How-To Library

Informatica

Configuring Pushdown Optimization by Using Google BigQuery ODBC Driver

© Copyright Informatica LLC 2019, 2021. Informatica, the Informatica logo, and Informatica Cloud are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at https://www.informatica.com/trademarks.html.

Abstract

You can use pushdown optimization to push transformation logic to source databases or target databases. Use pushdown optimization when you use database resources to improve the performance of the task. When you run a task configured for pushdown optimization, the task converts the transformation logic to an SQL query. The task sends the query to the database, and the database executes the query.

This article explains how to use pushdown optimization for the ODBC connection type that uses Google BigQuery ODBC drivers for mappings. This article also explains the supported pushdown optimization functions, transformations, operators, and data types.

Supported Versions

- Informatica Cloud® Data Integration Fall 2020

Table of Contents

Overview

You can use Google BigQuery Connector to read data from Google BigQuery. To read large amount of data, you can use source or full pushdown for the ODBC connection type. Using the ODBC connection type with pushdown optimization enhances the performance of the task.

For example, consider you are a sales manager in a rapidly growing manufacturing organization. Your organization stores the product transaction details such as transactionID, customerID, productID, quantity, product_revenue, and OrderDate in Google BigQuery. You need to calculate the total revenue generated from the sales of a particular product. Use Google BigQuery Connector to create a mapping to read all the product revenue details of a particular product

from the Google BigQuery source, apply aggregate function to calculate the total revenue, and write the records to Google BigQuery target for data analysis.

The organization plans to implement a business intelligence service to build visualization and perform real-time analysis. Therefore, you need to port the vast amount of data stored in Google BigQuery database to the business intelligence service. To read this large amount of data, you can use full pushdown for the ODBC connection type. Using the ODBC connection type with pushdown optimization enhances the performance of the task.

To read data from a Google BigQuery source using the ODBC connection, perform the following steps:

1. Configure a Google BigQuery ODBC connection.
2. Create an ODBC connection.
3. Create a mapping.
4. Create a mapping task.

Note: When you use pushdown optimization to write data to a Google BigQuery target, you cannot perform update, upsert, or delete operation on a Google BigQuery target.

Prerequisites

Before you create an ODBC connection to connect to Google BigQuery, perform the following prerequisite tasks:

1. Ensure that you have a Google service account to access Google BigQuery.
2. Download the .p12 or JSON key file for the service account. To generate a .p12 or JSON key file for the service account, click the following URL:
   https://cloud.google.com/iam/docs/creating-managing-service-account-keys
3. When you run a mapping task for full pushdown optimization, you must have the following permissions:
   - bigquery.datasets.get
   - bigquery.datasets.getIamPolicy
   - bigquery.models.getData
   - bigquery.models.getMetadata
   - bigquery.models.list
   - bigquery.routines.get
   - bigquery.routines.list
   - bigquery.tables.create
   - bigquery.tables.delete
   - bigquery.tables.export
   - bigquery.tables.get
   - bigquery.tables.getData
   - bigquery.tables.getIamPolicy
   - bigquery.tables.list
   - resourcemanager.projects.get
   - resourcemanager.projects.list
   - bigquery.jobs.create

   Note: When you configure the **Allow Temporary View for Pushdown** in the mapping task, you must have the bigquery.tables.create and bigquery.tables.delete to permission create and drop views.

The dataset configured to create and drop views is used as the default dataset in the Google BigQuery ODBC driver.

Configuring a Google BigQuery ODBC connection

You can set the pushdown optimization for the ODBC connection type that uses Google BigQuery ODBC driver to enhance the mapping performance. To use an ODBC connection to connect to Google BigQuery, you must configure the ODBC connection.

After you create a Google BigQuery ODBC connection, add the **Pushdown Optimization** property under **Advanced Session Properties** tab when you create a mapping task and specify **Full** or **Source** in the **Session Property Value** field.

You cannot configure target-side pushdown optimization by using Google BigQuery ODBC driver. To verify that the pushdown optimization occurred successfully, you can check the session log for the job. In Monitor, view the log for jobs.

Google BigQuery supports Google BigQuery ODBC drivers on Windows and Linux systems. You can install one of the following 64-bit drivers based on your system requirement:

- 2.2.5.1012 version of the Informatica ODBC Driver for Google BigQuery
- 2.1.19.1024 version of the Simba ODBC Driver for Google BigQuery

Configuring a Google BigQuery ODBC connection using Informatica ODBC Driver for Google BigQuery

You can download the Informatica ODBC Driver for Google BigQuery from the following location:

Google BigQuery supports Informatica ODBC Driver for Google BigQuery on Windows and Linux systems. You can install the 2.2.5.1012 version of the Informatica ODBC Driver for Google BigQuery based on your system requirement.

For more information about downloading the Informatica ODBC Driver for Google BigQuery, contact Informatica Global Customer Support.

Configuring Google BigQuery ODBC connection on Windows

To establish an ODBC connection to connect to Google BigQuery on Windows using the Informatica ODBC Driver for Google BigQuery, install the Informatica ODBC driver for Google BigQuery, version 2.2.5.1012, on the Windows machine where the Secure Agent runs and configure the ODBC connection.

1. Download the `InformaticaODBCDriverforGoogleBigQuery_2.2.5.1012_Windows.zip` file, `Register.bat` file, and the `DriverInstallHelper.jar` file from the following location:

   `<Driver Location in the SFTP server>`
   For more information about downloading the drivers, contact Informatica Global Customer Support.

   Note: Download the `Register.bat` and `DriverInstallHelper.jar` files in the same location.
2. After you download the Informatica Google BigQuery ODBC (64-bit) driver, extract the downloaded file.
3. Navigate to the following directory where you extracted the `InformaticaODBCDriverforGoogleBigQuery_2.2.5.1012_Windows.zip` file:

   `<Informatica ODBC Driver for Google BigQuery installation directory>`
   `\InformaticaODBCDriverforGoogleBigQuery_2.2.5.1012_Windows`
4. Extract the `SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012.zip` file to the following directory.

SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012.zip file:
<SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>
\SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012\

5. Navigate to the directory where you downloaded the Register.bat and DriverInstallHelper.jar file.

   The Register.bat file prompts you to specify the **Input Installation Directory**. Specify the directory where you installed the Informatica ODBC Driver for Google BigQuery.

6. Open the Command Prompt as an administrator and navigate to the directory where you downloaded the Register.bat and DriverInstallHelper.jar file.

7. Run the following command in the command prompt: Register.bat

   Note: You must have administrator privileges to run the Register.bat file.

   The Register.bat file prompts you to specify the **Input Directory Containing Registry File Template** and **Input Directory Containing Installation Files**.

8. Specify the following directory for the **Input Directory Containing Registry File Template** and press **Enter**:

   <Informatica ODBC Driver for Google BigQuery installation directory>
   \InformaticaODBCDriverforGoogleBigQuery_2.2.5.1012_Windows\setup

9. Specify the following directory for the **Input Directory Containing Installation Files** and press **Enter**:

   <SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>
   \SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012\
   The Register.bat file creates the output.txt file and returns success or an error message.

