Connections with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-encrypted-connections).

* **[mysqlx\_ssl\_crl](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_crl)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-ssl-crl=file\_name** |
| **System Variable** | [**mysqlx\_ssl\_crl**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_crl) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | File name |
| **Default Value** | **NULL** |

* The **[mysqlx\_ssl\_crl](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_crl)** system variable is like **[ssl\_crl](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_ssl_crl)**, except that it applies to X Plugin rather than the MySQL Server main connection interface. For information about configuring encryption support for X Plugin, see [Section 20.5.3, “Using Encrypted Connections with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-encrypted-connections).

* **[mysqlx\_ssl\_crlpath](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_crlpath)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-ssl-crlpath=dir\_name** |
| **System Variable** | [**mysqlx\_ssl\_crlpath**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_crlpath) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Directory name |
| **Default Value** | **NULL** |

* The **[mysqlx\_ssl\_crlpath](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_crlpath)** system variable is like **[ssl\_crlpath](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_ssl_crlpath)**, except that it applies to X Plugin rather than the MySQL Server main connection interface. For information about configuring encryption support for X Plugin, see [Section 20.5.3, “Using Encrypted Connections with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-encrypted-connections).

* **[mysqlx\_ssl\_key](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_key)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-ssl-key=file\_name** |
| **System Variable** | [**mysqlx\_ssl\_key**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_ssl_key) |
| **Scope** | Global |
| **Dynamic** | No |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | File name |
| **Default Value** | **NULL** |

* The **[mysqlx\_ssl\_key](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_ssl_key)** system variable is like **[ssl\_key](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_ssl_key)**, except that it applies to X Plugin rather than the MySQL Server main connection interface. For information about configuring encryption support for X Plugin, see [Section 20.5.3, “Using Encrypted Connections with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-encrypted-connections).

* **[mysqlx\_wait\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_wait_timeout)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-wait-timeout=#** |
| **System Variable** | [**mysqlx\_wait\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_wait_timeout) |
| **Scope** | Session |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **28800** |
| **Minimum Value** | **1** |
| **Maximum Value** | **2147483** |

* The number of seconds that X Plugin waits for activity on a connection. After this time, if the read operation is not successful, X Plugin closes the connection. If the client is noninteractive, the initial value of the session variable is copied from the global **[mysqlx\_wait\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_wait_timeout)** variable. For interactive clients, the initial value is copied from the session **[mysqlx\_interactive\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_interactive_timeout)**.

* **[mysqlx\_write\_timeout](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_write_timeout)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx-write-timeout=#** |
| **System Variable** | [**mysqlx\_write\_timeout**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_write_timeout) |
| **Scope** | Session |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **60** |
| **Minimum Value** | **1** |
| **Maximum Value** | **2147483** |

* The number of seconds that X Plugin waits for blocking write operations to complete. After this time, if the write operation is not successful, X Plugin closes the connection.

* **[mysqlx\_zstd\_default\_compression\_level](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_zstd_default_compression_level)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx\_zstd\_default\_compression\_level=#** |
| **Introduced** | 8.0.20 |
| **System Variable** | [**mysqlx\_zstd\_default\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_zstd_default_compression_level) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **3** |
| **Minimum Value** | **-131072** |
| **Maximum Value** | **22** |

* The default compression level that the server uses for the zstd algorithm on X Protocol connections. For versions of the zstd library from 1.4.0, you can set positive values from 1 to 22 (the highest compression effort), or negative values which represent progressively lower effort. A value of 0 is converted to a value of 1. For earlier versions of the zstd library, you can only specify the value 3. This level is used if the client does not request a compression level during capability negotiation. If you do not specify this system variable, the server uses level 3 as the default. For more information, see [Section 20.5.5, “Connection Compression with X Plugin”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression).

* **[mysqlx\_zstd\_max\_client\_compression\_level](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_zstd_max_client_compression_level)**

|  |  |
| --- | --- |
| **Command-Line Format** | **--mysqlx\_zstd\_max\_client\_compression\_level=#** |
| **Introduced** | 8.0.20 |
| **System Variable** | [**mysqlx\_zstd\_max\_client\_compression\_level**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#sysvar_mysqlx_zstd_max_client_compression_level) |
| **Scope** | Global |
| **Dynamic** | Yes |
| [**SET\_VAR**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\optimization.html#optimizer-hints-set-var)**Hint Applies** | No |
| **Type** | Integer |
| **Default Value** | **11** |
| **Minimum Value** | **-131072** |
| **Maximum Value** | **22** |

* The maximum compression level that the server permits for the zstd algorithm on X Protocol connections. The range is the same as for the default compression level for this algorithm. If the client requests a higher compression level than this, the server uses the level you set here. If you do not specify this system variable, the server sets a maximum compression level of 11.

