

The Teradata Plugins For Dataiku

**Document For Release 0.2.2.0
July 2021
Copyright © 2021 Teradata**

Table of Contents

1. Introduction	3
1.1. <i>Teradata Advanced SQL Engine Functions Plugin</i>	<i>3</i>
1.2. <i>Teradata SCRIPT Table Operator Plugin.....</i>	<i>4</i>
2. Requirements.....	5
2.1. <i>Dataiku Data Science Studio versions 8.0.2 to 9.0.4</i>	<i>5</i>
2.2. <i>Plugin</i>	<i>5</i>
2.3. <i>Access Credentials</i>	<i>5</i>
2.4. <i>Teradata JDBC Driver.....</i>	<i>5</i>
2.5. <i>Teradata Vantage System</i>	<i>6</i>
3. Creating A Vantage Connection.....	7
4. Plugin Download And Installation	10
4.1. <i>Get The Plugin</i>	<i>10</i>
4.2. <i>Plugin Installation.....</i>	<i>10</i>
5. Using the Teradata Plugins For Dataiku	12
5.1. <i>Using the Teradata Advanced SQL Engine Functions Plugin.....</i>	<i>12</i>
5.1.1. <i>Instructions</i>	<i>12</i>
5.1.2. <i>Usage Notes</i>	<i>15</i>
5.1.3. <i>Troubleshooting.....</i>	<i>16</i>
5.2. <i>Using the Teradata SCRIPT Table Operator Plugin.....</i>	<i>17</i>
5.2.1. <i>Script Loading.....</i>	<i>17</i>
5.2.2. <i>SCRIPT Table Operator Arguments</i>	<i>19</i>
5.2.3. <i>Other SQL Arguments</i>	<i>20</i>
5.2.4. <i>Running the Teradata SCRIPT Table Operator Plugin.....</i>	<i>20</i>

1. Introduction

Dataiku's Data Science Studio (DSS) is a collaborative platform that enables teams of people with different data expertise, such as data engineers, data scientists and business analysts, to work together efficiently. Dataiku provides a broad set of built-in recipes (operations) that can be applied to transform or analyze a dataset. It also allows users to create their own recipes in Python, SQL or R. Further on, Dataiku features the use of plugins; these are custom, reusable recipes that users can program in Python with AngularJS-based user interfaces (UI).

Teradata Vantage users can establish Dataiku connections to target Vantage systems to perform a variety of tasks with data in Vantage. For instance:

- DSS visual recipes can be applied to explore and transform Vantage data by pushing down code for execution into Vantage nodes seamlessly through the recipes.
- Dataiku models can be exported to PMML format to implement a BYOM (bring your own model) approach in Vantage, and score Vantage tables with them.
- The Teradata plugins for Dataiku can provide easy access to in-Database analytics.

Teradata has created plugins for Dataiku that expand and enhance the built-in interaction capabilities of DSS with Teradata Vantage systems. Recipes within the plugins provide select Vantage analytic tools wrappers, so that you can push the corresponding operations into Vantage in a visual and SQL-less manner in the user-friendly style of the DSS environment.

Teradata currently supports 2 plugins for Dataiku, namely, (a) the Teradata Advanced SQL Engine Analytic Functions Plugin, and (b) the Teradata SCRIPT Table Operator (STO) Plugin. There are also presently unsupported offerings of a Teradata ML Engine Analytic Functions Plugin, and a Teradata BYOM plugin.

The present guide focuses on installation and usage of the supported Teradata plugins for Dataiku. In general, for Dataiku DSS configuration and troubleshooting issues, please contact your Dataiku support team. For issues with the Teradata plugins, please contact Teradata Customer Support. In the present user guide, any specific troubleshooting topics for a plugin are covered in the corresponding plugin instructions in Chapter 5.

1.1. Teradata Advanced SQL Engine Functions Plugin

The Teradata Advanced SQL Engine Functions Plugin for Dataiku integrates a set of analytic functions that reside in the Vantage Advanced SQL Engine, by providing a user-friendly, easy-to-use, no-SQL interface for these functions in the Dataiku DSS environment. The Vantage Advanced SQL Engine functions can be accessed through the [+RECIPE] menu of the FLOW view of a Dataiku project, and are grouped into the following 2 recipes-categories:

- Data Transformation
- Time Series Analysis

In the background of the Teradata Advanced SQL Engine Functions Plugin user interface, the plugin essentially translates the end-user input from the plugin screens into SQL queries that are sent to the Advanced SQL Engine of a connected Vantage system via JDBC. This way, all

analytic queries are executed in-database, while also all input and output managed datasets are physically located in the database of the Advanced SQL Engine on the connected Vantage system.

