Informatica

Informatica® PowerExchange for Google BigQuery

10.5.6

User Guide for PowerCenter

Informatica PowerExchange for Google BigQuery User Guide for PowerCenter
10.5.6
May 2024

Publication Date: 2024-05-30

Table of Contents

Preface

Use the *Informatica® PowerExchange® for Google BigQuery User Guide for PowerCenter* to learn how to read from and write to Google BigQuery by using PowerCenter Client. Learn to create a Google BigQuery connection, develop mappings, and run sessions in an Informatica domain.

Informatica Resources

Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit https://network.informatica.com.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- Review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit https://kb.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com.

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions.

If you are an Informatica Network member, you can access Informatica Velocity resources at http://velocity.informatica.com.

If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that augment, extend, or enhance your Informatica implementations. By leveraging any of the hundreds of solutions from Informatica developers and partners, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through Online Support on Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
http://www.informatica.com/us/services-and-training/support-services/global-support-centers.

If you are an Informatica Network member, you can use Online Support at http://network.informatica.com.

CHAPTER 1

Introduction to PowerExchange for Google BigQuery

This chapter includes the following topics:

PowerExchange for Google BigQuery Overview

You can use PowerExchange for Google BigQuery for connectivity between PowerCenter and Google BigQuery.

You can use Google BigQuery objects as sources and targets in mappings. When you use Google BigQuery objects in mappings, you must configure properties specific to Google BigQuery.

Example

Your organization is an open source log data collector, which collects log data from multiple sources and unifies them.

Logs help you understand how systems and applications perform. As the scale and complexity of the system increases, it is difficult to manage multiple logs from different sources.

To overcome this problem, you can use PowerExchange for Google BigQuery to write data to a Google BigQuery target and query terabytes of logs in seconds. You can then use the data to fix and improve the system performance in near real time.

Introduction to Google BigQuery

Google BigQuery is a fast, highly scalable, cost-effective and fully managed enterprise data warehouse that the Google Cloud Platform provides. You can store and analyze massive amounts of data using ANSI:2011 compliant SQL queries.

Administration of Google BigQuery

Google BigQuery is a RESTful web service that the Google Cloud Platform provides.

Before you use PowerExchange for Google BigQuery, you must complete the following prerequisite tasks:

- Ensure you have a service account in your Google account to access Google BigQuery.
- Ensure you have the client_email, project_id, and private_key values for the service account. You will need to enter these details when you create a Google BigQuery connection in Developer tool.
- Ensure you have the project ID, dataset ID, source table name, and target table name when you create mappings in PowerCenter.
- Verify that you have read and write access to the Google BigQuery dataset that contains the source table and target table.
- When you read data from or write data to a Google BigQuery table, you must have the following permissions:
  - bigquery.datasets.create
  - bigquery.datasets.get
  - bigquery.datasets.getIamPolicy
  - bigquery.datasets.updateTag
  - bigquery.models.*
  - bigquery.routines.*
  - bigquery.tables.create
  - bigquery.tables.delete
  - bigquery.tables.export
  - bigquery.tables.get
  - bigquery.tables.getData
  - bigquery.tables.list
  - bigquery.tables.update
  - bigquery.tables.updateData
  - bigquery.tables.updateTag
  - resourcemanager.projects.get
  - resourcemanager.projects.list
  - bigquery.jobs.create
- When you only read data from a Google BigQuery table, 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.export

- bigquery.tables.get
- bigquery.tables.getData
- bigquery.tables.list
- resourcemanager.projects.get
- resourcemanager.projects.list
- bigquery.jobs.create
- bigquery.tables.create

- If you use bulk mode, verify that you have write access to the Google Cloud Storage path where the PowerCenter Integration Service creates the staging file.
- If you use staging mode, verify that you have read access to the Google Cloud Storage path where the PowerCenter Integration Service creates the staging file to store the data from the Google BigQuery source.

CHAPTER 2

PowerExchange for Google BigQuery Configuration

This chapter includes the following topics:

PowerExchange for Google BigQuery Configuration Overview

PowerExchange for Google BigQuery installs with Informatica services.

If you upgrade from a previous version, you must register the PowerExchange for Google BigQuery plug-in with the PowerCenter repository.

Registering the PowerExchange for Google BigQuery Plug-in

After you install PowerExchange for Google BigQuery, upgrade it from a previous version, or apply a hotfix, it is mandatory to register the PowerExchange for Google BigQuery plug-in with the PowerCenter repository.

To register the plug-in, the repository must be running in exclusive mode. Use the Administrator tool or the `pmrep RegisterPlugin` command line program to register the plug-in. If you do not have the correct privileges to register the plug-in, contact the user who manages the PowerCenter Repository Service.

The plug-in file is an `.xml` file that defines the functionality of the adapter. When you install the server component, the installer copies the plug-in file to the following directory:

```
<Informatica installation directory>/server/bin/Plugin
```

The name of the plug-in file for PowerExchange for Google BigQuery is `bigqueryPlugin.xml`.

Registering the PowerExchange for Google BigQuery Plug-in from the Administrator Tool

Register a repository plug-in to add its functionality to the repository.

1. Run the PowerCenter Repository Service in exclusive mode.
2. In the **Navigator**, select the PowerCenter Repository Service to which you want to add the plug-in.
3. In the **Contents** panel, click the **Plug-ins** view.
4. In the **Actions** menu of the **Domain** tab, select **Register Plug-in**.
5. On the **Register Plugin** page, click the **Browse** button to locate the plug-in file.
6. Enter your user name and password.
7. Click **OK**.

   The PowerCenter Repository Service registers the plug-in with the repository. The results of the registration operation appear in the activity log.
8. Run the PowerCenter Repository Service in normal mode.

Registering the PowerExchange for Google BigQuery Plug-in from the Command Line Program

You can use the pmrep RegisterPlugin command to register the plug-in from the command line program.

1. Run the PowerCenter Repository Service in exclusive mode.
2. Run the `pmrep Connect` command to connect to the Repository Service using a user account with Administrator Repository privilege.

   The RegisterPlugin command uses the following syntax:

   ```
   pmrep connect -r <repository name> -d <domain_name> -n <domain user name> -x
   <domain_password>
   ```
3. Find `bigqueryPlugin.xml` in the following directory:

   ```
   <Informatica installation directory>\server\bin\Plugin
   ```
4. Run the `pmrep RegisterPlugin` command to update the repository.

   The `RegisterPlugin` command uses the following syntax:

   ```
   pmrep registerplugin -i <Informatica installation directory>\server\bin\Plugin
   \bigqueryPlugin.xml -e –N
   ```

CHAPTER 3

Google BigQuery Sources and Targets

This chapter includes the following topics:

Google BigQuery Sources and Targets Overview

You can create a mapping with a Google BigQuery source to extract data from Google BigQuery. You can create a mapping with any source and a Google BigQuery target to load data to Google BigQuery.

When the PowerCenter Integration Service extracts data from the source or loads data to the target, it converts the data based on the data types associated with the source or the target.

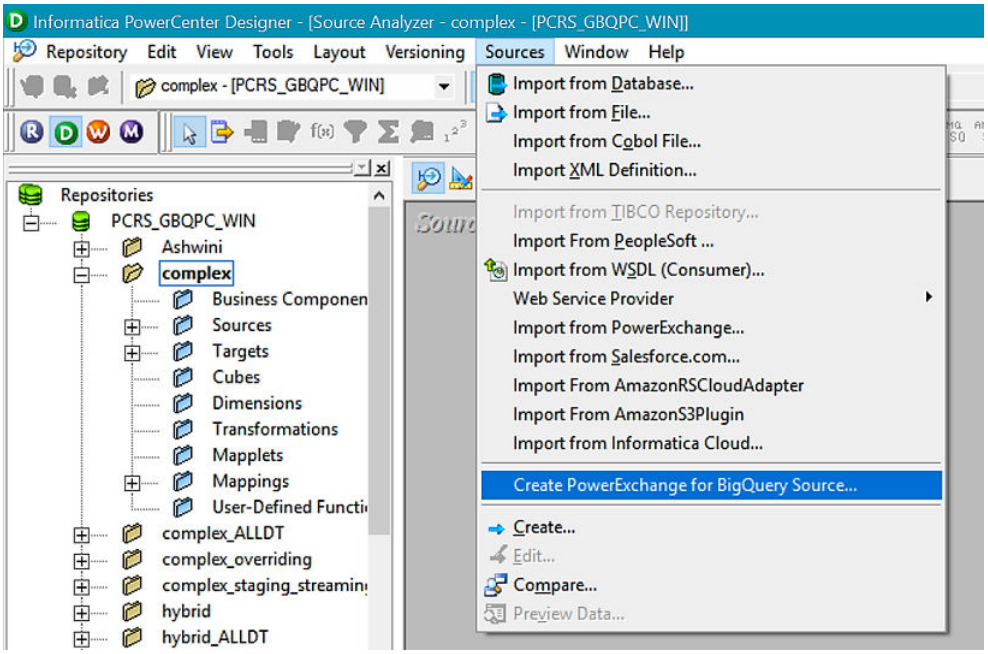
Import Google BigQuery Source and Target Definitions

Use the **Create PowerExchange for BigQuery Source** or **Create PowerExchange for BigQuery Target** wizard to import Google BigQuery source and target definitions into the PowerCenter repository.

You must import Google BigQuery source and target objects before you create a mapping.

