Using AWS Glue

Contents:

This section describes how you interact through the Trifacta® platform with your AWS Glue data warehouse via AWS Glue Catalog.

Uses of Glue

The Trifacta platform can use Glue for the following tasks:

1. Create datasets by reading from Glue tables.

Before You Begin Using Glue

- **Read Access:** Your Glue administrator must configure read permissions to Glue databases.
- **Write Access:** Not supported.

Secure Access

For more information, see *Configure for AWS*.

Reading Partitioned Data

The Trifacta platform can read in partitioned tables. However, it cannot read individual partitions of partitioned tables.

> **Tip:** If you are reading data from a partitioned table, one of your early recipe steps in the Transformer page should filter out the unneeded table data so that you are reading only the records of the individual partition.

Storing Data in Glue

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 Glue. Datasets sourced from Glue are read without modification from their source locations.

Reading from Glue

You can create a Trifacta dataset from a table or view stored in Glue. For more information, see *AWS Glue Browser*.

Notes on reading from views using custom SQL

If you have enabled custom SQL and are reading data from a view, nested functions are written to a temporary filename, unless they are explicitly aliased.

> **Tip:** If your custom SQL uses nested functions, you should create an explicit alias from the results. Otherwise, the job is likely to fail.

Problematic Example:

```
SELECT
 UPPER(`t1`.`colum1`),
 TRIM(`t1`.`column2`),...
```

When these are read from a Glue view, the temporary column names are: `_c0`, `_c1`, etc. During job execution, Spark ignores the `column1` and `column2` reference.

Improved Example:

```
SELECT
 UPPER(`t1`.`column1`) as col1,
  TRIM(`t1`.`column2`) as col2,...
```

In this improved example, the two Glue view columns are aliased to the explicit column names, which are correctly interpreted and used by the Spark running environment during job execution.

Writing to Glue

Not supported.