The current plugin versioning is tied to the Vantage Advanced SQL Engine analytic functions releases. In that light, plugin version `x.y.z-a` is interpreted as follows: `x.y.z` is the Vantage Advanced SQL Engine analytic functions release that the plugin version caters to, and `a` is the plugin release, which is a number that may increase in case of subsequent fix/feature releases. For example, the inaugural plugin version is tied to the Vantage Advanced SQL Engine analytic functions release 2.0, and, per the previous, the plugin version will be 2.0-1. The plugin is currently at version 2.0-3.

1.2. Teradata SCRIPT Table Operator Plugin

The Teradata STO Plugin allows the execution of R or Python scripts inside the Advanced SQL Engine Database. The plugin will take an R or Python script within a DSS notebook, or an R or Python script uploaded to the plugin and install the scripts and other related files (i.e., saved models in RDS or pickle files) onto the Advanced SQL Engine.

The Teradata STO Plugin translates the user-requested tasks in the plugin into SQL queries. Queries are then sent in the background to a connected Vantage system to set up and invoke the SCRIPT Table Operator in the target Advanced SQL Engine.

To execute R or Python scripts inside the Advanced SQL Engine Database with the SCRIPT Table Operator, note that the Teradata In-nodes R and Python packages must be installed in advance in the target Advanced SQL Engine; see Section 2.5 for detailed requirements.

The plugin versioning follows the syntax `x.y.z` that designates the major version as `x`, the minor version as `y`, and bug fix releases as `z`. The plugin is currently at version 1.0.3.

2. Requirements

2.1. Dataiku Data Science Studio versions 8.0.2 to 9.0.4

The Dataiku DSS Enterprise Edition is required to import datasets from Vantage tables. Dataiku offers both downloadable and online options which can be obtained from the Dataiku website at <https://www.dataiku.com>. The downloadable option can be licensed to operate in the mode of one of the few available editions ("Free", "Discover", "Business" and "Enterprise"). For more information, details, and comparisons, visit the Dataiku dedicated site at <https://www.dataiku.com/product/plans-and-features>.

The Teradata plugins for Dataiku have been tested on Dataiku DSS versions 8.0.2 – 9.0.4.

2.2. Plugin

To use any of the plugins, you will need to download and install them first; see Chapter 4.

2.3. Access Credentials

To use the plugins, you will need 2 different kinds of credentials, that is, both Dataiku credentials and Vantage Advanced SQL Engine credentials. Specifically:

- a. Dataiku DSS user credentials allow a user to login to a DSS instance. Your DSS server administrator can provide you with these credentials.
- b. Vantage credentials allow a user to connect to the Advanced SQL Engine Database of a Vantage system, and, with appropriate permissions, read and write tables into the Advanced SQL Engine. Your Vantage database administrator (DBA) can provide you with credentials and suitable permissions for one or more databases on a Vantage system.

Use your DSS user credentials to log on to a DSS instance, and then use your Vantage credentials to establish a connection between DSS and a Vantage system. In the present guide, Section 3 ("Creating A Vantage Connection") provides instructions on how to setup a DSS connection to a Vantage Advanced SQL Engine Database. It is suggested to create one connection per each database for which you intend to store output tables in.

2.4. Teradata JDBC Driver

The Teradata JDBC Driver 16.20 or later is required to establish a connection between DSS and a Vantage System.

2.5. Teradata Vantage System

The Teradata Plugins for Dataiku require a connection to a Teradata Vantage system that minimally comprises of an Advanced SQL Engine.

Furthermore, to use the Teradata STO Plugin with a Vantage system Advanced SQL Engine and execute R and Python scripts in the Advanced SQL Engine nodes, the corresponding language bundles need to be installed directly on each node of the Advanced SQL Engine, per the following table:

PID	Product name
9687-2000-0120	R Interpreter and Add-on Pkg on Teradata Advanced SQL
9687-2000-0122	Python Interpreter and Add-on Pkg on Teradata Advanced SQL

Moreover, your DBA must grant you in advance the additional following privileges:

- `EXECUTE` Function privilege on `TD_SYSFNLIB.SCRIPT`
This is needed in order to invoke the `SCRIPT` Table Operator.
- `EXECUTE` privilege on the functions `SYSUIF.INSTALL_FILE`, `SYSUIF.REMOVE_FILE`, and `SYSUIF.REPLACE_FILE`.