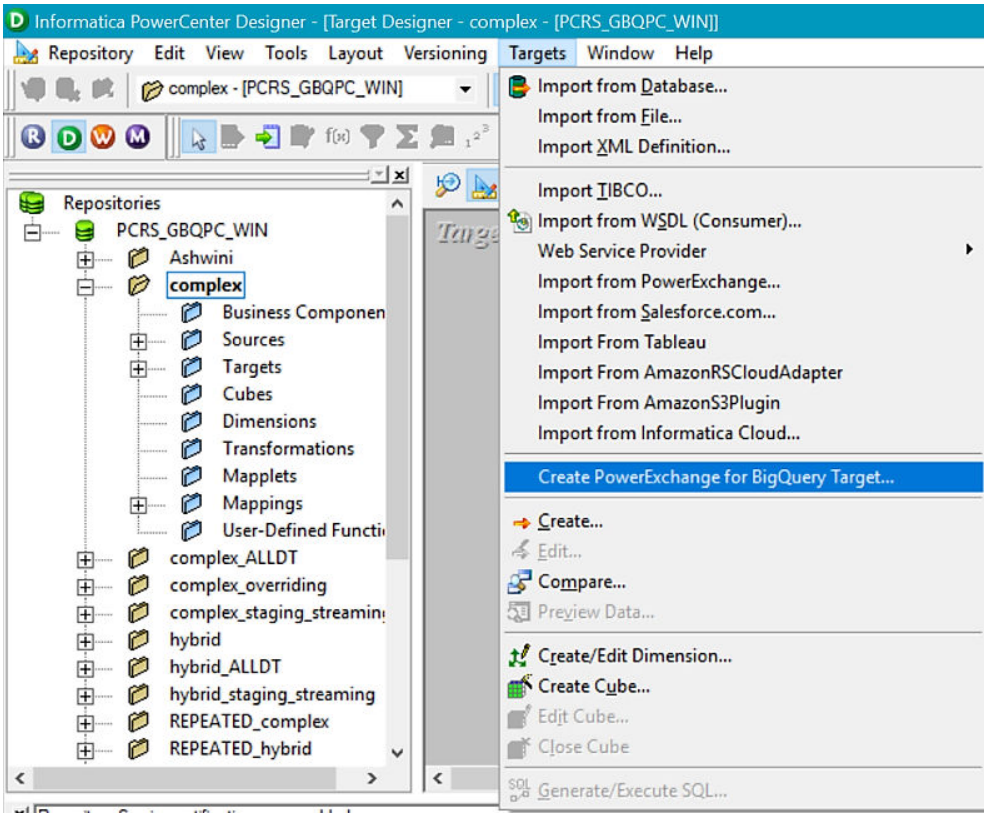
1. Start PowerCenter Designer, and connect to a PowerCenter repository configured with a Google BigQuery instance.
2. Open a source or target folder.

3. Select **Source Analyzer** or **Target Designer**.
   - In the Source Analyzer, click **Sources** > **Create PowerExchange for BigQuery Source** as shown in the following image:



   The **Google BigQuery Connection** wizard appears.

- In the Target Analyzer, click **Targets** > **Create PowerExchange for BigQuery Target** as shown in the following image:



The **Google BigQuery Connection** dialog box appears.

4. Configure the following connection parameters:

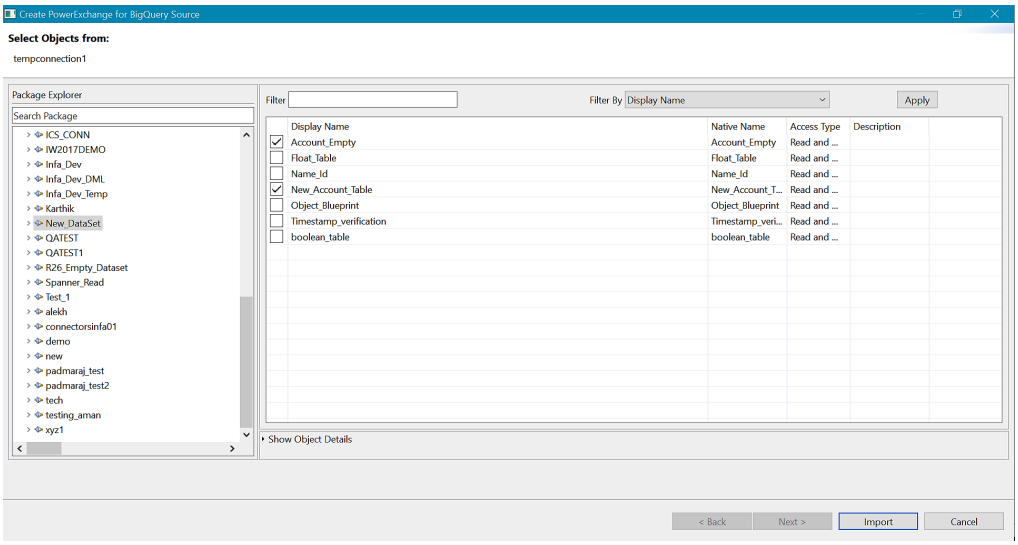
Connection Parameter	Description
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account.
Connection mode	The mode that you want to use to read data from or write data to Google BigQuery. Select one of the following connection modes: - Simple. Flattens each field within the Record data type field as a separate field in the mapping. - Hybrid. Displays all the top-level fields in the Google BigQuery table including Record data type fields. PowerExchange for Google BigQuery displays the top-level Record data type field as a single field of the String data type in the mapping. - Complex. Displays all the columns in the Google BigQuery table as a single field of the String data type in the mapping. Default is Simple.

Connection Parameter	Description
Schema Definition File Path	Specifies a directory on the client machine where the PowerCenter Integration Service must create a JSON file with the sample schema of the Google BigQuery table. The JSON file name is the same as the Google BigQuery table name. Alternatively, you can specify a storage path in Google Cloud Storage where the PowerCenter Integration Service must create a JSON file with the sample schema of the Google BigQuery table. You can download the JSON file from the specified storage path in Google Cloud Storage to a local machine.
Project ID	Specifies the project_id value present in the JSON file that you download after you create a service account. If you have created multiple projects with the same service account, enter the ID of the project that contains the dataset that you want to connect to.
Storage Path	This property applies when you read or write large volumes of data. Path in Google Cloud Storage where the PowerCenter Integration Service creates a local stage file to store the data temporarily. You can either enter the bucket name or the bucket name and folder name. For example, enter `gs://<bucket_name>` or `gs://<bucket_name>/<folder_name>`
Use Legacy SQL For Custom Query	Uses Legacy SQL to define a custom query. You can clear this option, you must use Standard SQL to define a custom query.
Dataset Name for Custom Query	When you define a custom query, you must specify a Google BigQuery dataset.
Region id	The region name where the Google BigQuery dataset resides. For example, if you want to connect to a Google BigQuery dataset that resides in Las Vegas region, specify **us-west4** as the **Region ID**. **Note:** In the **Storage Path** connection property, ensure that you specify a bucket name or the bucket name and folder name that resides in the same region as the dataset in Google BigQuery. For more information about the regions supported by Google BigQuery, see the following Google BigQuery documentation:https://cloud.google.com/bigquery/docs/locations
Optional Properties	Specifies whether you can configure certain source and target functionalities through custom properties. You can select one of the following options: - None. If you do not want to configure any custom properties, select None. - Required. If you want to specify custom properties to configure the source and target functionalities. Default is None.
Provide Optional Properties	Comma-separated key-value pairs of custom properties in the Google BigQuery connection to configure certain source and target functionalities. Appears only when you select **Required** in the Optional Properties.

5. Click **Test** to test the connection.
6. Click **Finish** to add the connection.

   The **Select Objects from** tab appears.
7. Select the dataset in **Package Explorer**.

A list of table appears as shown in the following image:



8. Select the table that you want to import, and then click **Import**. You can import multiple tables from a Google BigQuery dataset.

   To view the table metadata, select the table, and double-click the table name.

Configure a Google BigQuery Source Definition Using a Custom SQL Query

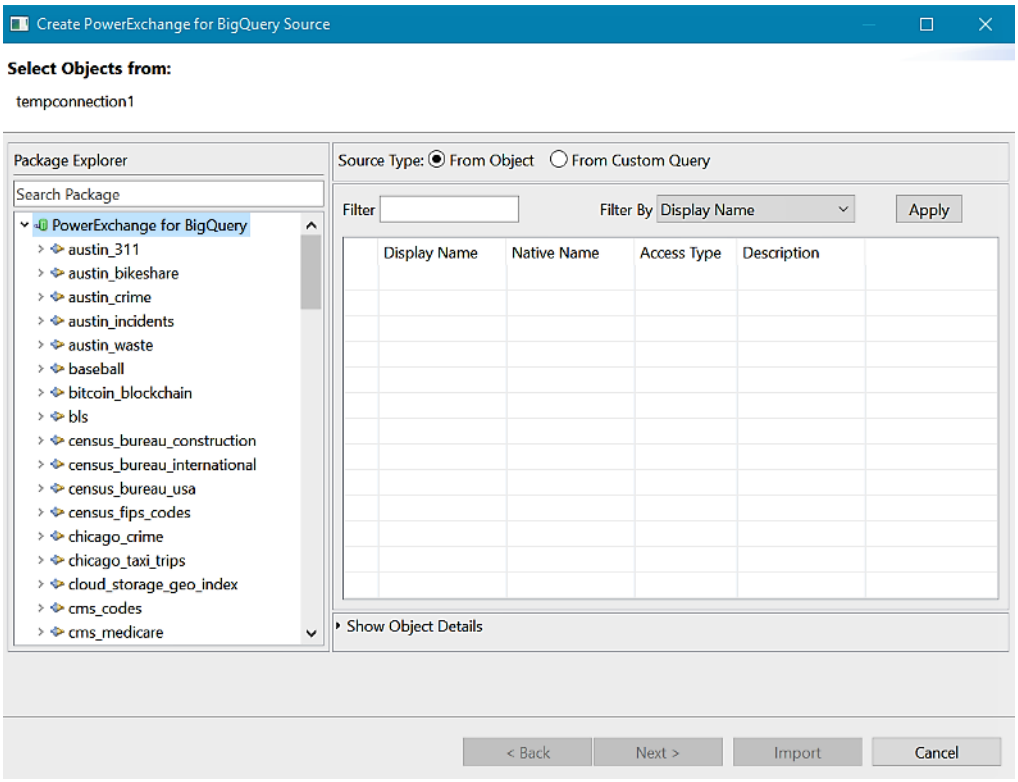
You can configure a Google BigQuery source definition either by selecting the tables in the package explorer or by using a custom SQL query.