#### 20.5.6.3 X Plugin Status Variables

The X Plugin status variables have the following meanings.

* **[Mysqlx\_aborted\_clients](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_aborted_clients)**

The number of clients that were disconnected because of an input or output error.

* **[Mysqlx\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_address)**

The network address or addresses for which X Plugin accepts TCP/IP connections. If multiple addresses were specified using the **[mysqlx\_bind\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_bind_address)** system variable, **[Mysqlx\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_address)** displays only those addresses for which the bind succeeded. If the bind has failed for every network address specified by **[mysqlx\_bind\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_bind_address)**, or if the **[skip\_networking](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_skip_networking)** option has been used, the value of **[Mysqlx\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_address)** is **UNDEFINED**. If X Plugin startup is not yet complete, the value of **[Mysqlx\_address](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_address)** is empty.

* **[Mysqlx\_bytes\_received](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_received)**

The total number of bytes received through the network. If compression is used for the connection, this figure comprises compressed message payloads measured before decompression (**[Mysqlx\_bytes\_received\_compressed\_payload](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_received_compressed_payload)**), any items in compressed messages that were not compressed such as X Protocol headers, and any uncompressed messages.

* **[Mysqlx\_bytes\_received\_compressed\_payload](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_received_compressed_payload)**

The number of bytes received as compressed message payloads, measured before decompression.

* **[Mysqlx\_bytes\_received\_uncompressed\_frame](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_received_uncompressed_frame)**

The number of bytes received as compressed message payloads, measured after decompression.

* **[Mysqlx\_bytes\_sent](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_sent)**

The total number of bytes sent through the network. If compression is used for the connection, this figure comprises compressed message payloads measured after compression (**[Mysqlx\_bytes\_sent\_compressed\_payload](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_sent_compressed_payload)**), any items in compressed messages that were not compressed such as X Protocol headers, and any uncompressed messages.

* **[Mysqlx\_bytes\_sent\_compressed\_payload](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_sent_compressed_payload)**

The number of bytes sent as compressed message payloads, measured after compression.

* **[Mysqlx\_bytes\_sent\_uncompressed\_frame](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_bytes_sent_uncompressed_frame)**

The number of bytes sent as compressed message payloads, measured before compression.

* **[Mysqlx\_compression\_algorithm](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_compression_algorithm)**

(Session scope) The compression algorithm in use for the X Protocol connection for this session. The permitted compression algorithms are listed by the **[mysqlx\_compression\_algorithms](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "sysvar_mysqlx_compression_algorithms)** system variable.

* **[Mysqlx\_compression\_level](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_compression_level)**

(Session scope) The compression level in use for the X Protocol connection for this session.

* **[Mysqlx\_connection\_accept\_errors](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_connection_accept_errors)**

The number of connections which have caused accept errors.

* **[Mysqlx\_connection\_errors](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_connection_errors)**

The number of connections which have caused errors.

* **[Mysqlx\_connections\_accepted](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_connections_accepted)**

The number of connections which have been accepted.

* **[Mysqlx\_connections\_closed](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_connections_closed)**

The number of connections which have been closed.

* **[Mysqlx\_connections\_rejected](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_connections_rejected)**

The number of connections which have been rejected.

* **[Mysqlx\_crud\_create\_view](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_crud_create_view)**

The number of create view requests received.

* **[Mysqlx\_crud\_delete](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_crud_delete)**

The number of delete requests received.

* **[Mysqlx\_crud\_drop\_view](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_crud_drop_view)**

The number of drop view requests received.

* **[Mysqlx\_crud\_find](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_crud_find)**

The number of find requests received.

* **[Mysqlx\_crud\_insert](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_crud_insert)**

The number of insert requests received.

* **[Mysqlx\_crud\_modify\_view](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_crud_modify_view)**

The number of modify view requests received.

* **[Mysqlx\_crud\_update](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_crud_update)**

The number of update requests received.

* **[Mysqlx\_cursor\_close](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_cursor_close)**

The number of cursor-close messages received

* **[Mysqlx\_cursor\_fetch](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_cursor_fetch)**

The number of cursor-fetch messages received

* **[Mysqlx\_cursor\_open](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_cursor_open)**

The number of cursor-open messages received

* **[Mysqlx\_errors\_sent](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_errors_sent)**

The number of errors sent to clients.

* **[Mysqlx\_expect\_close](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_expect_close)**

The number of expectation blocks closed.

* **[Mysqlx\_expect\_open](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_expect_open)**

The number of expectation blocks opened.

* **[Mysqlx\_init\_error](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_init_error)**

The number of errors during initialisation.

* **[Mysqlx\_messages\_sent](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_messages_sent)**

The total number of messages of all types sent to clients.

* **[Mysqlx\_notice\_global\_sent](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_notice_global_sent)**

The number of global notifications sent to clients.