3. Creating A Vantage Connection

1. Follow the instructions in the Dataiku Reference Document for Installing Database Drivers at <https://doc.dataiku.com/dss/latest/installation/custom/jdbc.html>

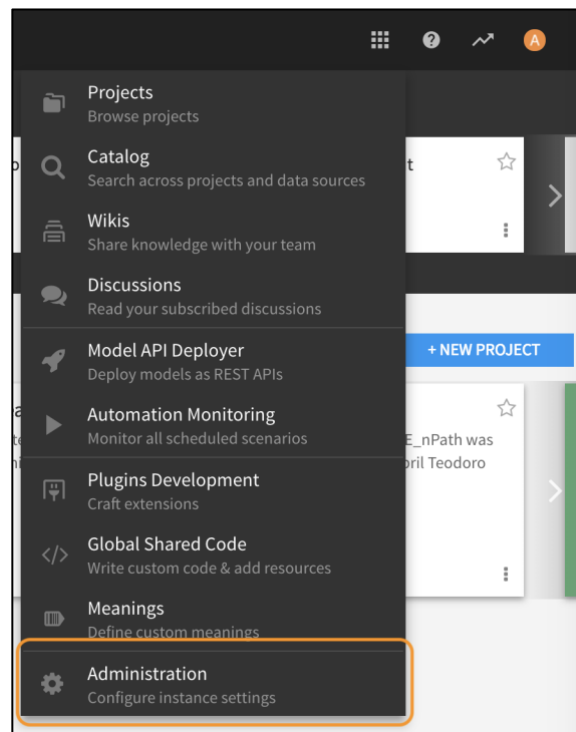
In summary, one needs to execute from the command line of a DSS server:

- a. Stop the Data Science Studio server, where `DATA_DIR` is the data directory where Data Science Studio is installed:
`DATA_DIR/bin/dss stop`
- b. Copy the Teradata JDBC driver to the `DATA_DIR/lib/jdbc` directory.
- c. Restart Data Science Studio:
`DATA_DIR/bin/dss start`

2. Access Dataiku DSS on a browser. Then, on the Dataiku DSS home page click on Apps.

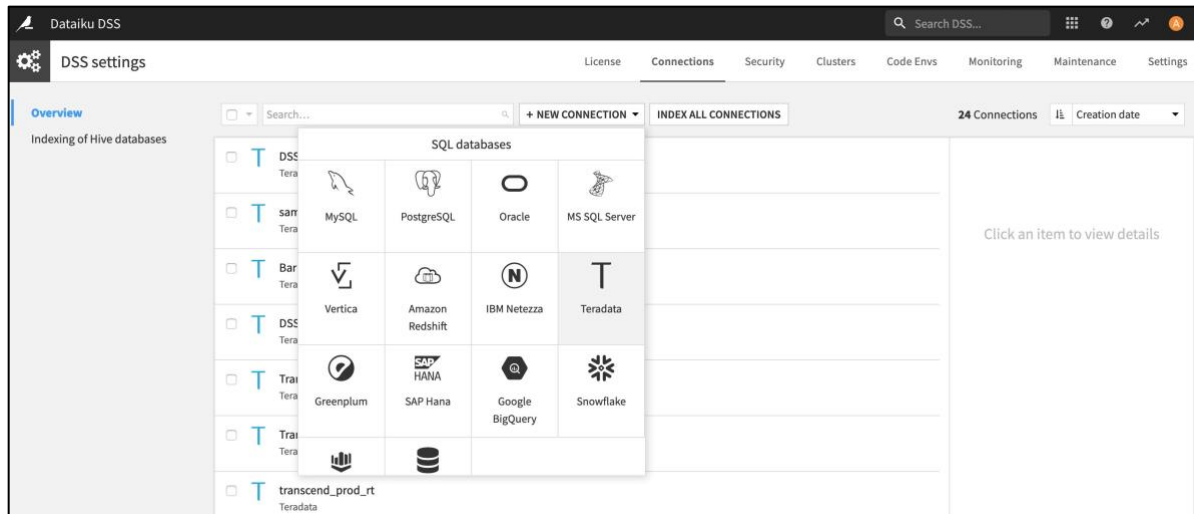


Then, on the submenu click [Administration] (gear icon).