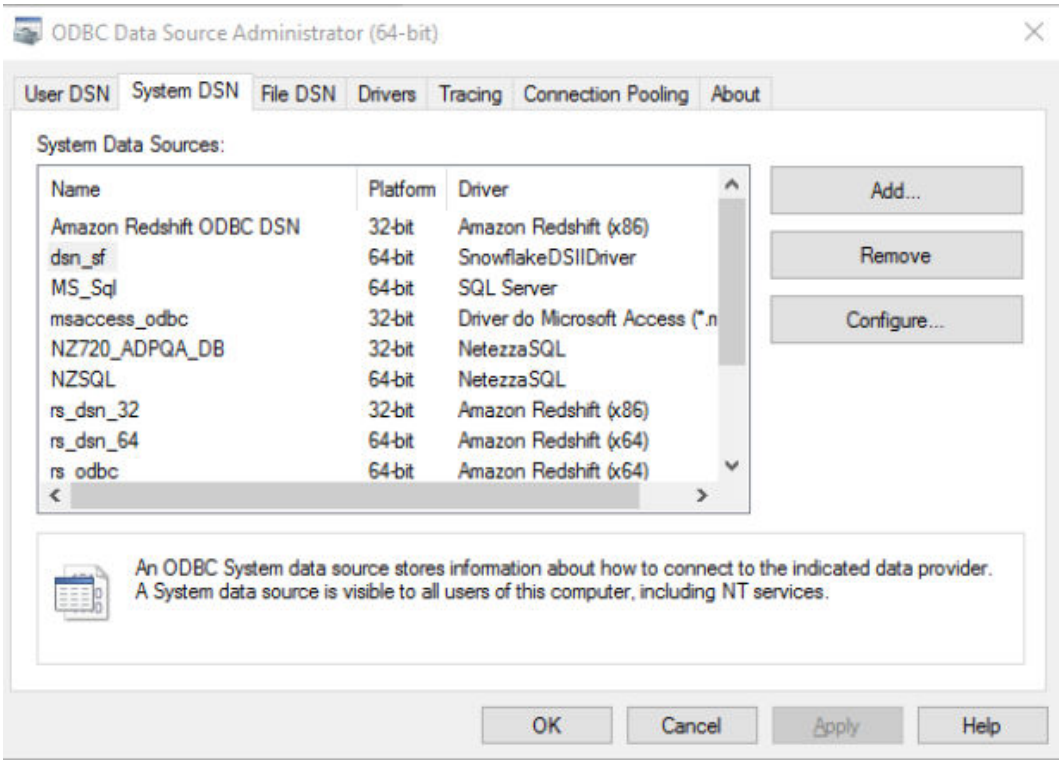
10. Click **Start** > **Control Panel**.
11. Click **Administrative Tools**.
12. Click **Data Sources (ODBC)**.

    The **ODBC Data Source Administrator** dialog box appears.

13. Click the **System DSN** tab.

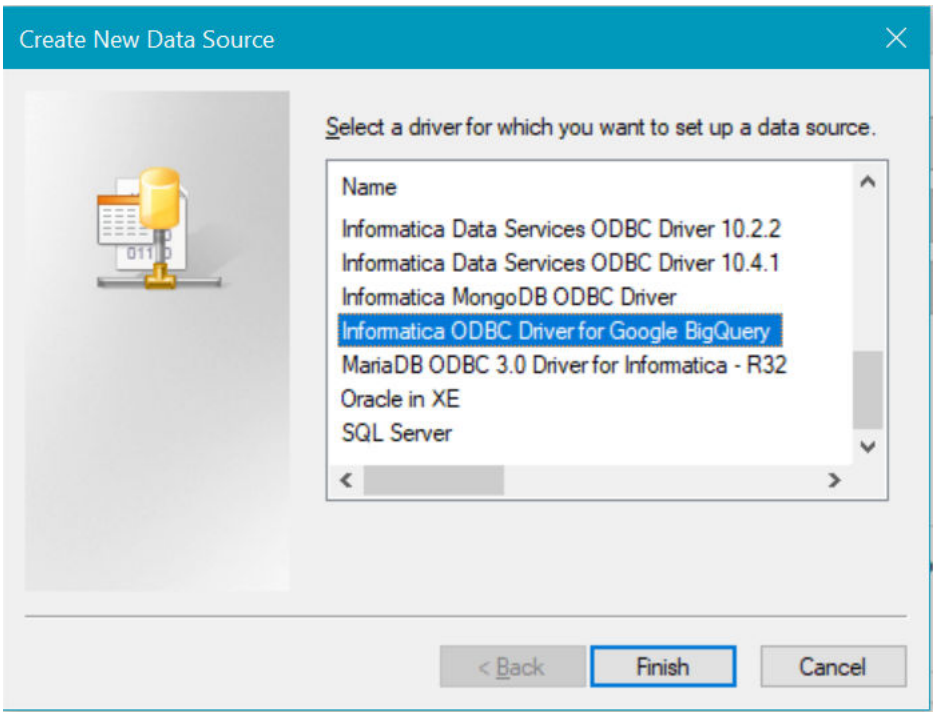
    The **System DSN** tab appears.

The following image shows the System DSN tab on the ODBC Data Source Administrator (64-bit) dialog box:



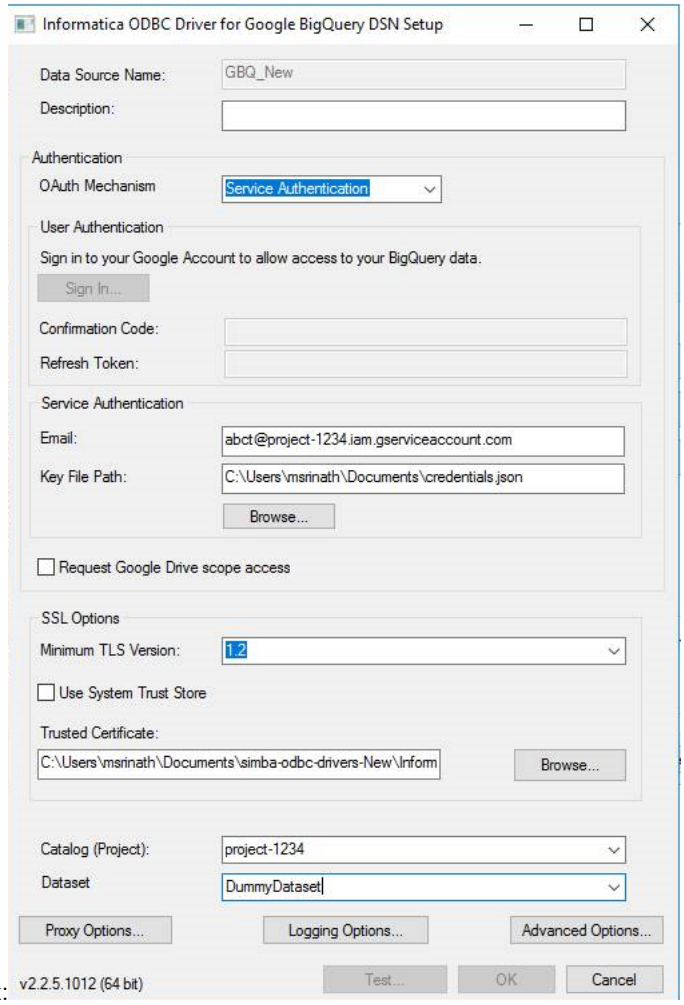
14. Click **Add**.