* **[Mysqlx\_notice\_other\_sent](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_notice_other_sent)**

The number of other types of notices sent back to clients.

* **[Mysqlx\_notice\_warning\_sent](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_notice_warning_sent)**

The number of warning notices sent back to clients.

* **[Mysqlx\_notified\_by\_group\_replication](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_notified_by_group_replication)**

Number of Group Replication notifications sent to clients.

* **[Mysqlx\_port](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_port)**

The TCP port which X Plugin is listening to. If a network bind has failed, or if the **[skip\_networking](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\server-administration.html" \l "sysvar_skip_networking)** system variable is enabled, the value shows **UNDEFINED**.

* **[Mysqlx\_prep\_deallocate](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_prep_deallocate)**

The number of prepared-statement-deallocate messages received

* **[Mysqlx\_prep\_execute](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_prep_execute)**

The number of prepared-statement-execute messages received

* **[Mysqlx\_prep\_prepare](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_prep_prepare)**

The number of prepared-statement messages received

* **[Mysqlx\_rows\_sent](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_rows_sent)**

The number of rows sent back to clients.

* **[Mysqlx\_sessions](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_sessions)**

The number of sessions that have been opened.

* **[Mysqlx\_sessions\_accepted](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_sessions_accepted)**

The number of session attempts which have been accepted.

* **[Mysqlx\_sessions\_closed](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_sessions_closed)**

The number of sessions that have been closed.

* **[Mysqlx\_sessions\_fatal\_error](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_sessions_fatal_error)**

The number of sessions that have closed with a fatal error.

* **[Mysqlx\_sessions\_killed](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_sessions_killed)**

The number of sessions which have been killed.

* **[Mysqlx\_sessions\_rejected](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_sessions_rejected)**

The number of session attempts which have been rejected.

* **[Mysqlx\_socket](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_socket)**

The Unix socket which X Plugin is listening to.

* **[Mysqlx\_ssl\_accept\_renegotiates](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_accept_renegotiates)**

The number of negotiations needed to establish the connection.

* **[Mysqlx\_ssl\_accepts](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_accepts)**

The number of accepted SSL connections.

* **[Mysqlx\_ssl\_active](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_active)**

If SSL is active.

* **[Mysqlx\_ssl\_cipher](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_cipher)**

The current SSL cipher (empty for non-SSL connections).

* **[Mysqlx\_ssl\_cipher\_list](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_cipher_list)**

A list of possible SSL ciphers (empty for non-SSL connections).

* **[Mysqlx\_ssl\_ctx\_verify\_depth](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_ctx_verify_depth)**

The certificate verification depth limit currently set in ctx.

* **[Mysqlx\_ssl\_ctx\_verify\_mode](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_ctx_verify_mode)**

The certificate verification mode currently set in ctx.

* **[Mysqlx\_ssl\_finished\_accepts](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_finished_accepts)**

The number of successful SSL connections to the server.

* **[Mysqlx\_ssl\_server\_not\_after](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_server_not_after)**

The last date for which the SSL certificate is valid.

* **[Mysqlx\_ssl\_server\_not\_before](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_server_not_before)**

The first date for which the SSL certificate is valid.

* **[Mysqlx\_ssl\_verify\_depth](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_verify_depth)**

The certificate verification depth for SSL connections.

* **[Mysqlx\_ssl\_verify\_mode](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_verify_mode)**

The certificate verification mode for SSL connections.

* **[Mysqlx\_ssl\_version](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_ssl_version)**

The name of the protocol used for SSL connections.

* **[Mysqlx\_stmt\_create\_collection](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_create_collection)**

The number of create collection statements received.

* **[Mysqlx\_stmt\_create\_collection\_index](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_create_collection_index)**

The number of create collection index statements received.

* **[Mysqlx\_stmt\_disable\_notices](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_disable_notices)**

The number of disable notice statements received.

* **[Mysqlx\_stmt\_drop\_collection](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_drop_collection)**

The number of drop collection statements received.

* **[Mysqlx\_stmt\_drop\_collection\_index](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_drop_collection_index)**

The number of drop collection index statements received.

* **[Mysqlx\_stmt\_enable\_notices](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_enable_notices)**

The number of enable notice statements received.

* **[Mysqlx\_stmt\_ensure\_collection](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_ensure_collection)**

The number of ensure collection statements received.

* **[Mysqlx\_stmt\_execute\_mysqlx](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_execute_mysqlx)**

The number of StmtExecute messages received with namespace set to **mysqlx**.

* **[Mysqlx\_stmt\_execute\_sql](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_execute_sql)**

The number of StmtExecute requests received for the SQL namespace.

* **[Mysqlx\_stmt\_execute\_xplugin](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_execute_xplugin)**

The number of StmtExecute requests received for the **xplugin** namespace. From MySQL 8.0.19, the **xplugin** namespace has been removed so this status variable is no longer used.