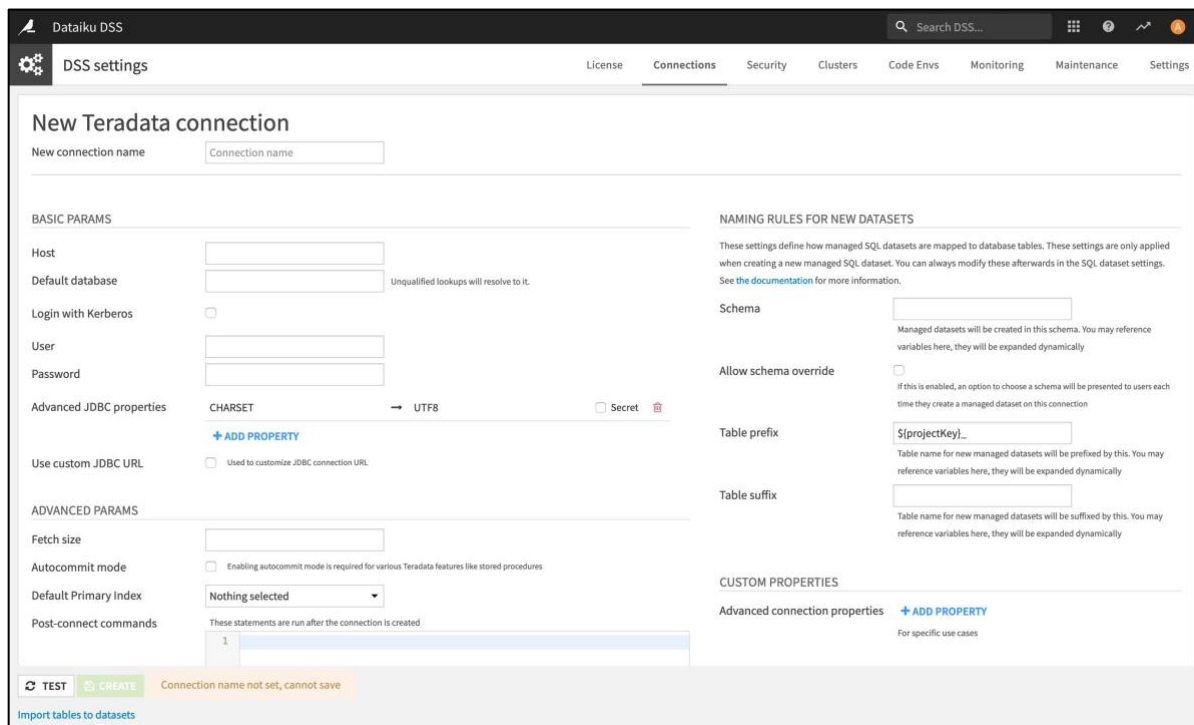


Alternatively, you can go to `http://<dataikuhost>:<port>/admin/`.

- On the DSS settings page, go to the [Connections] tab. Click on [+ NEW CONNECTION] menu button. Choose [Teradata] among the options presented.



- You are then presented with a screen to specify the characteristics of the connection to the desired Vantage system.



4.1. Fill in the fields as needed:

Basic Params Host: <database.host.name>

User: <your_database_username>

Password: <your_database_user_password>

Default Database: <default_database>

Advanced JDBC properties:

CHARSET: UTF8

TMODE: TERA

Note: If your target system connections is LDAP-based, then also specify:

LOGMECH: LDAP

Autocommit Mode: Check the button to enable the autocommit mode.

All other fields can be left as-is.

4.2. Modify “Details readable by” to either “Every Analyst” or “Selected Groups”.

SECURITY SETTINGS

See [the documentation](#) for more information.

Freely usable by

☒ Every analyst
☐ Selected groups

Who can create new datasets in this connection, and more generally "browse" this connection.

Details readable by

☐ Nobody
☒ Every analyst
☐ Selected groups

Who can access the details of the connection (including credentials, if connection has some).