1. Start PowerCenter Designer and connect to a PowerCenter repository configured with a Google BigQuery instance.
2. Open a source folder.
3. Open the **Source Analyzer**, and then click **Sources** > **Create PowerExchange For BigQuery Source**.

   The **Google BigQuery Connection Details** dialog box appears.
4. Add the connection details, and click **Finish** to add the connection.
5. Click **Next**.

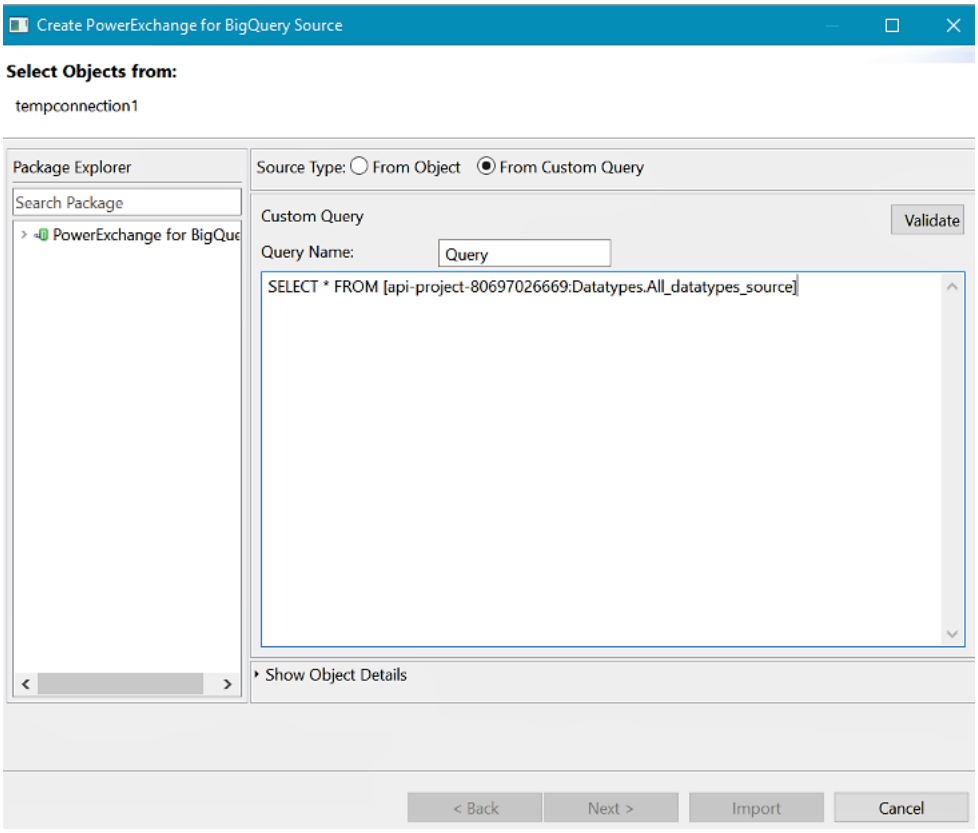
   The **Select Objects from** tab appears.
6. Select the dataset in **Package Explorer**.
7. To configure a Google BigQuery source, select the source type:
   - Select **From Object** to select from the list of tables that you want to import, and then click **Import**.

The following image shows the list of tables that appear when you select the **From Object** source type:



- Select **From Custom Query** to configure the source using a custom SQL query, and click **Import**.

The following image shows the configured custom query to import the tables:



CHAPTER 4

Google BigQuery Mappings

This chapter includes the following topics:

Google BigQuery Mappings Overview

After you import a Google BigQuery source or target definition into the PowerCenter repository, you can create a mapping to extract data from a Google BigQuery source or load data to a Google BigQuery target.

You can extract data from one or more Google BigQuery sources, and load data to one or more Google BigQuery targets.

You can enter a filter condition to reduce the number of source rows the PowerCenter Integration Service returns from Google BigQuery sources. You can enter a single filter condition or a series of conditions.

Note: You cannot preview data of a Google BigQuery source or target definition.

Source Filter

You can enter a filter condition to reduce the number of source rows the PowerCenter Integration Service returns from Google BigQuery sources. You can enter a single filter condition or a series of conditions.

Use the source filter in the Application Source Qualifier to retrieve rows from an entity that meet a condition.

You can provide a source filter to improve the performance when you read data from Google BigQuery.

You can use the Native or Platform expression to apply a filter condition on Google BigQuery columns of the following data types:

- Integer
- Float
- Numeric
- String
- Timestamp

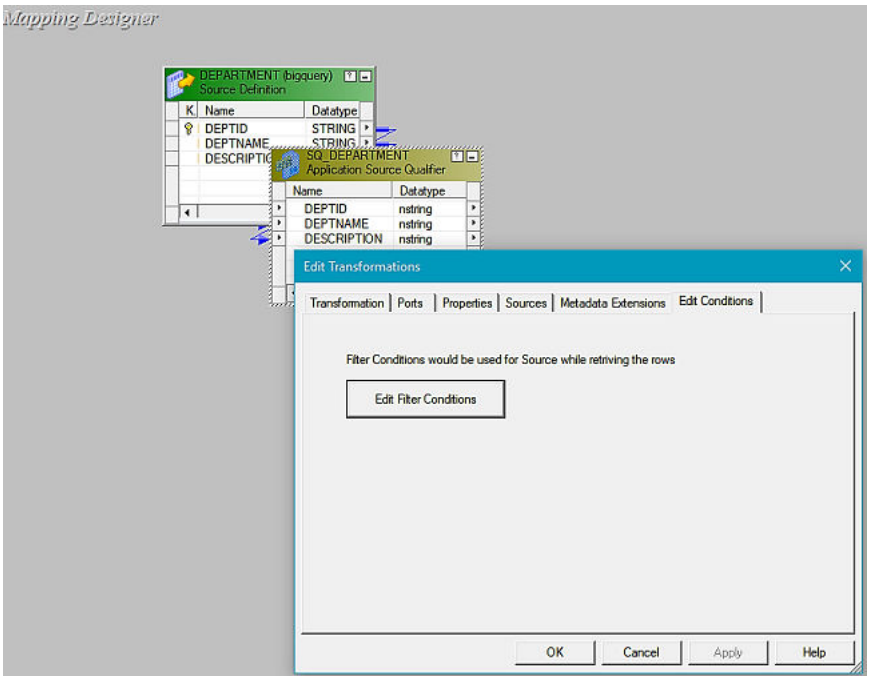
Note: When you create a native or platform expression to query columns of Numeric data type in a Google BigQuery table, you must ensure that you use the Google BigQuery connection in hybrid connection mode.

Configuring a Source Filter

Configure a source filter in the Application Source Qualifier.

1. In the **Mapping Designer**, double-click the Application Source Qualifier.

   The **Edit Transformations** dialog box appears as shown in the following image:

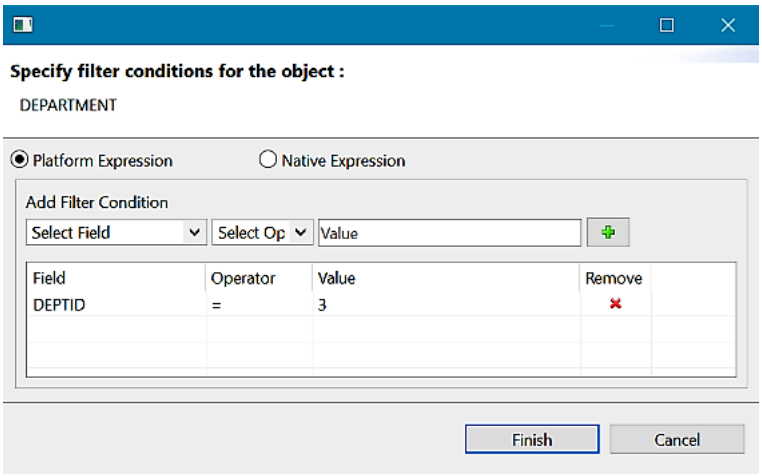


2. Click the **Edit Conditions** tab.
3. Click **Edit Filter Conditions**.

   The **Add Filter Condition** dialog box appears.