    The **Create New Data Source** dialog appears.
    The following image shows the Create New Data Source dialog where you can select the Google BigQuery data source:



15. Select the **Informatica ODBC Driver for Google BigQuery** to set up the data source.
16. Click **Finish**.

    The **Informatica ODBC Driver for Google BigQuery DSN Setup** dialog box appears.
    The following image shows the connection properties in the **Informatica ODBC Driver for Google BigQuery DSN Setup** dialog box:



17. Specify the following connection properties:

Property	Description
Data Source Name	Enter a name for the data source. The ODBC Driver for Google BigQuery uses the DSN to connect to the Google BigQuery.
Description	Enter a description.

Property	Description
OAuth Mechanism	The OAuth 2.0 authentication mechanism used to authenticate the driver. Select the **Service Authentication** option to authenticate the driver through a Google service account. If you select the **User Authentication** option, you will need to sign in to your Google service account. Click on **Sign In** and in the browser that opens, enter your credentials and then click **Sign In**. Click **Accept** to allow the Google BigQuery Client Tools to access Google BigQuery objects.
Confirmation Code	Code that Google provides when you click **Accept**. Copy and paste the code in the **Confirmation Code** field.
Refresh Token	The OAuth Mechanism populates the **Refresh Token** field when you paste the confirmation code.
Email	Specify the Google service account email ID. This field is needed to authenticate the service account.
Key File Path	Enter the path to the .p12 or JSON key file that is used to authenticate the Google service account.
Request Google Drive scope access	Allows the driver to access Google Drive so that the driver can support federated tables that combine Google BigQuery data with data from Google Drive.
Trusted Certificates	Path of the `.pem` file. Use the trusted CA certificates from a specific `.pem` file or use the trusted CA certificates `.pem` file that is installed with the driver. **Note:** If you specify the Trusted Certificates .pem file path, you do not need to select the **Use System Trust Store** option.
Use System Trust Store	If you select the **Use System Trust Store** option, you do not need to specify the Trusted Certificates `.pem` file path.
Catalog (Project)	Name of the Google BigQuery project associated with your billing account that the Simba ODBC Driver for Google BigQuery queries against.

Configuring existing data sources on Windows to use Informatica ODBC Driver for Google BigQuery

If you have created a data source using the 2.1.19.1024 version of the Simba ODBC Driver for Google BigQuery and you want the same data source to use the 2.2.5.1012 version of the Informatica ODBC Driver for Google BigQuery on a Windows machine, perform the following steps after you configure the Informatica ODBC Driver for Google BigQuery:

1. Open the Windows Registry Editor.
2. Navigate to following location on the Registry Editor:
   `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\ODBC Data Sources`
3. Select the existing DSN and modify the registry value from Simba ODBC Driver for Google BigQuery to Informatica ODBC Driver for Google BigQuery.
4. Navigate to following location on the Registry Editor:
   `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\<OLD DSN KEY>`
5. Modify the value of the **Driver** registry from Simba ODBC Driver for Google BigQuery to Informatica ODBC Driver for Google BigQuery.
6. Modify the value of the **TrustedCerts** registry to the following location:
   `<SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>/lib/cacerts.pem`

Configuring Google BigQuery ODBC connection on Linux

To establish an ODBC connection to connect to Google BigQuery on Linux using the Informatica ODBC Driver for Google BigQuery, install the Informatica Google BigQuery ODBC (64-bit) driver, version 2.2.5.1012, on the Linux machine where the Secure Agent runs and configure the ODBC connection.

1. Download the `InformaticaODBCDriverforGoogleBigQuery_2.2.5.1012-Linux.tar.gz` file from the following location:

   `<Driver Location in the SFTP server>`
   For more information about downloading the drivers, contact Informatica Global Customer Support.
2. After you download the Informatica Google BigQuery ODBC (64-bit) driver, use the following command to extract the downloaded file:

   ```
   tar -xvf InformaticaODBCDriverforGoogleBigQuery_2.2.5.1012-Linux.tar.gz
   ```
3. Navigate to the following directory where you extracted the `InformaticaODBCDriverforGoogleBigQuery_2.2.5.1012-Linux.tar.gz` file:

   `<Informatica ODBC Driver for Google BigQuery installation directory>/InformaticaODBCDriverforGoogleBigQuery_2.2.5.1012-Linux`
4. Use the following command to extract the `SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012.tar.gz` file to any directory:

   ```
   tar -xvf SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012.tar.gz
   ```
   Consider the following directory where you extracted the `SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012.tar.gz` file:
   `<SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>/SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012`
5. Move the `GoogleBigQueryODBC.did` file from the `<Informatica ODBC Driver for Google BigQuery installation directory>/InformaticaODBCDriverforGoogleBigQuery_2.2.5.1012-Linux` folder to the `<SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>/SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012/lib` folder.
6. Move the `simba.googlebigqueryodbc.ini` file from the `<Informatica ODBC Driver for Google BigQuery installation directory>/InformaticaODBCDriverforGoogleBigQuery_2.2.5.1012-Linux/setup` folder to the `<SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>/SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012/lib` folder.
7. Navigate to the `<SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>/SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012/lib` folder and rename the `simba.googlebigqueryodbc.ini` file to `informatica.googlebigqueryodbc.ini`.
8. Edit the `informatica.googlebigqueryodbc.ini` file with the following:
   - Change the value for the `DriverManagerEncoding` property from UTF-32 to UTF-16.
   - Set the `<INSTALLDIR>` property to the following directory:
     `<SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>/SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012`
9. Create the `odbc.ini` file and add the following properties:

   ```
   [ODBC Data Sources]
   GBQ_ODBC=Simba ODBC Driver for Google BigQuery 64-bit
   Description=<DSN Description>
   [Sample DSN]
   Driver=<SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>/lib/
   libgooglebigqueryodbc_sb64.so
   Catalog=<project_id>
   DefaultDataset=<Dataset name>
   SQLDialect=<1>
   ```

```
OAuthMechanism=0
Email=<Client Email ID of the Google service account>
KeyFilePath=<.p12 or JSON key file path>
```

For example,

```
[ODBC Data Sources]
GBQ_ODBC=Simba ODBC Driver for Google BigQuery 64-bit
[GBQ_ODBC_IICS]
Driver=<SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>/lib/
libgooglebigqueryodbc_sb64.so
Catalog=api-project-1243343
DefaultDataset=QATEST
SQLDialect=1
OAuthMechanism=0
Email=simbaaccount@api-project-1243343.iam.gserviceaccount.com
KeyFilePath=/export/Simba_GBQ_ODBC/API Project-c993e990af5.json
```

10. Specify the following properties in the `odbc.ini` file:

Property	Description
ODBC Data Sources	Name of the data source.
Description	Description of the data source.
Driver	Full path of the Simba ODBC Driver for Google BigQuery library file.
Catalog	Name of the Google BigQuery project associated with your billing account that the Simba ODBC Driver for Google BigQuery queries against.
SQLDialect	The SQL dialect used to run queries against the Google BigQuery tables using the DSN. To perform pushdown optimization, specify the value of **SQLDialect** property as 1.
DefaultDataset	Specify a dataset name in Google BigQuery that the Simba ODBC driver queries by default.