4.3. Click on the [Test] button to verify that connection details provided are valid.

4.4. Finally, click on the [Save] button.

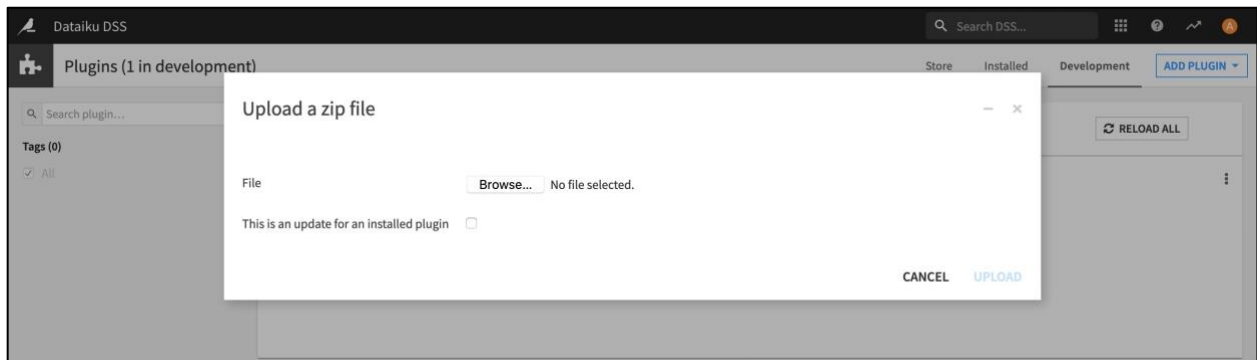
4. Plugin Download And Installation

4.1. Get The Plugin

The Teradata Plugins for Dataiku can be downloaded from the "Teradata Plugins for Dataiku" link on the downloads.teradata.com website. The plugin itself is a compressed zip file that contains the corresponding plugin software and metadata.

4.2. Plugin Installation

1. Locate in your local filesystem the zip file of the downloaded plugin you wish to install.
2. In DSS Settings page (accessible through the Admin Tools button), select the [Plugins] tab, then click the option [Upload] under the [ADD PLUGIN] menu on the top right.



3. Click on [Browse] and navigate to the location of the plugin zip file in your local filesystem.
4. If a previous installation of the plugin exists, check the button "This is an update for an installed plugin".
5. Click on [UPLOAD] button.

6. When the upload succeeds, changes may not take effect immediately.
 - In DSS Settings page (accessible through the Admin Tools button), select the [Plugins] tab, then select the [INSTALLED] tab on the resulting screen. Then, push the [RELOAD ALL] button on the top right corner of the screen.
 - Alternatively, perform a hard refresh (Ctrl + F5) on all open Dataiku browsers.

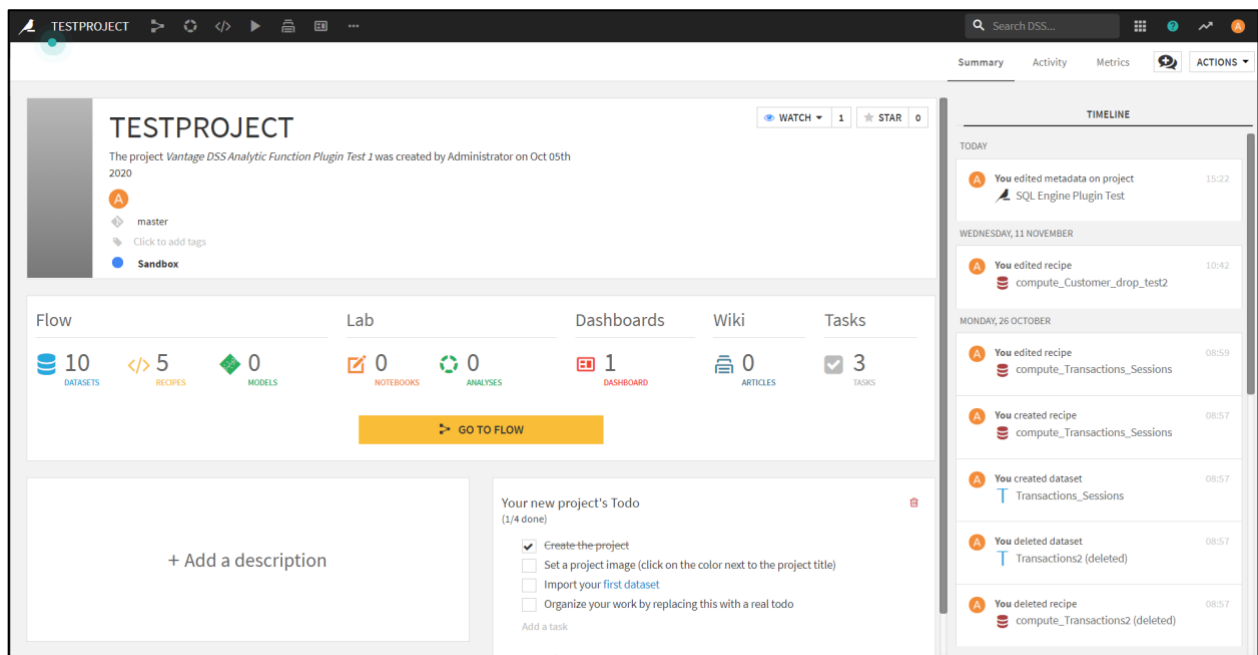
5. Using the Teradata Plugins For Dataiku