4. Select **Platform Expression** or **Native Expression**.

   - If you configure a platform expression, select the filter field and operator that you want to specify in the condition, enter a value for the condition, and click **Add Condition**
     The following image shows a platform filter expression configured for a Google BigQuery source:

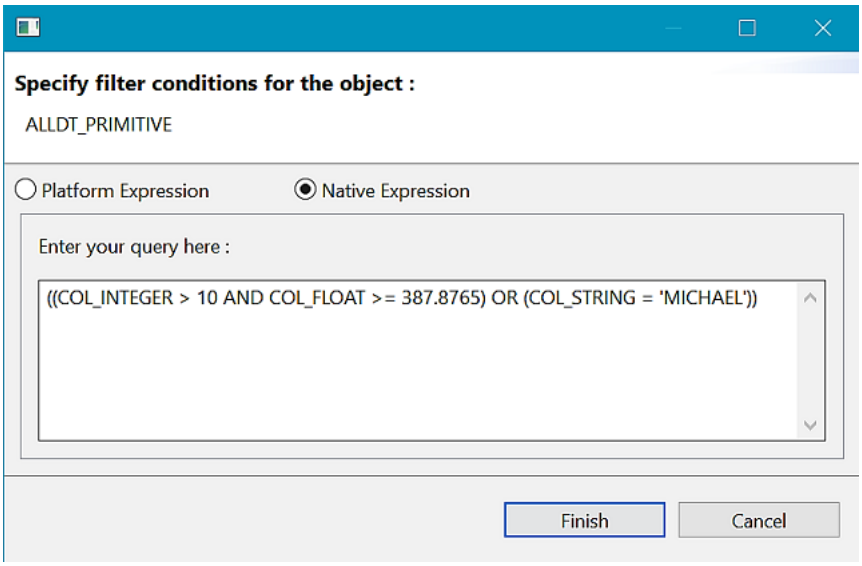


   - If you configure a native expression, specify a filter expression in the following format:

```
<Attribute><Operator><Value>
```

You can use AND, OR, or nested conditions in the filter expression. The expression that you enter becomes the WHERE clause in the query used to retrieve records from the source.

The following image shows a native filter expression configured for a Google BigQuery source:



5. Click **Finish** to add the filter condition.
6. Click **OK**.

CHAPTER 5

Google BigQuery Sessions

This chapter includes the following topics:

Google BigQuery Sessions Overview

After you create mappings, you can create a session to extract and load data.

You must configure a Google BigQuery connection in the **Workflow Manager** to extract data from or load data to a Google BigQuery table. You can define properties in a session to determine how the PowerCenter Integration Service must extract data from a Google BigQuery source or load data to a Google BigQuery target.

Google BigQuery Connections

Create a Google BigQuery connection to read data from a Google BigQuery source and write data to a Google BigQuery target. You must create a connection for each dataset that you want to connect to. You can use

Google BigQuery connections in mappings. When you create a Google BigQuery connection, you can configure a connection mode based on how you want to read and write the data.

Connection Modes

You can configure a Google BigQuery connection to use one of the following connection modes:

Simple mode

If you use simple mode, PowerExchange for Google BigQuery flattens each field within the Record data type field as a separate field in the source or target definition.

Hybrid mode

If you use hybrid mode, PowerExchange for Google BigQuery displays all the top-level fields in the Google BigQuery table including Record data type fields. PowerExchange for Google BigQuery displays the top-level Record data type field as a single field of the String data type in the source or target definition.

Complex mode

If you use complex mode, PowerExchange for Google BigQuery displays all the columns in the Google BigQuery table as a single field of the String data type in the source or target definition.

Connection Mode Example

PowerExchange for Google BigQuery reads and writes the Google BigQuery data based on the connection mode that you configure for the Google BigQuery connection.

You have a Customers table in Google BigQuery that contains primitive fields and the **Address** field of the Record data type. The Address field contains two primitive sub-fields, **City** and **State**, of the String data type.

The following image shows the schema of the Customers table in Google BigQuery:

ID	INTEGER	NULLABLE
Name	STRING	NULLABLE
Address	RECORD	NULLABLE
Address.**City**	STRING	NULLABLE
Address.**State**	STRING	NULLABLE
Mobile	STRING	REPEATED
Totalpayments	FLOAT	NULLABLE
age	INTEGER	REPEATED

The following table shows the Customers table data in Google BigQuery:

ID	Name	Address.City	Address.State	Mobile	Totalpayments
14	John	LOS ANGELES	CALIFORNIA	+1-9744884744	18433.90
				+1-8267389993	
29	Jane	BOSTON	MANHATTAN	+1-8789390309	28397.33
				+1-9876553784	
				+1-8456437848	

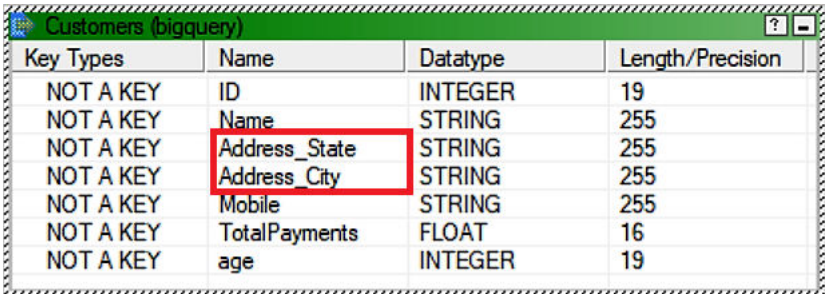
Simple Mode

If you use simple connection mode, PowerExchange for Google BigQuery flattens each field within the Record data type field as a separate field in the **Source Analyzer**.

The following table shows two separate fields, Address_City and Address_State, for the respective sub-fields within the Address Record field in the Customers table:

ID	Name	Address_City	Address_State	Mobile	Totalpayments
14	John	LOS ANGELES	CALIFORNIA	+1-9744884744	18433.90
14	John	LOS ANGELES	CALIFORNIA	+1-8267389993	18433.90
29	Jane	BOSTON	MANHATTAN	+1-8789390309	28397.33
29	Jane	BOSTON	MANHATTAN	+1-9876553784	28397.33
29	Jane	BOSTON	MANHATTAN	+1-8456437848	28397.33

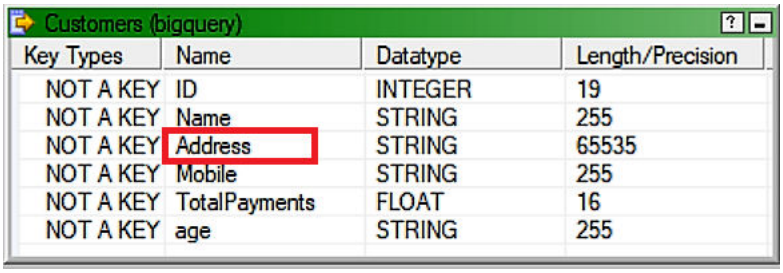
The following image shows the Address_State and Address_City fields in the **Source Analyzer**:



Hybrid Mode

If you use hybrid connection mode, PowerExchange for Google BigQuery displays all the top-level fields in the Google BigQuery table including Record data type fields. PowerExchange for Google BigQuery displays the top-level Record data type field as a single field of the String data type in the **Source Analyzer**.

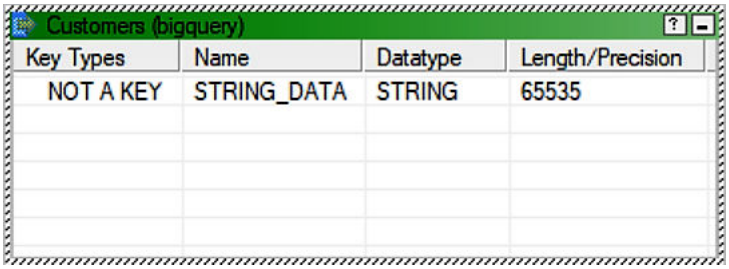
The following image shows the Address field in the **Source Analyzer**:



Complex Mode

If you use complex connection mode, PowerExchange for Google BigQuery displays all the columns in the Google BigQuery table as a single field of the String data type in the **Source Analyzer**.

The following image shows the STRING_DATA field in the **Source Analyzer**:



Rules and Guidelines for Google BigQuery Connection Modes

Simple Mode

Consider the following rules and guidelines when you configure a Google BigQuery connection to use simple connection mode:

- If the Google BigQuery source table contains repeated columns, you cannot configure source filters for these columns.
- If the Google BigQuery table contains more than one repeated column, you cannot preview data.
- If the Google BigQuery target table contains columns of the Record data type and repeated columns, you cannot configure update, upsert, and delete operations for these columns.
- If the Google BigQuery target table contains columns that are marked as required in Google BigQuery, you cannot configure upsert operation for these columns.
- You can configure upsert operation on columns of primitive data types in the Google BigQuery table.
- When you configure an upsert operation, you cannot perform update operation on multiple rows in the Google BigQuery target.
- When you read data from a Google BigQuery source, you must not map more than one repeated column in a single mapping. You must create multiple mappings for each repeated column.

Hybrid Mode

Consider the following rules and guidelines when you configure a Google BigQuery connection to use hybrid connection mode:

- If the Google BigQuery source table contains columns of the Record data type and repeated columns, you cannot configure source filters for these columns.
- You cannot configure update, upsert, and delete operations for columns of the Record data type and repeated columns.
- You must select JSON (Newline Delimited) format as the data format of the staging file under the target session properties. You can use CSV format as the data format of the staging file only when the Google BigQuery table does not contain columns of the Record data type or repeated columns.
- The following CSV formatting options in the target session properties are not applicable:
  - Allow Quoted Newlines
  - Field Delimiter
  - Allow Jagged Rows

Complex Mode

Consider the following rules and guidelines when you configure a Google BigQuery connection to use complex connection mode:

- When you configure a Google BigQuery source connection to use complex connection mode, you cannot configure data filters for the source.
- You cannot configure update, upsert, and delete operations.
- You must select JSON (Newline Delimited) format as the data format of the staging file under the target session properties.
- You cannot use CSV format as the data format of the staging file. The following CSV formatting options in the target session properties are not applicable:
  - Allow Quoted Newlines
  - Field Delimiter
  - Allow Jagged Rows

PowerExchange for Google BigQuery Connections

When you configure a Google BigQuery connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the Google BigQuery database.

