

# Using SQL DW

## Contents:

- *Limitations*
  - *Uses of SQL DW*
  - *Before You Begin Using SQL DW*
    - *Secure Access*
  - *Storing Data in SQL DW*
  - *Reading from SQL DW*
  - *Writing to SQL DW*
- 

This section describes how you interact through the Trifacta® platform with your SQL DW data warehouse.

- SQL DW is a scalable data warehouse solution available through Microsoft Azure. For more information, see <https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-overview-what-is>.

## Limitations

- Microsoft SQL DW connections are available only if you have deployed the Trifacta platform onto Azure.
- When using the SQL DW read-write connection, the defined length of a table row cannot exceed 1 MB.

**NOTE:** In this release, this connection cannot be created through the APIs. Please create connections of this type through the application.

## Uses of SQL DW

The Trifacta platform can use SQL DW for the following tasks:

1. Create datasets by reading from SQL DW tables.
2. Write to SQL DW tables with your job results.
3. Ad-hoc publication of data to SQL DW.

## Before You Begin Using SQL DW

- **Enable SQL DW Access:** SQL DW integration requires the following:
  - Installation of the Trifacta platform on Microsoft Azure.
  - Either ADL or WASB is supported as the base storage layer. For more information, see *Set Base Storage Layer*.
- **Read Access:** Your SQL DW administrator must configure read permissions. Your administrator should provide a database for upload to your SQL DW datastore.
- **Write Access:** You can write and publish jobs results to SQL DW.

## Secure Access

SQL DW connections require SSL access.

## Storing Data in SQL DW

Your SQL DW administrator should provide database access for storing datasets. Users should know where shared data is located and where personal data can be saved without interfering with or confusing other users.

**NOTE:** The Trifacta platform does not modify source data in SQL DW. Datasets sourced from SQL DW are read without modification from their source locations.

## Reading from SQL DW

You can create a Trifacta dataset from a table stored in SQL DW.

For more information, see *Database Browser*.

## Writing to SQL DW

You can write back data to SQL DW using one of the following methods:

- Job results can be written directly to SQL DW as part of the normal job execution. Create a new publishing action to write to SQL DW. See *Run Job Page*.
- As needed, you can publish results to SQL DW for previously executed jobs.

**NOTE:** You cannot re-publish results to SQL DW if the original job published to SQL DW. However, if the dataset was transformed but publication to SQL DW failed, you can publish from the Publishing dialog.

**NOTE:** To publish to SQL DW, the source results must be Parquet format.

See *Publishing Dialog*.

- For more information on how data is converted to SQL DW, see *SQL DW Data Type Conversions*.

### Data Validation issues:

- No validation is performed for the connection and any required permissions during job execution. So, you can be permitted to launch your job even if you do not have sufficient connectivity or permissions to access the data. The corresponding publish job fails at runtime.
- No data validation is performed during writing and publication to SQL DW. Your job fails if the schema for the Trifacta dataset varies from the target schema.
- Prior to publication, no validation is performed on whether a target is a table or a view, so the job that was launched fails at runtime.