5.1. Using the Teradata Advanced SQL Engine Functions Plugin

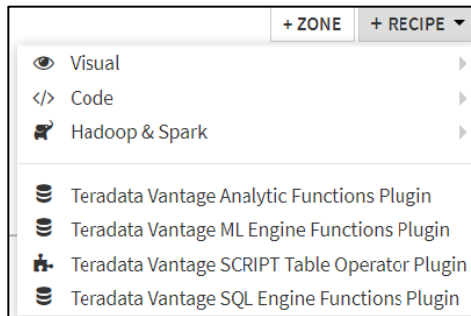
5.1.1. Instructions

This section assumes that a Dataiku DSS project already exists, and input datasets have already been imported. Note that recipes need a non-empty dataset as input to run.

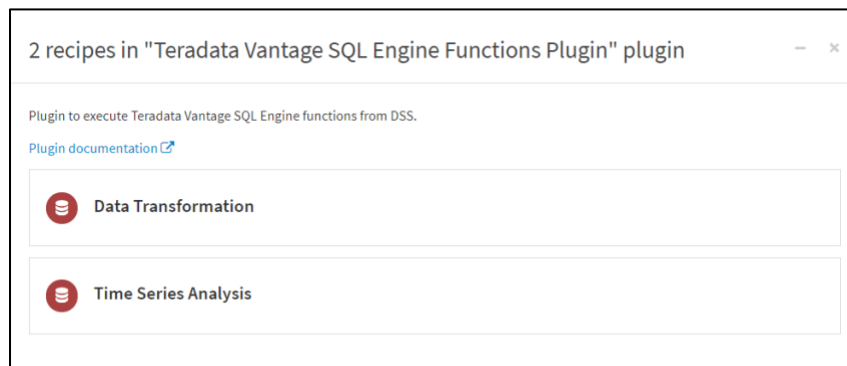
1. Go to the flow view of the DSS project, where the recipe is to be created, by clicking on the [GO TO FLOW] button, or by clicking on the flow icon in the project menu.



2. In the Flow view, click on the [+RECIPE] button, then select the [Teradata SQL Engine Functions Plugin].



Then proceed to select the desired recipe. Available recipe names correspond to the different categories Advanced SQL Engine Functions, as illustrated in the following figure.