The following table describes the Google BigQuery connection properties:

Property	Description
Service Account ID	Specifies the client_email value present in the JSON file that you download after you create a service account.
Service Account Key	Specifies the private_key value present in the JSON file that you download after you create a service account.
Connection mode	The mode that you want to use to read data from or write data to Google BigQuery. Select one of the following connection modes: - Simple. Flattens each field within the Record data type field as a separate field in the mapping. - Hybrid. Displays all the top-level fields in the Google BigQuery table including Record data type fields. PowerExchange for Google BigQuery displays the top-level Record data type field as a single field of the String data type in the mapping. - Complex. Displays all the columns in the Google BigQuery table as a single field of the String data type in the mapping. Default is Simple.
Schema Definition File Path	Specifies a directory on the client machine where the PowerCenter Integration Service must create a JSON file with the sample schema of the Google BigQuery table. The JSON file name is the same as the Google BigQuery table name. Alternatively, you can specify a storage path in Google Cloud Storage where the PowerCenter Integration Service must create a JSON file with the sample schema of the Google BigQuery table. You can download the JSON file from the specified storage path in Google Cloud Storage to a local machine.
Project ID	Specifies the project_id value present in the JSON file that you download after you create a service account. If you have created multiple projects with the same service account, enter the ID of the project that contains the dataset that you want to connect to.

Property	Description
Storage Path	This property applies when you read or write large volumes of data. Path in Google Cloud Storage where the PowerCenter Integration Service creates a local stage file to store the data temporarily. You can either enter the bucket name or the bucket name and folder name. For example, enter `gs://<bucket_name>` or `gs://<bucket_name>/<folder_name>`
Dataset ID	Not applicable for PowerExchange for Google BigQuery.
Use Legacy SQL For Custom Query	Uses Legacy SQL to define a custom query. You can clear this option, you must use Standard SQL to define a custom query.
Dataset Name for Custom Query	When you define a custom query, you must specify a Google BigQuery dataset.
Region id	The region name where the Google BigQuery dataset resides. For example, if you want to connect to a Google BigQuery dataset that resides in Las Vegas region, specify **us-west4** as the **Region ID**. **Note:** In the **Storage Path** connection property, ensure that you specify a bucket name or the bucket name and folder name that resides in the same region as the dataset in Google BigQuery. For more information about the regions supported by Google BigQuery, see the following Google BigQuery documentation:https://cloud.google.com/bigquery/docs/locations
Optional Properties	Specifies whether you can configure certain source and target functionalities through custom properties. You can select one of the following options: - None. Select if you do not want to configure any custom properties. - Required. If you want to specify custom properties to configure the source and target functionalities. Default is None.
Provide Optional Properties	Comma-separated key-value pairs of custom properties to enable additional source and target functionalities. Appears only when you select **Required** in the Optional Properties.

Configuring a Google BigQuery Connection

Configure a Google BigQuery connection in the **Workflow Manager** to define the connection attributes that the PowerCenter Integration Service uses to connect to the Google BigQuery database.

1. In the Workflow Manager, click **Connections** > **Application**.

   The **Application Connection Browser** dialog box appears.
2. Click **New**.

   The **Select Subtype** dialog box appears.
3. Select **bigquery** and click **OK**.

   The **Application Connection Editor** dialog box appears.
4. Enter a name for the Google BigQuery connection.
5. Enter the Google BigQuery connection attributes.
6. Click **OK** to create a Google BigQuery connection.

Read Modes

When you use PowerExchange for Google BigQuery, you can read data by using direct mode or staging mode. Before you choose a mode, see the Google documentation to understand the cost implications and trade-offs for each mode.

You can read data from a Google BigQuery source by using one of the following modes:

Direct Mode

Use direct mode when the volume of data that you want to read is small. In direct mode, PowerExchange for Google BigQuery directly reads data from a Google BigQuery source. You can configure the number of rows that you want PowerExchange for Google BigQuery to read in one request.

Staging Mode

Use staging mode when you want to read large volumes of data in a cost-efficient manner.

In staging mode, PowerExchange for Google BigQuery first exports the data from the Google BigQuery source into Google Cloud Storage. After the export is complete, PowerExchange for Google BigQuery downloads the data from Google Cloud Storage into a local stage file. You can configure the local stage file directory in the advanced source properties. PowerExchange for Google BigQuery then reads the data from the local stage file.

When you enable staging file compression, PowerExchange for Google BigQuery compresses the size of the staging file in Google Cloud Storage. PowerExchange for Google BigQuery then downloads the staging file and decompresses the staging file before it reads the file. To improve the performance and download data in parallel, you can configure the number of threads for downloading the staging file.

Write Modes

When you use PowerExchange for Google BigQuery, you can write data by using bulk mode or streaming mode. Before you choose a mode, see the Google documentation to understand the cost implications and trade-offs for each mode.

You can write data to a Google BigQuery target by using one of the following modes:

Bulk mode

Use bulk mode when you want to write large volumes of data in a cost-efficient manner.

In bulk mode, PowerExchange for Google BigQuery first writes the data to a staging file in Google Cloud Storage. When the staging file contains all the data, PowerExchange for Google BigQuery loads the data from the staging file to the Google BigQuery target.

When you enable staging file compression, PowerExchange for Google BigQuery compresses the size of the staging file before it writes data to Google Cloud Storage. PowerExchange for Google BigQuery writes the compressed file to Google Cloud Storage and then submits a load job to the Google BigQuery target.

Note: Enabling compression reduces the time that PowerExchange for Google BigQuery takes to write data to Google Cloud Storage. However, there will be a performance degradation when PowerExchange for Google BigQuery writes data from Google Cloud Storage to the Google BigQuery target.

PowerExchange for Google BigQuery deletes the staging file unless you configure the task or mapping to persist the staging file. You can choose to persist the staging file if you want to archive the data for future reference.

Streaming mode

Use streaming mode when you want the Google BigQuery target data to be immediately available for querying and real-time analysis. Evaluate Google's streaming quota policies and billing policies before you use streaming mode.

In streaming mode, PowerExchange for Google BigQuery directly writes data to the Google BigQuery target. PowerExchange for Google BigQuery appends the data into the Google BigQuery target.

You can configure the number of rows that you want PowerExchange for Google BigQuery to stream in one request. If you want to stream a larger number of rows than the maximum permissible limit prescribed by Google, you can write the data to multiple smaller target tables instead of one large target table. You can create a template table based on which Google BigQuery must create multiple tables. You can define a unique suffix for each table. Google BigQuery creates each table based on the template table and adds the suffix to uniquely identify each table.

CDC mode

Use CDC mode only when you capture changed data from a CDC source. In CDC mode, you can configure PowerExchange for Google BigQuery to capture changed data from any CDC source and write the changed data to a Google BigQuery target table.

Pre-SQL and Post-SQL Commands

You can specify **pre SQL** and **post SQL** session properties for Google BigQuery sources and targets. When you create a Session task in the **Workflow Manager**, you can specify SQL commands on the **Mapping** tab.

You can perform the following operations by using pre SQL and post SQL commands:

- SELECT
- UPDATE
- DELETE

You can configure the options in Google BigQuery with a pre SQL or post SQL statement in the **pre SQL Configuration** or **post SQL Configuration** session properties for Google BigQuery sources and targets.

You must use the following format to specify a pre SQL configuration or a post SQL configuration:

```
<Option1:Value1,Option2:Value2,...OptionN:ValueN>
```

The following table shows the configuration options and supported values that you can specify in a pre SQL configuration or post SQL configuration:

Options	Supported Values
DestinationDataset	Dataset ID in Google BigQuery
DestinationTable	Table name in Google BigQuery
FlattenResults	True and False
UseLegacySQL	True and False
WriteDisposition	WRITE_TRUNCATE, WRITE_APPEND, and WRITE_EMPTY

Configure Google BigQuery Source Session Properties

You can configure the session properties for a Google BigQuery source on the **Mapping** tab. Define the properties for the source instance in the session.

The following table describes the session properties that you can configure for a Google BigQuery source session:

Property	Description
Source Dataset ID	Optional. Overrides the Google BigQuery dataset name that you specified in the connection.
Source Table Name	Overrides the Google BigQuery table name that you specified in the source.
Number of Rows to Read	Specifies the number of rows to read from the Google BigQuery source.
Allow Large Results	Determines whether PowerExchange for Google BigQuery must produce arbitrarily large result tables to query large source tables. If you select this option, you must specify a destination table to store the query results.
Query Results Table Name	Required if you select the **Allow Large Results** option. Specifies the destination table name to store the query results. If the table is not present in the dataset, PowerExchange for Google BigQuery creates the destination table with the name that you specify.
Job Poll Interval in Seconds	The number of seconds after which PowerExchange for Google BigQuery polls the status of the read job operation. Default is 10.
Read Mode	Specifies the read mode to read data from the Google BigQuery source. You can select one the following read modes: - Direct. In direct mode, PowerExchange for Google BigQuery reads data directly from the Google BigQuery source table. **Note:** When you use hybrid and complex connection mode, you cannot use direct mode to read data from the Google BigQuery source. - Staging. In staging mode, PowerExchange for Google BigQuery exports the data from the Google BigQuery source table into Google Cloud Storage. After the download is complete, PowerExchange for Google BigQuery downloads the data from Google Cloud Storage into the local stage file that you specify into the local stage file and then reads data from the local stage file. Default is Direct mode.
Number of Threads for Downloading Staging Files	Specifies the number of files that PowerExchange for Google BigQuery downloads at a time to enable parallel download. This property applies to staging mode.