* **[Mysqlx\_stmt\_get\_collection\_options](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_get_collection_options)**

The number of get collection object statements received.

* **[Mysqlx\_stmt\_kill\_client](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_kill_client)**

The number of kill client statements received.

* **[Mysqlx\_stmt\_list\_clients](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_list_clients)**

The number of list client statements received.

* **[Mysqlx\_stmt\_list\_notices](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_list_notices)**

The number of list notice statements received.

* **[Mysqlx\_stmt\_list\_objects](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_list_objects)**

The number of list object statements received.

* **[Mysqlx\_stmt\_modify\_collection\_options](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_modify_collection_options)**

The number of modify collection options statements received.

* **[Mysqlx\_stmt\_ping](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_stmt_ping)**

The number of ping statements received.

* **[Mysqlx\_worker\_threads](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_worker_threads)**

The number of worker threads available.

* **[Mysqlx\_worker\_threads\_active](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\document-store.html" \l "statvar_Mysqlx_worker_threads_active)**

The number of worker threads currently used.

### 20.5.7 Monitoring X Plugin

For general X Plugin monitoring, use the status variables that it exposes. See [Section 20.5.6.3, “X Plugin Status Variables”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-status-variables). For information specifically about monitoring the effects of message compression, see [Monitoring Connection Compression for X Plugin](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\document-store.html#x-plugin-connection-compression-monitoring).

#### Monitoring SQL Generated by X Plugin

This section describes how to monitor the SQL statements which X Plugin generates when you run X DevAPI operations. When you execute a CRUD statement, it is translated into SQL and executed against the server. To be able to monitor the generated SQL, the Performance Schema tables must be enabled. The SQL is registered under the [**performance\_schema.events\_statements\_current**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\performance-schema.html#performance-schema-events-statements-current-table), [**performance\_schema.events\_statements\_history**](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\performance-schema.html#performance-schema-events-statements-history-table), and **[performance\_schema.events\_statements\_history\_long](file:///E:\\backup\\%E4%B8%8B%E8%BD%BD\\refman-8.0-en.html-chapter\\refman-8.0-en.html-chapter\\performance-schema.html" \l "performance-schema-events-statements-history-long-table" \o "27.12.6.3 The events_statements_history_long Table)** tables. The following example uses the **world\_x** schema, imported as part of the quickstart tutorials in this section. We use MySQL Shell in Python mode, and the **\sql** command which enables you to issue SQL statements without changing to SQL mode. This is important, because if you instead try to switch to SQL mode, the procedure shows the result of this operation rather than the X DevAPI operation. The **\sql** command is used in the same way if you are using MySQL Shell in JavaScript mode.

1. Check if the **events\_statements\_history** consumer is enabled. Issue:
2. mysql-py> **\sql SELECT enabled FROM performance\_schema.setup\_consumers WHERE NAME = 'events\_statements\_history'**
3. +---------+
4. | enabled |
5. +---------+
6. | YES |
7. +---------+
8. Check if all instruments report data to the consumer. Issue:

mysql-py> **\sql SELECT NAME, ENABLED, TIMED FROM performance\_schema.setup\_instruments WHERE NAME LIKE 'statement/%' AND NOT (ENABLED and TIMED)**

If this statement reports at least one row, you need to enable the instruments. See [Section 27.4, “Performance Schema Runtime Configuration”](file:///E:\backup\%E4%B8%8B%E8%BD%BD\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\performance-schema.html#performance-schema-runtime-configuration).

1. Get the thread ID of the current connection. Issue:

mysql-py> **\sql SELECT thread\_id INTO @id FROM performance\_schema.threads WHERE processlist\_id=connection\_id()**

1. Execute the X DevAPI CRUD operation for which you want to see the generated SQL. For example, issue:

mysql-py> **db.CountryInfo.find("Name = :country").bind("country", "Italy")**

You must not issue any further operations for the next step to show the correct result.

1. Show the last SQL query made by this thread ID. Issue:
2. mysql-py> **\sql SELECT THREAD\_ID, MYSQL\_ERRNO,SQL\_TEXT FROM performance\_schema.events\_statements\_history WHERE THREAD\_ID=@id ORDER BY TIMER\_START DESC LIMIT 1;**
3. +-----------+-------------+--------------------------------------------------------------------------------------+
4. | THREAD\_ID | MYSQL\_ERRNO | SQL\_TEXT |
5. +-----------+-------------+--------------------------------------------------------------------------------------+
6. | 29 | 0 | SELECT doc FROM `world\_x`.`CountryInfo` WHERE (JSON\_EXTRACT(doc,'$.Name') = 'Italy') |
7. +-----------+-------------+--------------------------------------------------------------------------------------+

The result shows the SQL generated by X Plugin based on the most recent statement, in this case the X DevAPI CRUD operation from the previous step.