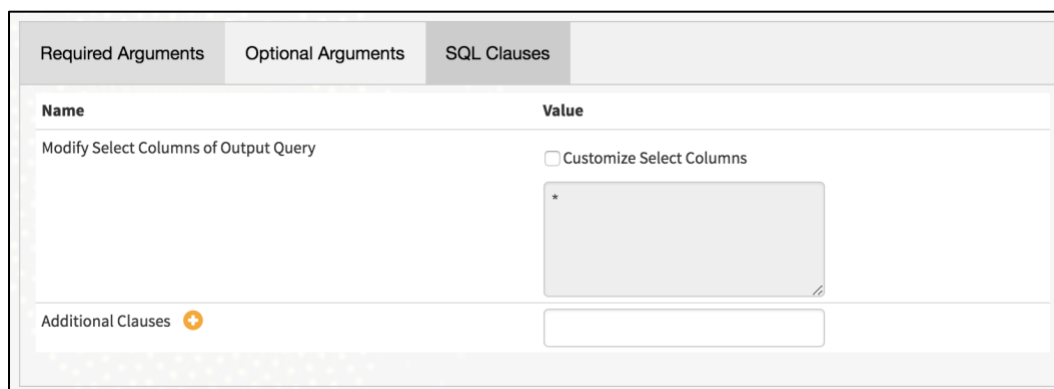


3. In the [New custom recipe] popup, specify the input and output datasets. There can be more than one input dataset, as in the case of multiple-input analytic functions. The same is also the case for Vantage functions with multiple output datasets. The output dataset will be stored in the database and schema corresponding to the connection selected in the [Store into] field. Click on [CREATE DATASET] button when done. The following figure illustrates an example for the Data Transformation recipe in the Teradata Advanced SQL Engine Functions plugin.

4. In the recipe settings, select the most suitable function for the manipulation or analysis of the input dataset. Configure the chosen analytic recipe by specifying parameters such as the input tables, partition and order attributes, and arguments. A recipe's required and optional fields are separated into different tabs.

Name	Value
Inputtable	complaints_testtoken
Modeltable	ldamodel
Outputtable	ldaout2
Docidcolumn	doc_id (int)
Wordcolumn	token (string)

5. The [SQL Clauses] tab allows you to explicitly modify the query to be executed.



The screenshot shows a tabbed interface with three tabs: "Required Arguments", "Optional Arguments", and "SQL Clauses". The "SQL Clauses" tab is active. It contains a table with two columns: "Name" and "Value". The first row has the name "Modify Select Columns of Output Query" and a value consisting of a checkbox labeled "Customize Select Columns" and a text area containing an asterisk (*). The second row has the name "Additional Clauses" followed by a plus icon (+) and a text input field.

The field next to "Modify Select Columns of Output Query" enables you to modify the SELECT clause of the query. The field next to "Additional Clauses" enables you to append additional SQL clauses to the query such as WHERE, ORDER BY, GROUP BY, and other similar clauses. These fields have equivalent effects as if the query were modified as:

```
SELECT {modified select} FROM function_name(  
    ...  
)  
    {additional clauses}
```

6. Click on the [RUN] button or save the recipe settings for later use.



5.1.2. Usage Notes

A function with multiple output datasets will typically require an output dataset for the function's output message/result, in addition to any other output tables/datasets specified in the recipe. Please note that the output dataset(s) name(s) should also match the name within the recipe's settings.

Furthermore, in the background the plugins submit SQL queries to create output tables in the connected Vantage system by using "CREATE SET TABLE". That is, the function output is checked for duplicate rows. If any duplicate rows are found, then they are removed from the output table. The check for duplicate rows adds a slight performance burden compared to using unfiltered output. This behavior is not adjustable in the present version of the plugins.

5.1.3. Troubleshooting

5.1.3.1. "External Code Failed" Error

Assume that upon attempting to load a recipe from the Teradata Advanced SQL Engine Functions Plugin, you encounter an error immediately after the plugin user interface screen appears. The error looks like the following banner:



This scenario might appear after a change occurs in the plugins registered with DSS on your server. A possible fix is to attempt reloading the plugins as follows:

1. In DSS Settings page (accessible through the Admin Tools button), select the [Plugins] tab, then select the [INSTALLED] tab on the resulting screen.
2. Push the [RELOAD ALL] button on the top right corner of the screen.

Now you can return to your project flow, and attempt to create the recipe anew.

5.2. Using the Teradata SCRIPT Table Operator Plugin

This section assumes that a Dataiku DSS project already exists and input datasets have already been imported. Note that recipes need a non-empty dataset as input to run.

There are three (3) main tabs containing arguments used to install/replace the script files on the Advanced SQL Engine Database and/or invoke the SCRIPT Table Operator (STO).