Property	Description
Data Format of the staging file	Specifies the data format of the staging file. You can select one of the following data formats: - JSON (Newline Delimited). Supports flat and record data with nested and repeated fields. - CSV. Supports flat data. **Note:** If you use hybrid and complex connection mode, you cannot use CSV format as the data format of the staging file. **Note:** Avro format is not applicable for PowerExchange for Google BigQuery.
Local Stage File Directory	Specifies the directory on your local machine where PowerExchange for Google BigQuery stores Google BigQuery source data temporarily before it reads the data. This property applies to staging mode.
Staging File Name	Name of the staging file where data from the Google BigQuery source table is exported to Google Cloud Storage. This property applies to staging mode.
Enable Staging File Compression	Indicates whether to compress the size of the staging file in Google Cloud Storage before PowerExchange for Google BigQuery reads data from the staging file. You can enable staging file compression to reduce cost and transfer time. This property applies to staging mode.
Persist Destination Table	Indicates whether PowerExchange for Google BigQuery must persist the query results table after it reads data from the query results table. By default, PowerExchange for Google BigQuery deletes the query results table
pre SQL	SQL statement that you want to run before reading data from the source. For example, if you want to select records in the database before you read the records from the table, specify the following pre SQL statement: `SELECT * FROM [api-project-80697026669:EMPLOYEE.DEPARTMENT] LIMIT 1000;`

pre SQL Configuration	Specify a pre SQL configuration.
	For example,
	`DestinationTable:PRESQL_SRC,DestinationDataset:EMPLOYEE,`
	`FlattenResults:False,WriteDisposition:WRITE_TRUNCATE,UseLegacySql:False`

post SQL	SQL statement that you want to run after reading data from the source.
	For example, if you want to update records in a table after you read the records from a source table, specify the following post SQL statement:
	`UPDATE [api-project-80697026669.EMPLOYEE.PERSONS_TGT_DEL]`
	`SET phoneNumber.number=1000011, phoneNumber.areaCode=100 where fullname='John Doe'SET phoneNumber.number =1000011, phoneNumber.areaCode=100 where fullname='John Doe'`

post SQL Configuration	Specify a post SQL configuration.
	For example,
	`DestinationTable:POSTSQL_SRC,DestinationDataset:EMPLOYEE,`
	`FlattenResults:True,WriteDisposition:WRITE_TRUNCATE,UseLegacySql:False`

Property	Description
SQL Override Query	Overrides the default SQL query used to extract data from the Google BigQuery source. **Note:** When you select staging mode and specify SQL override query, you must specify a dataset name in the Source Dataset ID advanced source property. Ensure that the list of selected columns, data types, and the order of the columns that appear in the query matches the columns, data types, and order in which they appear in the source object. **Note:** Ensure that you only map all the columns in the SQL override query to the target.
Use Legacy SQL for SQL Override	Uses legacy SQL override query. You can clear this option to define a standard SQL override query.

Configure Google BigQuery Target Session Properties

You can configure the session properties for a Google BigQuery target on the **Mapping** tab. Define the properties for the target instance in the session.

The following table describes the session properties that you can configure for a Google BigQuery target session:

Property	Description
UpdateMode	Determines the mode that PowerExchange for Google BigQuery uses to update rows in the Google BigQuery target. If you select an update mode, you need to select **Update** for the **Treat Source Rows As** session property in the **Properties** page. You can select one of the following modes: - Update As Update. PowerExchange for Google BigQuery updates all rows flagged for update if the entries exist. - Update Else Insert. PowerExchange for Google BigQuery first updates all rows flagged for update if the entries exist in the target. If the entries do not exist, PowerExchange for Google BigQuery inserts the entries. Default is Update As Update.
Target Dataset ID	Optional. Overrides the Google BigQuery dataset name that you specified in the target definition.
Target Table Name	Optional. Overrides the Google BigQuery target table name that you specified in the target definition.
Create Disposition	Specifies whether PowerExchange for Google BigQuery must create the target table if it does not exist. You can select one of the following values: - Create if needed. If the table does not exist, PowerExchange for Google BigQuery creates the table. - Create never. If the table does not exist, PowerExchange for Google BigQuery does not create the table.

Property	Description
Write Disposition	Specifies how PowerExchange for Google BigQuery must write data in bulk mode if the target table already exists. You can select one of the following values: - Write append. If the target table exists, PowerExchange for Google BigQuery appends the data to the existing data in the table. - Write truncate. If the target table exists, PowerExchange for Google BigQuery overwrites the existing data in the table. - Write empty. If the target table exists and contains data, PowerExchange for Google BigQuery displays an error and does not write the data to the target. PowerExchange for Google BigQuery writes the data to the target only if the target table does not contain any data. **Note:** Write disposition is applicable for bulk mode. Write disposition is applicable only when you perform an insert operation on a Google BigQuery target.
Write Mode	Specifies the mode to write data to the Google BigQuery target. You can select one of the following modes: - Bulk. In bulk mode, PowerExchange for Google BigQuery first writes the data to a staging file in Google Cloud Storage. When the staging file contains all the data, PowerExchange for Google BigQuery loads the data from the staging file to the BigQuery target. Google BigQuery then deletes the staging file unless you configure the mapping to persist the staging file. - Streaming. In streaming mode, PowerExchange for Google BigQuery directly writes data to the BigQuery target. PowerExchange for Google BigQuery writes the data into the target row by row. - CDC. Applies only when you capture changed data from a CDC source. In CDC mode, PowerExchange for Google BigQuery captures changed data from any CDC source and writes the changed data to a Google BigQuery target table. Default is Bulk mode.
Streaming Template Table Suffix	Specifies the suffix that PowerExchange for Google BigQuery adds to the individual target tables that it creates based on the template target table. This property applies to streaming mode.
Rows per Streaming Request	Specifies the number of rows that PowerExchange for Google BigQuery streams to the BigQuery target for each request. Default is 500 rows. The maximum row size that PowerExchange for Google BigQuery can stream to the BigQuery target for each request is 10 MB. This property applies to streaming mode.
Staging File Name	Name of the staging file that PowerExchange for Google BigQuery creates in the Google Cloud Storage before it loads the data to the Google BigQuery target. This property applies to bulk mode.
Data Format of the staging file	Specifies the data format of the staging file. You can select one of the following data formats: - JSON (Newline Delimited). Supports flat and record data with nested and repeated fields. - CSV. Supports flat data. **Note:** If you use hybrid and complex connection mode, you cannot use CSV format as the data format of the staging file.

Property	Description
Persist Staging File After Loading	Indicates whether PowerExchange for Google BigQuery must persist the staging file in the Google Cloud Storage after it writes the data to the Google BigQuery target. You can persist the staging file if you want to archive the data for future reference. By default, PowerExchange for Google BigQuery deletes the staging file in Google Cloud Storage. This property applies to bulk mode.
Enable Staging File Compression	Select this option to compress the size of the staging file before PowerExchange for Google BigQuery writes the data to the Google Cloud Storage and decompress the staging file before it loads the data to the Google BigQuery target. You can enable staging file compression to reduce cost and transfer time.
Job Poll Interval in Seconds	The number of seconds after which PowerExchange for Google BigQuery polls the status of the write job operation. Default is 10.
Number of Threads for Uploading Staging file	The number of files that PowerExchange for Google BigQuery must upload to Google Cloud Storage in bulk mode.
Local Stage File Directory	Specifies the directory on your local machine where PowerExchange for Google BigQuery stores the files temporarily before writing the data to the staging file in Google Cloud Storage. This property applies to bulk mode.
Allow Quoted Newlines	Indicates whether PowerExchange for Google BigQuery must allow the quoted data sections with newline character in a .csv file.
Field Delimiter	Delimiter character for the fields in a .csv file.
Quote Char	Specifies the quote character that defines the boundaries of text strings in a .csv file. You can configure parameters such as single quote or double quote.
Allow Jagged Rows	Indicates whether PowerExchange for Google BigQuery must accept the rows without trailing columns in a .csv file.
Pre SQL	SQL statement that you want to run before writing data to the target. For example, if you want to select records from the database before you write the records into the table, specify the following pre SQL statement: `SELECT * FROM `api-project-80697026669.EMPLOYEE.RegionNation` LIMIT 1000`
Pre SQL Configuration	Specify a pre SQL configuration. For example, `DestinationTable:PRESQL_TGT2,DestinationDataset:EMPLOYEE, FlattenResults:False,WriteDisposition:WRITE_TRUNCATE,UseLegacySql:False`
Post SQL	SQL statement that you want to run after writing the data into the target. For example, if you want to update records in a table after you write the records into the target table, specify the following post SQL statement: `UPDATE [api-project-80697026669.EMPLOYEE.PERSONS_TGT_DEL] SET phoneNumber.number =1000011, phoneNumber.areaCode=100 where fullname='John Doe'`

Property	Description

Post SQL Configuration	Specify a post SQL configuration.
	For example,
	`DestinationTable:POSTSQL_SRC,DestinationDataset:EMPLOYEE,`
	`FlattenResults:True,UseLegacySQL:False`

Enable Merge	Implements the Merge query to perform update, upsert, or delete operations on the Google BigQuery target table. Default is not selected.
Treat Empty String as Null	Determines whether PowerExchange for Google BigQuery must treat empty strings in the source as null in the Google BigQuery target.
INSERT	Inserts all rows to the Google BigQuery target.
DELETE	Deletes rows from the Google BigQuery target. If you select DELETE, you need to select **Delete** for the **Treat Source Rows As** session property in the **Properties** page.
UPDATE	Not applicable for PowerExchange for Google BigQuery. **Note:** Configure the update strategy for a target object in the **UpdateMode** session property.
Success File Directory	Not applicable for PowerExchange for Google BigQuery.
Error File Directory	Not applicable for PowerExchange for Google BigQuery.
Forward Rejected Rows	Not applicable for PowerExchange for Google BigQuery.

Using Merge Query for Update, Upsert, and Delete Operations

You can implement the Merge query to perform the update, upsert, and delete operations on a Google BigQuery target in a single SQL statement.

To implement the Merge query, select the **Enable Merge** option in the target session properties.

