

Using Databases

Contents:

- *Before You Begin*
 - *Access*
 - *Storing Data in Relational Databases*
 - *Reading from Database Tables and Views*
 - *Additional Notes on Database Views*
 - *Running Jobs from Database Sources*
 - *Writing to Databases*
-

This section describes how you interact with your databases through Trifacta® Self-Managed Enterprise Edition.

- Specific versions of each database are supported.
- Connections must be enabled and configured for each type of supported database.
- See *Connection Types*.

Before You Begin

- **Read Access:** Your database administrator must configure read permissions to the appropriate databases, tables and views for your use.

NOTE: To ensure that all user credentials used to access the database system are securely stored, you must first deploy the encryption key file to the Trifacta node. See *Enable Relational Connections*.

- **Write Access:** Some relational connection types support write access. For more information, see *Connection Types*.
 - This feature must be enabled. See *Enable Relational Connections*.

Access

Database access is managed through connections.

- Individual users can create private connections through the application. See *Create Connection Window*.
- An administrator can make your connection public or create public connections through the application.

Storing Data in Relational Databases

NOTE: Trifacta Self-Managed Enterprise Edition does not modify source data nor store transformed data in the relational systems. Datasets sourced from database tables or views are read without modification from their source locations.

Reading from Database Tables and Views

You can create a Trifacta dataset from a table or view stored in a connected database.

Tip: In some scenarios, you can improve performance of loading from database tables by creating a view on the table to restrict the amount of data loaded to only the required fields. Additionally, you can pre-filter the dataset using custom SQL statements. See *Create Dataset with SQL*.

Additional Notes on Database Views

- Some metadata, such as row counts, is not available for database views.
- For complex view definitions that require significant processing on the database, there may be a significant delay when previewing the contents of those views. In some cases, the preview may time out waiting for the database to respond with the view contents.

For more information, see *Database Browser*.

Running Jobs from Database Sources

NOTE: When executing a job using a relational source, the job may fail if one or more columns has been dropped from the underlying source table. As a workaround, the recipe panel may show steps referencing the missing columns, which be used to fix to either fix the recipe or the source data.

Writing to Databases

Relational connections can be configured to support writing results back to the database.

NOTE: You can only write to databases from the Run Job page. You cannot ad-hoc publish to a relational database.

NOTE: When writing to a new table in a relational target, the first entry in any mapping is used for writing out the value. Subsequent entries in the mapping are used for validation only on writing to new tables. See *Connection Types*.

Natively supported connection types are automatically enabled for writeback.