5.2.1. Script Loading

The screenshot shows the 'Script Loading' tab of a configuration window. It features two main sections: a top section for a single file and a bottom section for multiple files. The top section has a 'File Name' column and a 'File Details' column. The 'File Details' column contains a 'Script File Location' dropdown menu (currently showing '--No Selection--'), a 'Script File Alias' text input field, a 'Replace Script' checkbox, and a 'Replace Current Script' checkbox. The bottom section has an 'Add More Files' button and a message: 'Added files will be installed in the following path: ./DSSExamples/<filename>'. Below this, there is a table with columns 'File Name' and 'File Details'. The 'File Name' column has a 'Remove File' button. The 'File Details' column contains a 'File Location' dropdown menu, a 'File Alias' text input field, a 'File Format' dropdown menu, a 'Replace File' checkbox, and a 'Replace Current Script' checkbox.

- Script File Name
 - The name of the script file to be uploaded.
 - This is the main script used in the SCRIPT Table Operator.
 - Depending on the selected Script File Location this input changes:
 - If the script is on the Vantage Server – A text input field is provided to enter a file name.
 - If the script is in the DSS Managed Folders and DSS Notebooks – A drop-down box containing a list of the files under their respective locations is provided.
 - The Script File Name will not appear until the Script File Location is selected.
- Script File Location
 - The location of the script to be installed, either on the Vantage server, a DSS Jupyter Notebook, or a DSS Managed Folder

- Script File Alias
 - The file alias to be used in the SQL statement
 - This is mainly used by the SCRIPT Installation/Replace process in the metadata tables.
- Script File Address
 - The fully qualified file location on the Vantage Server
 - This only appears if the selected option for Script File Location is "Vantage Server"
- Add More Files
 - This button allows the user to have additional files installed in the Vantage Advanced SQL Engine.
 - There is a file path specified to the right of the button in which the additional files are installed.
 - This may normally be used in instances where the user's main script references an additional file.
- Additional Files:
 - File Name
 - This is the file name of an additional file.
 - Similar to the Script File Name it is a Text Field for files located in the Vantage Advanced SQL Engine and a drop-down box if DSS Managed Folder is selected as the File Location
 - File Location
 - The location of the file to be installed, either on the Vantage server or a DSS Managed Folder
 - File Address
 - The fully qualified file location on the Vantage server
 - Similar to the Script File Address this only appears when "Vantage Server" is selected as the file location.
 - File Format
 - Specifies whether the additional file to be installed is a BINARY or TEXT file.

5.2.2. SCRIPT Table Operator Arguments

Clause/Argument	Value
Script Type	Other
Script Command	Type script command here
Script Arguments	
ON	SELECT * FROM ex2tbl_tera
	<input type="checkbox"/> Customize the ON clause
HASH BY	
PARTITION BY	
ORDER BY	
LOCAL ORDER BY	
RETURNS	

- Script Type
 - The type of script to be used typically Python or R'
 - Script Command
 - This is a Text area where the user can enter a custom Script Command.
 - This argument only appears if the selected Script type is "Other".
- Script Arguments
 - The arguments for the script, place one argument per box. Click on the (+) button to add more arguments'
- ON
 - The ON Clause used as the input data for the script
 - If UNMODIFIED the clause defaults to *"SELECT * FROM {input_table}"*
- Customize the ON clause
 - A checkbox which specifies whether the ON clause should be modified.
- HASH BY
 - A HASH BY clause will cause the rows in the ON clause to be redistributed to AMPs based on the hash value of the column(s) specified'
- PARTITION BY
 - A PARTITION BY clause will cause the STO to execute against specific groups (partitions) based on the column(s) specified
- ORDER BY
 - 'An ORDER BY clause specifies the order in which values in a group (partition) are sorted

- **LOCAL ORDER BY**
 - A LOCAL ORDER BY clause orders the rows qualified on each AMP
- **RETURNS**
 - **RETURNS NAME**
 - The first column under returns
 - Specifies the name of the column(s) to be returned by the STO'
 - **RETURNS TYPE**
 - The second column under returns
 - Specifies the data type of the column(s) to be returned by the STO

5.2.3. Other SQL Arguments

- **Select Columns**
 - Specifies the contents of a user customized SELECT statement (data to be returned by the query)
 - Default is to SELECT all column(s) in the RETURNS clause
- **Customize Select Columns Checkbox**
 - Determines whether the SELECT (output) columns (data to be returned by the query) should be modified.
- **Additional Clauses**
 - Specifies any additional clauses to the output such as a HAVING or QUALIFY clause

5.2.4. Running the Teradata SCRIPT Table Operator Plugin

After setting up the arguments, click on the [RUN] button to run the SCRIPT Table Operator.