OAuthMechanism	The OAuth 2.0 authentication mechanism used to authenticate the driver.
	To authenticate the driver through a Google service account, specify the value of **OAuthMechanism** property as **0**.
	Specify the value of the **Email** property to the Google service account email ID.
	To authenticate the driver through a Google user account, specify the value of **OAuthMechanism** property as **1**.
	Obtain a **Refresh Token** based on your Google user account and set the following property in the DSN:
	`Auth_RefreshToken=<Refresh token value>`
	If you already have your refresh token, then you can set the following property in the DSN:
	`RefreshToken=<Refresh token value>`

Email	Applicable when you set the value of **OAuthMechanism** property as **0**. Specify the value of the **Email** property to the Google service account email ID.
KeyFilePath	Applicable when you set the value of **OAuthMechanism** property as **0**. Enter the path to the .p12 or JSON key file that is used to authenticate the Google service account.
RefreshToken	Applicable when you set the value of **OAuthMechanism** property as **1**. Specify the refresh token associated with the Google user account.

11. Run the following command to set the environment variable ODBCINI:

```
Export ODBCINI=/<odbc.ini file path>/odbc.ini
```

12. Restart the Secure Agent.

The Google BigQuery ODBC connection on Linux is configured successfully.

After you configure the Google BigQuery ODBC connection, you must create an ODBC connection to connect to Google BigQuery.

For more information about how to create an ODBC connection to connect to Google BigQuery, see "Create an ODBC connection" on page 17.

Configuring existing data sources on Linux to use Informatica ODBC Driver for Google BigQuery

If you have created a data source using the 2.1.19.1024 version of the Simba ODBC Driver for Google BigQuery and you want the same data source to use the 2.2.5.1012 version of the Informatica ODBC Driver for Google BigQuery on a Linux machine, perform the following steps after you configure the Informatica ODBC Driver for Google BigQuery:

1. Edit the `odbc.ini` file and update the Driver property for the DSN created with the 2.1.19.1024 version of the Simba ODBC Driver for Google BigQuery and change the value to the location of the 2.2.5.1012 version of the Informatica ODBC Driver for Google BigQuery.
   For example, set the following value for the Driver property:

   ```
   <SimbaODBCDriverforGoogleBigQuery64_2.2.5.1012 installation directory>/lib/
   libgooglebigqueryodbc_sb64.so
   ```

2. After you configure the `odbc.ini` file, you must create an ODBC connection using the same data source in Data Integration. You must ensure that the test connection is successful.

Configuring a Google BigQuery ODBC connection using Simba ODBC Driver for Google BigQuery

You can download the 2.1.19.1024 version of the Simba ODBC Driver with SQL Connector for Google BigQuery from your Simba account for Windows or Linux 64-bit operating system.

Google BigQuery supports ODBC Driver for Google BigQuery on Windows and Linux systems. You must install the 2.1.19.1024 version of the Simba ODBC Driver for Google BigQuery 64-bit driver based on your system requirement.

Configuring a Google BigQuery ODBC connection on Windows

To establish an ODBC connection to connect to Google BigQuery on Windows, you must download the 64-bit Google BigQuery ODBC driver on the machine where Secure Agent runs.

For more information about installing the Simba ODBC Driver for Google BigQuery on Windows, see the *Simba ODBC Driver for Google BigQuery Install and Configuration Guide* located in the following directory:

```
<SimbaODBCDriverforGoogleBigQuery64_2.1.20.1025 installation directory>\Simba ODBC Driver for Google
BigQuery
```

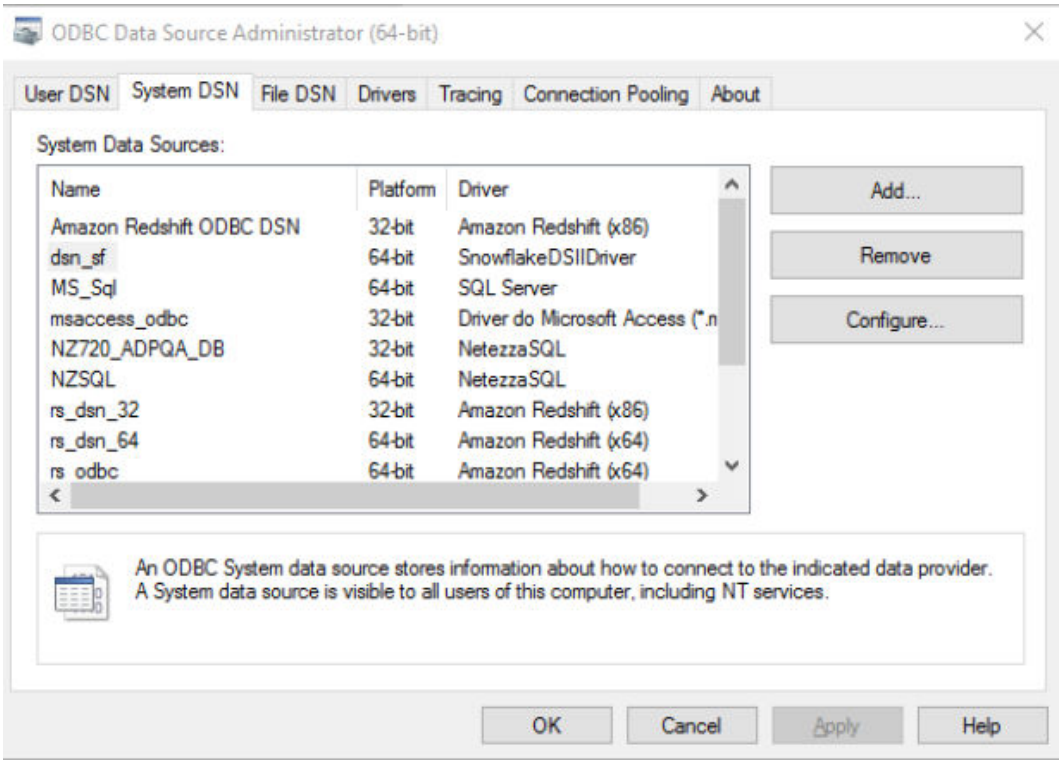
Perform the following steps to configure an ODBC connection on Windows:

1. Click **Start** > **Control Panel**.
2. Click **Administrative Tools**.
3. Click **Data Sources (ODBC)**.

   The **ODBC Data Source Administrator** dialog box appears.