Rules and Guidelines

Consider the following rules and guidelines when you use the Merge query:

- When you configure a Google BigQuery connection to use the simple connection mode and select CSV as the staging file format, you must not mark any columns of the Record data type for update or upsert operation.
- When you configure a Google BigQuery connection to use the simple connection mode, the Google BigQuery target table must not contain repeated columns.
- When you configure a Google BigQuery connection to use hybrid connection mode, the Google BigQuery target table must not contain repeated columns as the key field.

Lookups

You can create a connected or unconnected Lookup transformation.

You can create a connected cached lookup or unconnected cached lookup. When you create a cached lookup, the performance increases because the Integration Service caches a large lookup source or small lookup tables. When you cache the lookup source, the Integration Service queries the lookup cache instead of querying the lookup source for each input row.

Connected Lookups

You can create a connected Lookup transformation to perform a lookup on a Google BigQuery table. A connected Lookup transformation has a source qualifier as the lookup source.

When you configure a connected Lookup transformation, the lookup source and source qualifier are in a different connected from the Lookup transformation. The source and source qualifier are in a partial connected that contains no target. The PowerCenter Integration Service reads the source data in this connected and passes the data to the Lookup transformation to create the cache.

For more information, see the topic "Pipeline Lookups" in the *PowerCenter Transformation Guide*.

Unconnected Lookups

You can configure an unconnected lookup transformation for the source qualifier in a mapping.

An unconnected Lookup transformation is a Lookup transformation that is not connected to any source, target, or transformation in the pipeline.

An unconnected Lookup transformation receives input values from the result of a :LKP expression in another transformation. The Integration Service queries the lookup source or cache based on the lookup ports and condition in the Lookup transformation and passes the returned value to the port that contains the :LKP expression. The :LKP expression can pass lookup results to an expression in another transformation.

The syntax for the lookup expression is `:LKP.lookup_transformation_name(argument, argument, ...)`

The order in which you list each argument must match the order of the lookup conditions in the Lookup transformation. The Lookup transformation returns the result of the query through the Lookup transformation return port. The transformation that calls the lookup receives the lookup result value in the port that contains the :LKP expression. If the lookup query fails to return a value, the port receives a null value.

When you perform an unconnected lookup, you can perform the same lookup multiple times in a mapping. You can test the results of the lookup in another expression and filter rows based on the results.

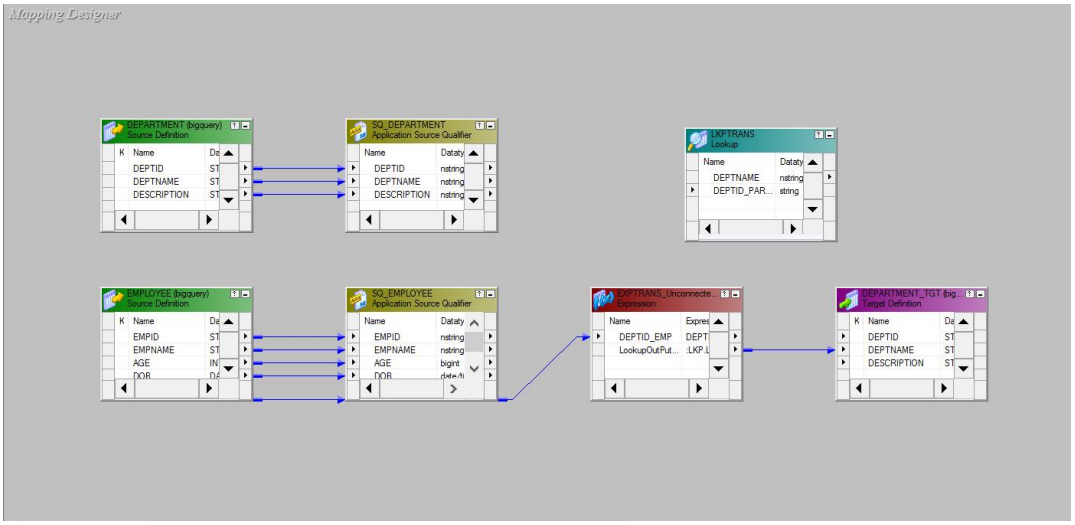
1. An unconnected Lookup transformation receives input values from the result of a :LKP expression in another transformation, such as an Aggregator transformation or Expression transformation.
2. The Integration Service queries the lookup source or cache based on the lookup ports and condition in the Lookup transformation.
3. The Integration Service returns a value through the return port of the Lookup transformation.
4. The Integration Service passes the return value to the port that contains the :LKP expression.

Configuring an unconnected Lookup Transformation in a Google BigQuery Mapping

A mapping that contains an unconnected Lookup transformation receives input values from the result of a :LKP expression in another transformation, such as an Aggregator transformation, Expression

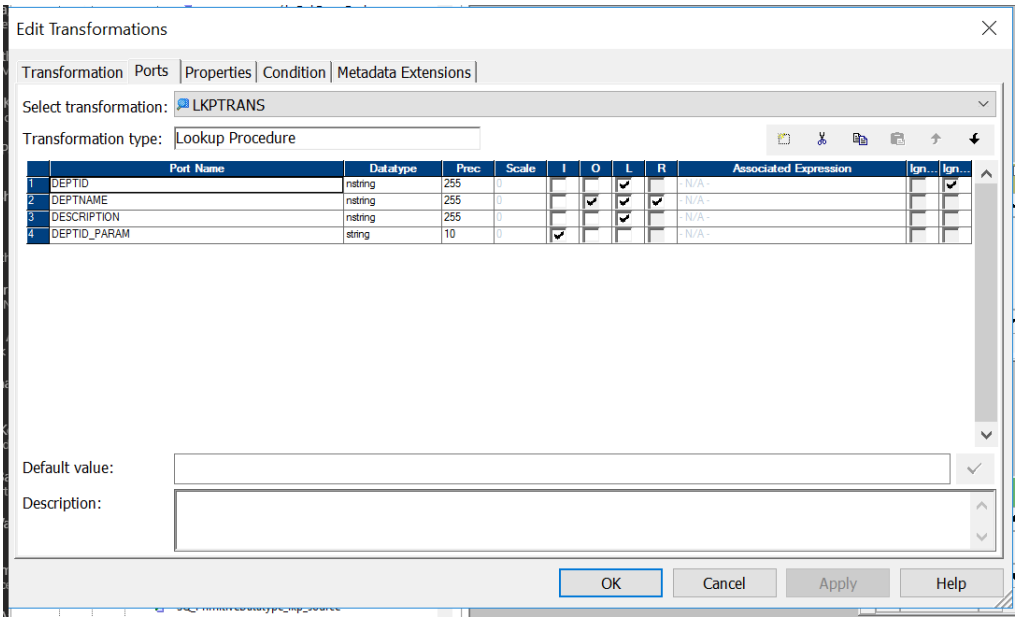
transformation, or Update Strategy transformation. The Integration Service queries the lookup source or cache based on the lookup ports and condition in the Lookup transformation. The Integration Service returns a value through the return port of the Lookup transformation. The Integration Service passes the return value to the port that contains the :LKP expression.

The following mapping shows a mapping that contains an unconnected Lookup transformation and the Expression transformation that processes the lookup expression:



Perform the following tasks to configure an unconnected lookup transformation:

1. Create an input port in the Lookup transformation for each argument in the :LKP expression. You can create a different port for each condition, or use the same input port in more than one condition.
   In this case, create a lookup condition that compares the DEPTID_EMP in the source with the DEPTID in the target. Retrieve the DEPTNAME for each employee in the source with respect to the department name in the target. Create an input port (DEPTID_PARAM) with the data type String (10,0) to match the DEPTID_PARAM lookup port.
2. After you configure the ports, define a lookup condition to compare transformation input values with values in the lookup source or cache. To increase performance, add conditions with an equal sign first. In this case, add the following lookup condition:
   ```
   DEPTID = DEPTID_PARAM
   ```
3. Designate one lookup/output port as a return port. In this case, you can define the DEPTNAME port as the return port.
   The following image shows a return port in a Lookup transformation:



Note: Do not configure the lookup policy to use all values on multiple matches for an unconnected Lookup transformation.

4. Supply input values for an unconnected Lookup transformation from a :LKP expression in another transformation. The arguments are local input ports that match the Lookup transformation input ports used in the lookup condition. In this case, add the following :LKP expression in the Expression transformation:

```
:LKP.LKPTRANS(DEPTID_EMP)
```

Rules and Guidelines for Google BigQuery Lookups

Consider the following rules and guidelines when you configure a lookup transformation in a Google BigQuery session:

- You cannot configure an uncached lookup transformation in a Google BigQuery session.
- When you configure a Lookup transformation, you must select the Google BigQuery lookup table for the Lookup transformation in the **Source** tab or the **Source Qualifier** tab.
- When you use a Google BigQuery connection in simple mode and the Lookup table contains multiple columns of repeated data type, you must specify one of the columns of repeated data type in the lookup condition and delete the remaining columns of repeated data type in the Lookup transformation and in the source qualifier.
- When you configure the **On multiple matches** property for a Lookup trasformation, the **Return first row** or **Return last row** options are not applicable.

Partitioning

When you read from Google BigQuery, you can configure pass-through partitioning to optimize the session performance at run time.

When you specify pass-through partitioning for a Google BigQuery Source Qualifier transformation, you can specify filter conditions in the Google BigQuery session properties to override the filter condition you specify in the source qualifier. The PowerCenter Integration Service uses the filter condition you specify in the session properties when it filters data from the source.

To configure pass-through partitioning, select the Source Qualifier transformation, and add a partition point from the **Mapping** tab of the session properties. Add the number of partitions you require and select the partition type as **Pass Through** for each of the partitions.

Based on the number or partitions you add, the PowerCenter Integration Service adds those many number of partition fields for the **Filter Override**, **Filter Override**, **Local Staging File Directory**, **SQL Override Query**, and **Use Legacy SQL for SQL override** property in the session properties. Specify the filter override condition for each of the partitions. The PowerCenter Integration Service uses the filter conditions you specify to pass data through the appropriate partition.

