

Using Redshift

Contents:

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

This section describes how you interact through Designer Cloud powered by Trifacta® Enterprise Edition with your Redshift data warehouse.

- Redshift is a scalable data warehouse solution available through Amazon. For more information, see <https://aws.amazon.com/redshift/>.

Limitations

- No schema validation is performed as part of writing results to Redshift.
- Credentials and permissions are not validated when you are modifying the destination for a publishing job.
- For Redshift, no validation is performed to determine if the target is a view and is therefore not a supported target.
- After a dataset has been imported using custom SQL from Redshift, disabling type inference may not revert to the source data types for some columns. The workaround is to create a new imported dataset using the same custom SQL with type inference disabled before import. After the dataset is created, use it as a replacement for the corrupted instances of the previous Redshift dataset.

Uses of Redshift

The Designer Cloud powered by Trifacta platform can use Redshift for the following tasks:

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

Before You Begin Using Redshift

-

Enable S3 Sources: Redshift integration requires the following:

- S3 is set to the base storage layer.
- For more information, see *Enable S3 Access*.
- **Read Access:** Your Redshift administrator must configure read permissions. Your administrator should provide a database for upload to your Redshift data store.
- **Write Access:** You can write and publish jobs results to Redshift.

Secure Access

SSL is required.

Storing Data in Redshift

Your Redshift 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.

i NOTE: Designer Cloud powered by Trifacta Enterprise Edition does not modify source data in Redshift. Datasets sourced from Redshift are read without modification from their source locations.

Reading from Redshift

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

i NOTE: The Redshift cluster must be in the same region as the default S3 bucket.

i NOTE: If a Redshift connection has an invalid iamRoleArn, you can browse, import datasets, and open the data in the Transformer page. However, any jobs executed using this connection fail. If the iamRoleArn is invalid, the only samples that you can generate are Quick Random samples; other sampling jobs fail.

For more information, see *Redshift Browser*.

Writing to Redshift

i NOTE: You cannot publish to a Redshift database that is empty. The database must contain at least one table.

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

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

i NOTE: You cannot re-publish results to Redshift if the original job published to Redshift. However, if the dataset was transform but publication to Redshift failed, you can publish from the Publishing dialog.

i NOTE: To publish to Redshift, the source results must be in Avro or JSON format.

- For more information on how data is converted to Redshift, see *Redshift 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.
- 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.