4. Click the **System DSN** tab.

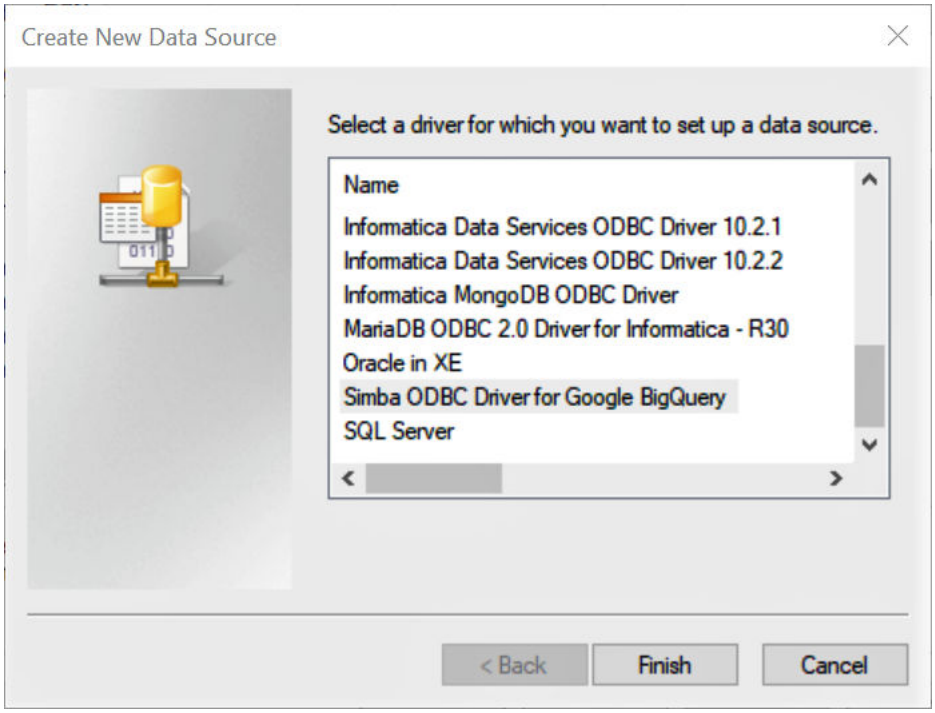
   The **System DSN** tab appears.
   The following image shows the System DSN tab on the ODBC Data Source Administrator (64-bit) dialog box:



5. Click **Add**.

   The **Create New Data Source** dialog appears.

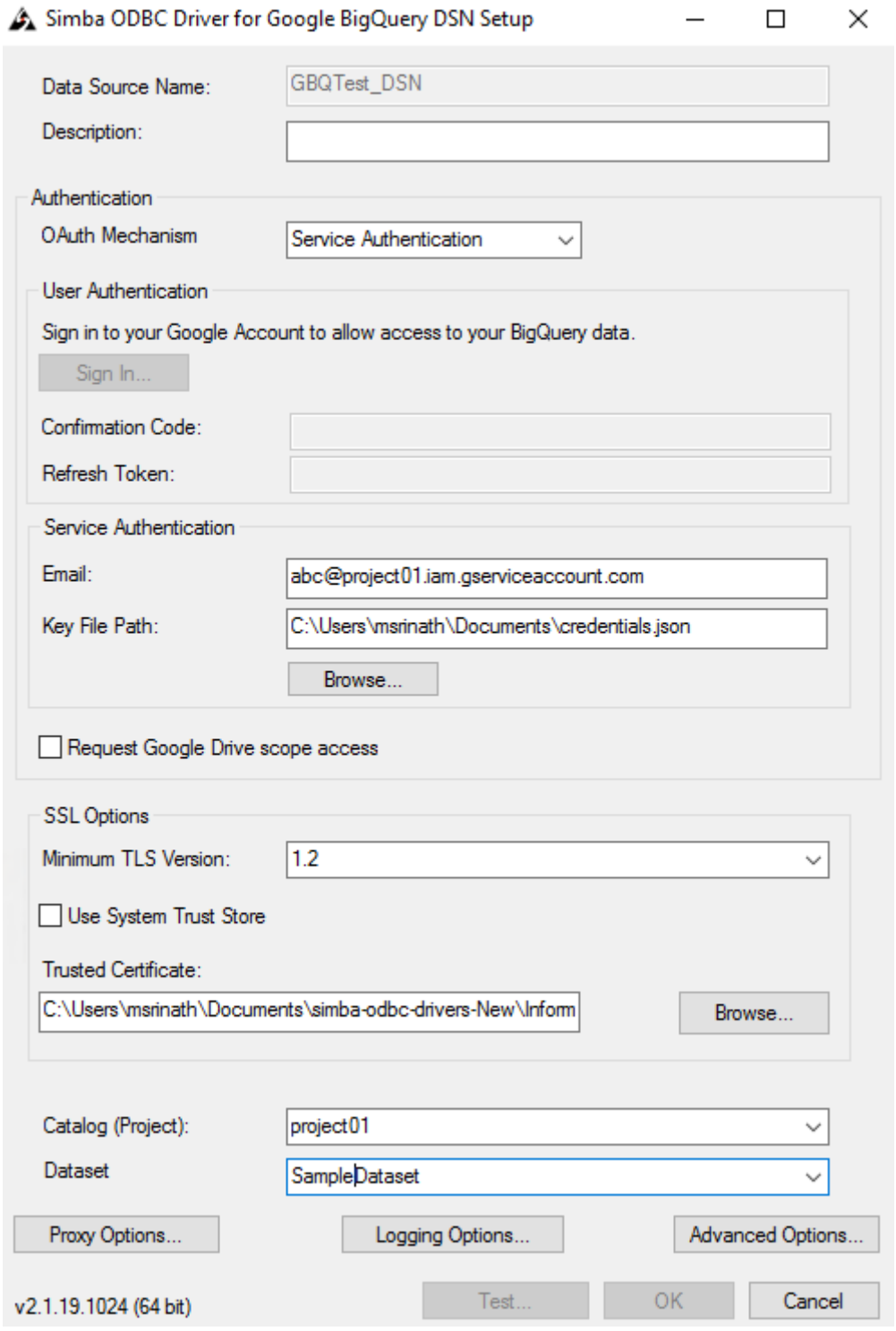
The following image shows the Create New Data Source dialog where you can select the Google BigQuery data source:



6. Select the **Simba ODBC Driver for Google BigQuery** to set up the data source.
7. Click **Finish**.

   The **Simba ODBC Driver for Google BigQuery DSN Setup** dialog box appears.

The following image shows the connection properties in the **Simba ODBC Driver for Google BigQuery DSN Setup** dialog box:



8. Specify the following connection properties:

Property	Description
Data Source Name	Enter a name for the data source. The ODBC Driver for Google BigQuery uses the DSN to connect to the Google BigQuery.
Description	Enter a description.
OAuth Mechanism	The OAuth 2.0 authentication mechanism used to authenticate the driver. Select the **Service Authentication** option to authenticate the driver through a Google service account. If you select the **User Authentication** option, you will need to sign in to your Google service account. Click on **Sign In** and in the browser that opens, enter your credentials and then click **Sign In**. Click **Accept** to allow the Google BigQuery Client Tools to access Google BigQuery objects.
Confirmation Code	Code that Google provides when you click **Accept**. Copy and paste the code in the **Confirmation Code** field.
Refresh Token	The OAuth Mechanism populates the **Refresh Token** field when you paste the confirmation code.
Email	Specify the Google service account email ID. This field is needed to authenticate the service account.
Key File Path	Enter the path to the .p12 or JSON key file that is used to authenticate the Google service account.
Request Google Drive scope access	Allows the driver to access Google Drive so that the driver can support federated tables that combine Google BigQuery data with data from Google Drive.
Trusted Certificates	Path of the `.pem` file. Use the trusted CA certificates from a specific `.pem` file or use the trusted CA certificates `.pem` file that is installed with the driver. **Note:** If you specify the Trusted Certificates .pem file path, you do not need to select the **Use System Trust Store** option.
Use System Trust Store	If you select the **Use System Trust Store** option, you do not need to specify the Trusted Certificates `.pem` file path.
Catalog (Project)	Name of the Google BigQuery project associated with your billing account that the Simba ODBC Driver for Google BigQuery queries against.

