

# MySQL Connections

## Contents:

- *Configure*
  - *Connection URL*
  - *Driver Information*
  - *Create via API*
- *Troubleshooting*
- *Use*
- *Data Conversion*

**Feature Availability:** This feature is available in the following editions:

- Trifacta® Enterprise Edition
- Trifacta Professional Edition
- Trifacta Premium

You can create connections to one or more MySQL databases from Trifacta®. For more information on MySQL, see <https://www.mysql.com/>.

**Tip:** You can create connections to databases of this type that are managed by your enterprise or are hosted in cloud infrastructure. The required configuration is the same. The cloud-based version is labeled **on Amazon RDS**. In the Create Connection dialog, you can search for that term.

If you are connecting Trifacta to any relational source of data, such as Redshift or Oracle, you must add the Trifacta a Service to your whitelist for those resources. See *Whitelist Platform Service*.

**Supported Versions:** 5.7 and 8.0 Community

- **Read:** Supported
- **Write:** Supported

## Configure

To create this connection:

- In the Import Data page, click the Plus sign. Then, select the Relational tab. Click the MySQL card.
- You can also create connections through the Connections page.
- See *Connections Page*.

Modify the following properties as needed:

Property	Description
----------	-------------

Host	Enter your fully qualified hostname. Example:  <code>mysql-server.example.net</code>
Port	Set this value to 3306.
Connect String Options	Insert any additional connection parameters, if needed. See below.
User Name	Username to use to connect to the database.
Password	Password associated with the above username.
Test Connection	After you have defined the connection credentials type, credentials, and connection string, you can validate those credentials.
Advanced options: Default Column Data Type Inference	Set to <code>disabled</code> to prevent the platform from applying its own type inference to each column on import. The default value is <code>enabled</code> .
Advanced options: Enable SSH Tunneling	<p>If available, the SSH tunneling options allow you to configure SSH tunneling authentication between the Trifacta application and your database.</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>NOTE:</b> SSH tunneling is available on a per-connection basis. It may not be available for all connections.</p> </div> <p>For more information, see <i>Configure SSH Tunnel Connectivity</i>.</p>
Connection Name	Display name of the connection
Connection Description	Description of the connection, which appears in the application.

## Connection URL

The properties that you provide are inserted into the following URL, which connects Trifacta to the connection:

```
jdbc:mysql://<host>:<port>/<database><connect-string-options>
```

## Connect string options

The connect string options are optional. If you are passing additional properties and values to complete the connection, the connect string options must be structured in the following manner:

```
&<prop1>=<val1>&<prop2>=<val2>...
```

where:

- `<prop>` : the name of the property
- `<val>` : the value for the property

delimiters:

- `&` : all additional property names must be prefixed with an ampersand (&).
- `=` : property names and values must be separated with an equal sign (=).

## Default connect string options

The following connect string options are specified by default.

**NOTE:** These options should not be overridden or modified.

The following connect string option requires the driver to use cursor-based fetching to retrieve rows.

```
useCursorFetch=true;
```

## Enable TLS (SSL)

You can insert the following connection string option to enable secure (TLS) connectivity with the MySQL server. Please note the TLS version numbers in the string listed below:

```
&enabledTLSProtocols=TLSv1,TLSv1.1,TLSv1.2
```

## Driver Information

This connection uses the following driver:

- **Driver name:** `com.mysql.cj.jdbc.Driver`
- **Driver version:** `8.0`
- **Driver documentation:** <https://dev.mysql.com/doc/connector-j/8.0/en/>

## Create via API

This connection can also be created using the API.

- Type: `jdbc`
- Vendor: `mysql`

For more information, see [Trifacta API Reference docs: Enterprise | Professional | Premium](#)

## Troubleshooting

Error message	Description
1042 - ER_BAD_HOST_ERROR	Unable to connect to host. Please verify the host and port values.
1045 - ER_ACCESS_DENIED_ERROR	Credentials failed to connect. Please verify your credentials.  <b>Tip:</b> Click the Test Connection button to verify that your credentials are working properly.
Error: zero date value prohibited	Set the following option in the connect string options: <pre>zeroDateTimeBehavior=convertToNull</pre>

Prepared statement needs to be re-prepared.	Database Cursor is not compatible with PREPARED statements in MySQL. The fix is to set the following in the Connect String Options: <div data-bbox="727 210 1458 283" style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <pre>useCursorFetch=false</pre> </div>
SSLHandshakeException: No appropriate protocol (protocol is disabled or cipher suites are inappropriate)	SSL ciphers need to be enabled. For more information, see "Enable TLS (SSL)" above.

For more information on error messages for this connection type, see <https://dev.mysql.com/doc/refman/8.0/en/error-handling.html>.

## Use

For more information, see *Database Browser*.

## Data Conversion

For more information on how values are converted during input and output with this database, see *MySQL Data Type Conversions*.