

Enable Teradata Connections

Contents:

- *Limitations*
 - *Download and Install Teradata drivers*
 - *Increase Read Timeout*
 - *Create Teradata Connection*
 - *Testing*
-

This section provides information on how to enable connection to Teradata databases.

- Teradata provides Datawarehousing & Analytics solutions and Marketing applications. The Teradata database supports all of their Datawarehousing solutions. For more information, see <http://www.teradata.com>.
- For more information on supported versions, see *Connection Types*.

This connection supports reading and writing. You can create multiple Teradata connections in the Trifacta application.

Limitations

- By default, Teradata does not permit the publication of datasets containing duplicate rows. Workarounds:
 - Your final statement for any recipe that generates results for Teradata should include a `REMOVE duplicate rows` transformation.

NOTE: The above transformation removes exact, case-sensitive duplicate rows. Teradata may still prevent publication for case-insensitive duplicates.

- It's possible to change the default writing method to Teradata to enable duplicate rows. For more information, contact *Trifacta Support*.
- When creating custom datasets using SQL from Teradata sources, the `ORDER BY` clause in standard SQL does not work. This is a known issue.

Download and Install Teradata drivers

To enable connectivity, you must download and install the Teradata drivers into an accessible location on the Trifacta® node.

NOTE: Please download and install the Teradata driver that corresponds to your version of Teradata. For more information on supported versions, see *Connection Types*.

Steps:

1. If you don't have a Teradata developer account, create one here:
<https://downloads.teradata.com/user/register>
2. Log in to the account. Navigate to <http://downloads.teradata.com/download/connectivity/jdbc-driver>
3. Download the JDBC driver in ZIP or TAR form.
4. Copy the downloaded ZIP or TAR file to the Trifacta node.
5. Extract and place the JAR file into a folder accessible to the Trifacta user.
6. Verify that the Trifacta user is the owner of the JAR file and its parent folder.

7. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`

. For more information, see *Platform Configuration Methods*.

8. Locate the `data-service.classpath`. To the classpath value add the folder where you installed the JAR file. For the new entry, remember to add the following to the entry:
 - a. Add a prefix of `:`.
 - b. Add a suffix of `/*`.
 - c. Example:

```
:/opt/trifacta/drivers/*
```

- d. Whole classpath example:

```
"data-service": { ...  
  "classpath": "%(topOfTree)s/services/data-service/build/libs/data-service.jar:%(topOfTree)s  
/services/data-service/build/conf:%(topOfTree)s/services/data-service/build/dependencies/*:/opt  
/trifacta/drivers/*"
```

9. Save your changes and restart the platform.

Increase Read Timeout

Particularly when reading from large Teradata tables, you might experience read timeouts in the Trifacta application.

The default setting is 300 seconds (5 minutes). You should consider raising this limit if you are working with large tables.

For more information, see *Configure Photon Running Environment*.

Create Teradata Connection

For more information on creating a Teradata connection, see *Create Connection Window*.

Testing

Steps:

1. After you create your connection, load a small dataset based on a table in the connected Teradata database. See *Import Data Page*.
2. Perform a few simple transformations to the data. Run the job. See *Transformer Page*.
3. Verify the results.

For more information, see *Verify Operations*.