9. Click **Proxy Options** to configure the connection to a data source through a proxy server.
10. Click **Advanced Options** to configure the Simba ODBC Driver for Google BigQuery advanced properties. To perform pushdown optimization, select **Standard SQL** from the **Language Dialect** option in the **Advanced Options** dialog box.
11. To enable logging, click **Logging Options**.
12. Click **Test** to verify the connection to Google BigQuery.
13. Click **OK** to close the **Simba ODBC Driver for Google BigQuery DSN Setup** dialog box.
14. Click **OK** to close the **ODBC Data Source Administrator** dialog box

The Google BigQuery ODBC connection is configured successfully on Windows.

After you configure the Google BigQuery ODBC connection, you must create an ODBC connection to connect to Google BigQuery. For more information about how to create an ODBC connection to connect to Google BigQuery, see "Create an ODBC connection" on page 17.

Configuring a Google BigQuery ODBC connection on Linux

To establish an ODBC connection to connect to Google BigQuery on Linux, install the Google BigQuery ODBC (64-bit) driver on the Linux machine where the Secure Agent runs and configure the ODBC connection.

For more information about installing the Simba ODBC Driver for Google BigQuery on Linux, see the *Simba ODBC Driver for Google BigQuery Install and Configuration Guide* located in the following directory:

```
<SimbaODBCDriverforGoogleBigQuery64_2.1.20.1025 installation directory>\simba\googlebigqueryodbc
```

Perform the following steps to configure an ODBC connection on Linux:

1. Open the `simba.googlebigqueryodbc.ini` configuration file located in the following directory:

   ```
   <Driver installation directory>/simba/googlebigqueryodbc/lib/64/
   ```

2. Add the following property

   ```
   DriverManagerEncoding=UTF-16
   ```

3. Save the `simba.googlebigqueryodbc.ini` configuration file.
4. Copy the `simba.googlebigqueryodbc.ini` configuration file to the following directory:

   ```
   <SimbaODBCDriverforGoogleBigQuery64_2.1.20.1025 installation directory>/simba/Setup
   ```

5. Create the `odbc.ini` file and add the following properties:

   ```
   [ODBC Data Sources]
   Sample DSN=Simba ODBC Driver for Google BigQuery 64-bit
   Description=<DSN Description>
   [Sample DSN]
   Driver=<SimbaODBCDriverforGoogleBigQuery64_2.1.20.1025 installation directory>/simba/
   googlebigqueryodbc/lib/64/libgooglebigqueryodbc_sb64.so
   Catalog=<project_id>
   DefaultDataset=<Dataset name>
   SQLDialect=<1>
   OAuthMechanism=0
   Email=<Client Email ID of the Google service account>
   KeyFilePath=<.p12 or JSON key file path>
   ```

   For example,

   ```
   [ODBC Data Sources]
   GBQ_ODBC=Simba ODBC Driver for Google BigQuery 64-bit
   [GBQ_ODBC]
   Driver=/export/googlebigqueryodbc/lib/64/libgooglebigqueryodbc_sb64.so
   Catalog=api-project-1243343
   DefaultDataset=QATEST
   SQLDialect=1
   OAuthMechanism=0
   Email=simbaaccount@api-project-1243343.iam.gserviceaccount.com
   KeyFilePath=/export/Simba_GBQ_ODBC/API Project-c993e990af5.json
   ```

6. Specify the following properties in the `odbc.ini` file:

Property	Description
ODBC Data Sources	Name of the data source.
Description	Description of the data source.
Driver	Full path of the Simba ODBC Driver for Google BigQuery library file.
Catalog	Name of the Google BigQuery project associated with your billing account that the Simba ODBC Driver for Google BigQuery queries against.

Property	Description
SQLDialect	The SQL dialect used to run queries against the Google BigQuery tables using the DSN. To perform pushdown optimization, specify the value of **SQLDialect** property as 1.
DefaultDataset	Specify a dataset name in Google BigQuery that the Simba ODBC driver queries by default.

OAuthMechanism	The OAuth 2.0 authentication mechanism used to authenticate the driver.
	To authenticate the driver through a Google service account, specify the value of **OAuthMechanism** property as **0**.
	Specify the value of the **Email** property to the Google service account email ID.
	To authenticate the driver through a Google user account, specify the value of **OAuthMechanism** property as **1**.
	Obtain a **Refresh Token** based on your Google user account and set the following property in the DSN:
	`Auth_RefreshToken=<Refresh token value>`
	If you already have your refresh token, then you can set the following property in the DSN:
	`RefreshToken=<Refresh token value>`

Email	Applicable when you set the value of **OAuthMechanism** property as **0**. Specify the value of the **Email** property to the Google service account email ID.
KeyFilePath	Applicable when you set the value of **OAuthMechanism** property as **0**. Enter the path to the .p12 or JSON key file that is used to authenticate the Google service account.
RefreshToken	Applicable when you set the value of **OAuthMechanism** property as **1**. Specify the refresh token associated with the Google user account.

7. Run the following command to set the environment variable ODBCINI:

```
Export ODBCINI=/<odbc.ini file path>/odbc.ini
```

8. Restart the Secure Agent.

The Google BigQuery ODBC connection on Linux is configured successfully.

After you configure the Google BigQuery ODBC connection, you must create an ODBC connection to connect to Google BigQuery.

For more information about how to create an ODBC connection to connect to Google BigQuery, see "Create an ODBC connection" on page 17.

Create an ODBC connection

You must create an ODBC connection to connect to Google BigQuery after you configure the ODBC connection.

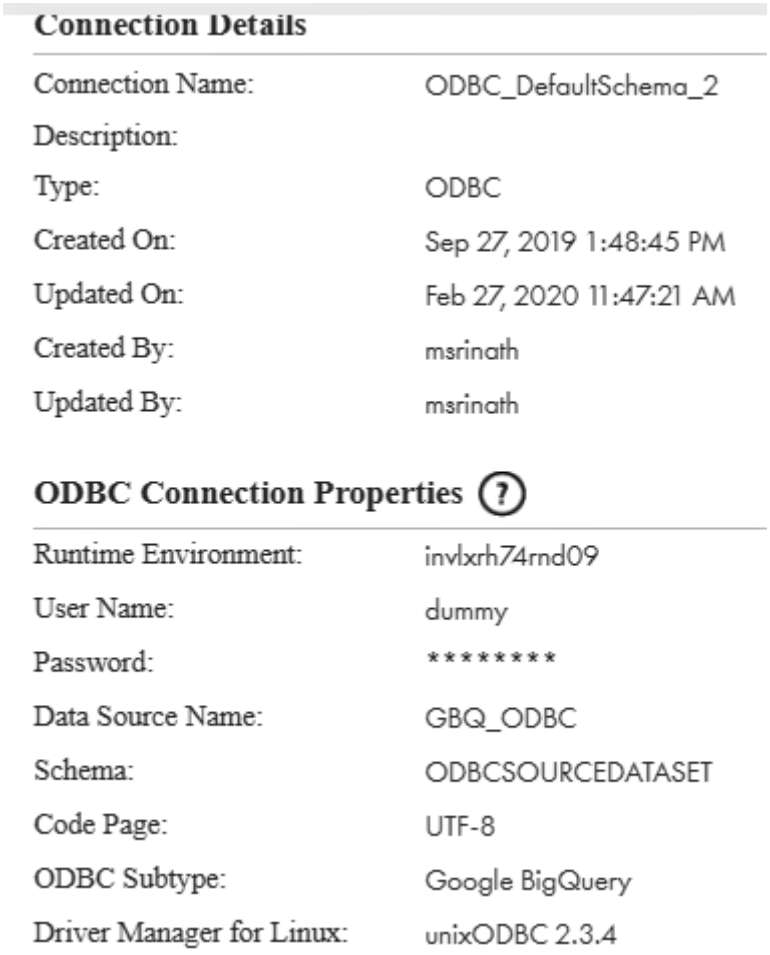
Perform the following steps to create a Google BigQuery ODBC connection on the **Connections** page:

1. In Administrator, click **Connections**.

   The Connections page appears.

2. Click **New Connection**.

The **New Connection** page appears. The following image shows the New Connection page:



3. Configure the following connection details in the **Connection Details** section:

Property	Description
Connection Name	Name of the ODBC connection. For example, sf_odbc.
Description	Description of the connection.
Type	Type of the connection. Select the type of the connection as **ODBC**.

4. Configure the following connection details in the **ODBC Connection Properties** section:

Property	Description
Runtime Environment	Runtime environment that contains the Secure Agent you can use to access the system.
User Name	Username to log in to the Google BigQuery database.

Property	Description
Password	Password to log in to the Google BigQuery database.
Data Source Name	Enter the name of the ODBC data source name that you created for the Google BigQuery database.
Schema	Name of the Google BigQuery schema.
Code Page	The code page of the database server or flat file defined in the connection.
ODBC Subtype	Select **Google BigQuery**.
Driver Manager for Linux	The driver that the Google BigQuery ODBC driver manager sends database calls to. Select **unixODBC 2.3.4** to connect to Google BigQuery.

The Google BigQuery ODBC connection is created successfully.

Create a mapping

Consider a use case where you want to read data form the product_revenue table in Google BigQuery and calculate total revenue for each product. You need to write total revenue for all products to a flat file.

To successfully run a mapping, you must perform the following tasks:

1. Create `gbq_odbc_1` Google BigQuery ODBC connection with `gbq_odbc_dsn` data source name and specify `GBQ_SCHEMA1` dataset in the **Schema** property.
2. Create `gbq_odbc_2` Google BigQuery ODBC connection with `gbq_odbc_dsn` data source name and specify `GBQ_SCHEMA1` dataset in the **Schema** property.
3. Create a Google BigQuery mapping, `m_gbq_product_revenue`.
4. Add a Source transformation and select the `gbq_odbc_1` connection and include the *product_revenue* table to read data using `GBQ_SCHEMA1`.
5. Use the Aggregator transformation to calculate the total revenue of the software products, and write the records to the *total_revenue* target table in Google BigQuery.
   The following image shows a sample mapping, `m_gbq_product_revenue`:

Create a mapping task

When you create a mapping task, select the gbq_product_revenue mapping. Select **Pushdown Optimization** and set the value as **To Source** in the advanced session properties. The following image shows the value selected for pushdown optimization:

Note: You can select Full pushdown when both the source and target is Google BigQuery.

Save and run the mapping task.

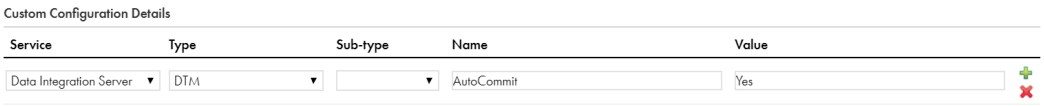
To verify that the pushdown optimization has taken place, you can check the session log for the job. Click **Monitor** > **Activity Log** to view the session log for jobs.

Configuring the Secure Agent for pushdown optimization

Before you configure pushdown optimization, you must add the `AutoCommitOff` flag in the **Custom Configuration Details** for the Secure Agent ans set the value to **Yes**.

To add the `AutoCommitOff` flag, perform the following steps:

1. In Administrator, select **Runtime Environments**.
   The **Runtime Environments** page appears.
2. Select the Secure Agent for which you want to set the `AutoCommitOff` property from the list of available Secure Agents.
3. In the upper-right corner, click **Edit**.
4. In the **Custom Configuration Details** section, select the **Service** as **Data Integration Server** and **Type** as **DTM**.
5. Add `AutoCommit` in the **Name** field and set the **Value** as **Yes**.
   The following image shows the Custom Configuration Details section:



Supported functions and operators for Google BigQuery ODBC mappings

The following table summarizes the availability of pushdown functions in a Google BigQuery database. Columns marked with an X indicate that the function can be pushed to the Google BigQuery database by using source-side or full pushdown optimization. Columns marked with a dash (-) symbol indicate that the function cannot be pushed to the database.

Function	Pushdown	Function	Pushdown	Function	Pushdown
ABORT()	-	INITCAP()	-	REG_MATCH()	-
ABS()	X	INSTR()	X	REG_REPLACE	-
ADD_TO_DATE()	X	IS_DATE()	X	REPLACECHR()	X
AES_DECRYPT()	-	IS_NUMBER()	-	REPLACESTR()	X
AES_ENCRYPT()	-	IS_SPACES()	-	REVERSE()	-
ASCII()	-	ISNULL()	X	ROUND(DATE)	X
AVG()	X	LAST()	-	ROUND(NUMBER)	X
CEIL()	X	LAST_DAY()	X	RPAD()	-

Function	Pushdown	Function	Pushdown	Function	Pushdown
CHOOSE()	-	LEAST()	-	RTRIM()	X
CHR()	-	LENGTH()	X	SET_DATE_PART()	-
CHRCODE()	-	LN()	-	SIGN()	-
COMPRESS()	-	LOG()	-	SIN()	X
CONCAT()	X	LOOKUP	-	SINH()	-
COS()	X	LOWER()	X	SOUNDEX()	-
COSH()	-	LPAD()	-	SQRT()	X
COUNT()	X	LTRIM()	X	STDDEV()	-
CRC32()	-	MAKE_DATE_TIME()	-	SUBSTR()	X
CUME()	-	MAX()	X	SUM()	X
DATE_COMPARE()	-	MD5()	-	SYSDATE()	X
DATE_DIFF()	-	MEDIAN()	-	SYSTIMESTAMP()	X
DECODE()	-	METAPHONE()	-	TAN()	X
DECODE_BASE64()	-	MIN()	X	TANH()	-
DECOMPRESS()	-	MOD()	X	TO_BIGINT	X
ENCODE_BASE64()	-	MOVINGAVG()	-	TO_CHAR(DATE)	X
EXP()	X	MOVINGSUM()	-	TO_CHAR(NUMBER)	X
FIRST()	-	NPER()	-	TO_DATE()	X
FLOOR()	X	PERCENTILE()	-	TO_DECIMAL()	X
FV()	-	PMT()	-	TO_FLOAT()	X
GET_DATE_PART()	X	POWER()	X	TO_INTEGER()	X
GREATEST()	-	PV()	-	TRUNC(DATE)	X
IIF()	-	RAND()	-	TRUNC(NUMBER)	X
IN()	-	RATE()	-	UPPER()	X
INDEXOF()	-	REG_EXTRACT()	-	VARIANCE()	-

Note: When you push the SYSDATE(), SYSTIMESTAMP(), IS_DATE(), ROUND(DATE), GET_DATE_PART(), ADD_TO_DATE(), or TRUNC(DATE) function to Google BigQuery, you must add the **Custom Properties** property under **Advanced Session Properties** tab when you create a mapping task and specify **EnableAdvancedPDOForBigQuery=Yes** in the **Session Property Value** field.

The following list summarizes the availability of pushdown operators in a Google BigQuery database:

+, -, *, /, %, ||, >, =, >=, <=, !=, AND, OR, NOT, ^=.

Supported transformations and data types for Google BigQuery ODBC mappings

When you use pushdown optimization, the Secure Agent converts the expression in the transformation by determining equivalent operators and functions in the database.

The following table lists the transformation logic that the Secure Agent can push to a Google BigQuery source or target:

Transformations Supported	Pushdown Type
Aggregator	Source, Full
Expression	Source, Full
Filter	Source, Full
Joiner	Source, Full
Lookup	Source, Full
Sorter	Source, Full
Union	Source, Full
Router	Full

The following table lists the Google BigQuery data types that can be used for pushdown optimization:

Google BigQuery Data Type	Transformation Data Type
Boolean	Integer
Date	DateTime
DateTime	DateTime
Float	Double
Integer	BigInt
Record	String
Numeric	Decimal **Note:** Google BigQuery ODBC connection supports maximum precision of 28 and maximum scale of 9.
String	String
Byte	Byte

Google BigQuery Data Type	Transformation Data Type
Time	DateTime
Timestamp	DateTime

Rules and guidelines for pushdown optimization

Consider the following rules and guidelines when you push functions to a Google BigQuery database:

- The **Update Override** ODBC advanced target property is not applicable when you use an ODBC connection to configure pushdown optimization to write data to a Google BigQuery target.
- Update, upsert, and delete operations are not applicable when you use an ODBC connection to configure pushdown optimization to write data to a Google BigQuery target.
- You cannot use a custom query as a source object when you use a ODBC connection to configure pushdown optimization.
- When you configure pushdown optimization to write data to a Google BigQuery target, the **Truncate Target** option is not supported. You can configure a pre SQL in the source to delete data from the target table.
- When you configure pushdown optimization, ensure that the transformation does not contain a variable port.
- To push the ADD_TO_DATE() function to the Google BigQuery database, you must define the arguments of the Date data type.
- To push the GET_DATE_PART() function to the Google BigQuery database, you must define the arguments of the Date, DateTime, or Timestamp data type.
- To push the INSTR() function to the Google BigQuery database, you must use the following format: INSTR(string, search_value)
- To push the LAST_DAY() function to the Google BigQuery database, you must define the arguments of the Date data type.
- To push the MAX() function to the Google BigQuery database, you must define the arguments of the Number data type.
- To push the MIN() function to the Google BigQuery database, you must define the arguments of the Date, Number, or String data type.
- To push the ROUND(DATE) or TRUNC(DATE) function to the Google BigQuery database, you must define the arguments of the Timestamp data type.
- To push the TO_CHAR(DATE) function to the Google BigQuery database, you must define the arguments of the Timestamp data type.
- When you push the SYSTIMESTAMP() function to the Google BigQuery database, do not specify any format. The Google BigQuery database returns the complete timestamp.
- When you push the SYSDATE() function to the Google BigQuery database, you must map the output of the expression transformation to a column of Date data type in the Google BigQuery target.
- When you push the TO_DATE() function to the Google BigQuery database, you must configure the output field in the expression transformation to a column of Timestamp data type.
- When you push TO_DATE(string, format) or IS_DATE(string, format) to Google BigQuery and specify the SS, SS.MS, or SS.US format, the function returns the same value for the formats in seconds and subseconds.

- When you push TO_DATE(string, format) or IS_DATE(string, format) to Google BigQuery, you must use the following format arguments:
  - YYYY
  - YY
  - MONTH
  - MON
  - MM
  - DD
  - HH24
  - HH12
  - MI
  - SS
  - SS.MS
  - SS.US
  - PM
  - AM
  - pm
  - am

  Note: If you specify HH12 in the format argument, you must specify AM, am, PM, or pm.
- When you push TO_CHAR() to Google BigQuery, you must use the following format arguments:
  - YYYY
  - YY
  - MONTH
  - MON
  - MM
  - Q
  - DD
  - DDD
  - D
  - DY
  - HH
  - HH24
  - HH12
  - MI
  - SS
  - SS.MS
  - SS.US
  - PM
  - AM

- pm
- am
- T

Note: If you specify HH12 in the format argument, you must specify AM, am, PM, or pm.

- When you push ROUND(string, format) or TRUNC(string, format) to Google BigQuery, you must use the following format arguments:
  - HH24
  - MI
  - SS
  - DD
  - MS
- When you push a function that returns a Boolean value, you must configure the output field in the expression transformation to a column of Integer data type.
- If you configure a Lookup condition, you must use only the equals to (=) operator. If you use any operator other than the equals to (=) operator, the mapping fails.
- When you configure the Lookup Source Filter or Lookup SQL Override property in a Lookup transformation, you must add the **Allow Temporary View for Pushdown** property under the **Advanced Session Properties** tab when you create a mapping task and select **Yes** in the **Session Property Value** field.
- If the Lookup transformation name contains Unicode characters, the mapping fails.
- When you configure an unconnected Lookup transformation, the fields specified in the Lookup SQL Override property are matched with the lookup fields based on the field names.
- When you configure a Lookup transformation and select **Report error** in the **Multiple Matches** property, the mapping fails and the Secure Agent logs the following error in the session log file:
  `FnName: Execute Direct - [Informatica] [BigQuery] (70) Invalid query: Scalar subquery produced more than one element`
- If you specify a function in the Lookup SQL Override property, you must specify the alias name for the function with the lookup field as an argument.
- When you read data of date, datetime, or timestamp data type and write the data as a string to the target, you must add the **DateTime Format String** property under the **Advanced Session Properties** tab when you create a mapping task and specify **YYYY-MM-DD HH24:MI:SS** in the **Session Property Value** field.
- Ensure that the Data Source Name, User name, and the Driver Manager for Linux in the source and target ODBC connection are same. If the values of the Data Source Name, User name, and the Driver Manager for Linux are different in the source and target ODBC connection, the mapping fails with the following error:
  `"Pushdown optimization stops because the connections are not pushdown compatible."`

Author

Anupam Nayak

Acknowledgements

The author would like to acknowledge Manoj Srinath, for his technical assistance.