You can configure filter override condition for the columns of the following data types:

- Integer
- Sting
- Timestamp
- Numeric

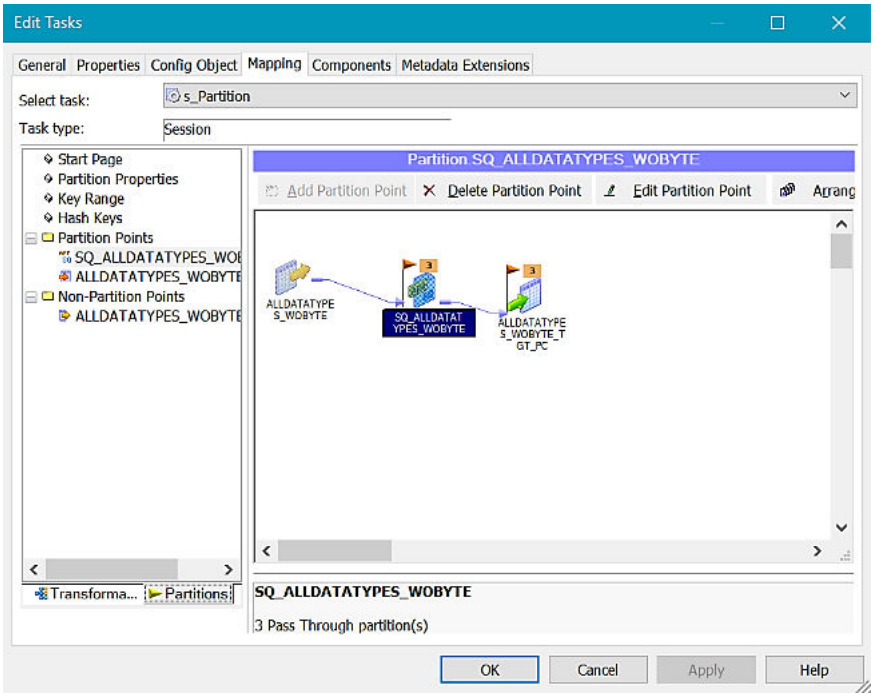
  Note: When you define filter override condition for columns of Numeric data type, you must ensure that you use a Google BigQuery connection in hybrid connection mode.

Configuring Pass-through Partitioning for a Google BigQuery Session

Configure pass-through partitioning for a Google BigQuery session. After you add the number of partitions, you can specify a filter override condition for each of the partitions. Alternatively, you can specify a SQL override query instead of a filter override condition for each of the partitions. When you configure pass-through partitioning, you can also specify a different staging file directory for each of the partitions.

1. In the **Workflow Designer**, open the session properties.
2. In the **Partitions** view, click **Add Partition Point**.

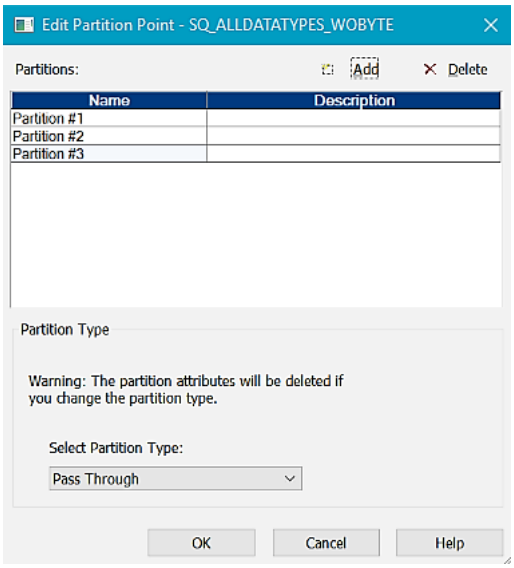
The transformation name appears under the **Partition Points** node.



3. Select the Source transformation, and click **Edit Partition Point**.

   The **Edit Partition Point** dialog box appears.

4. Click **Add** to add the partitions and enter a description for each partition.
5. Select the partition type as **Pass Through** for each of the partitions.

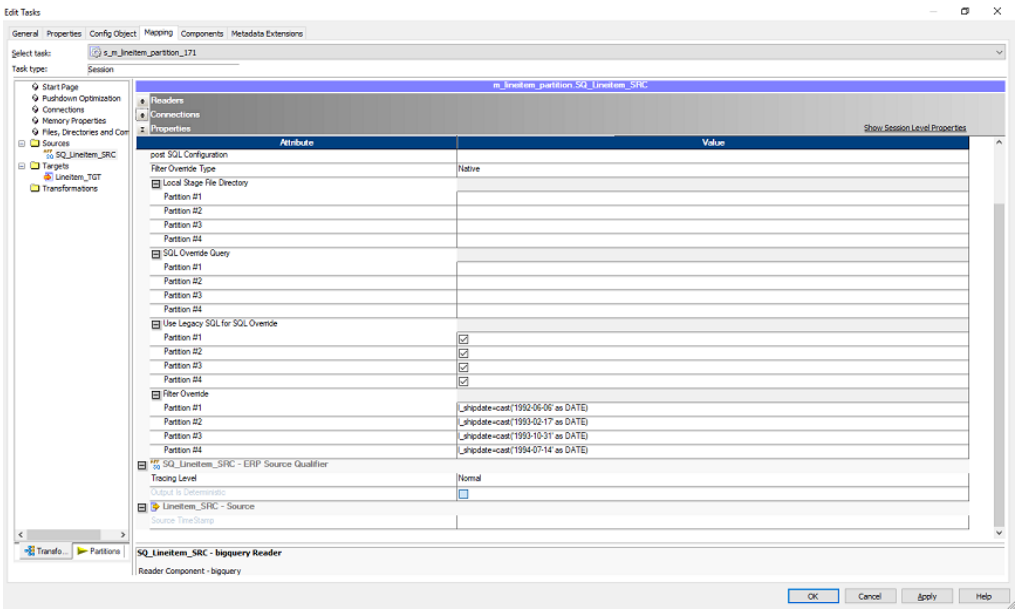


6. Click **OK**.
7. Click the **Transformations** tab in the session properties.

   Based on the number of partitions you create, those many number of partition fields appear for the **Filter Override**, **Local Staging File Directory**, **SQL Override Query**, and **Use Legacy SQL for SQL override** property.

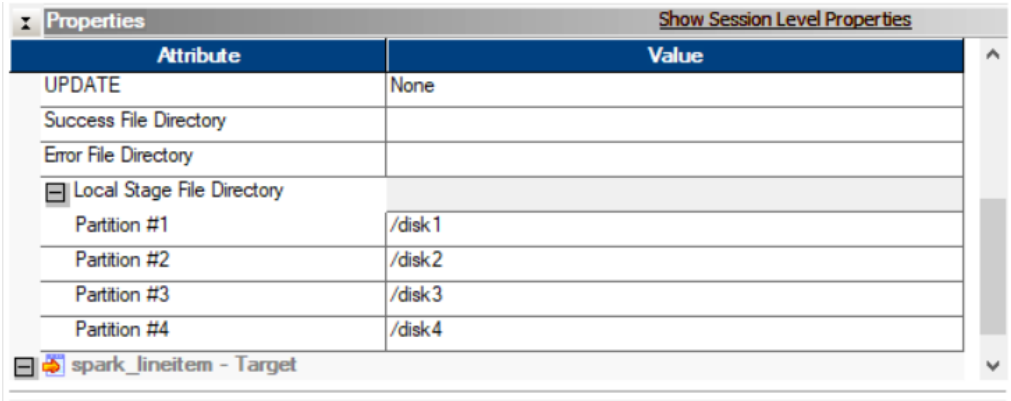
8. Specify the filter override condition for each partition.

   For example, in the following image, you can configure filter override conditions for each of the four partitions configured:



   You can also specify local staging file directory for each of the partitions. If you do not specify a local staging file directory for a partition, the PowerCenter Integration Service considers the default staging file directory.

   For example, in the following image, you can configure local staging file directory for each of the four partitions configured:



9. Click **OK**.

Configure the Java Heap Memory

When the source or target contains a large amount of data, configure the memory for the Java heap size on the node that runs the PowerCenter Integration Service. You must ensure that the minimum physical memory available on the server machine for each session is in the range of 300 to 500 MB.

You must set a maximum heap size value based on the amount of data that you want to process.

1. In the Administrator tool, navigate to the PowerCenter Integration Service for which you want to change the Java heap size.
2. On the **Processes** tab, edit the **General Properties** section.
3. Specify the maximum heap size in Java SDK Maximum Memory limit based on the amount of data you want to process.
4. Click **Ok**.
5. Restart the PowerCenter Integration Service.

Rules and Guidelines for Google BigQuery Sessions

Use the following rules and guidelines when you create a session:

- The connection mode that you specify in the Google BigQuery source or target connection during runtime must match with the connection mode that you specify when you import a Google BigQuery source or target definition.
- You cannot configure key range partitioning to read data from a Google BigQuery source.
- You cannot read hierarchical data from a Google BigQuery source, convert the hierarchical input to relational output, and write data to relational targets.
- You cannot read data from relational sources, convert the relational input to hierarchical output, and write data to a Google BigQuery target.
- You cannot create a target at run time.
- You must define a primary key in the target table for update and delete operations.
  If you do not define a primary key in the target table, the mapping fails to update records in or delete records from the target table.
- To perform an upsert operation on a column, you must ensure that the column in the target table is set to Required in Google BigQuery.
- When you perform an upsert operation on a Google BigQuery target table, you cannot perform an update operation on multiple rows.
- You cannot perform more than one operation with a pre SQL or post SQL command.
- When you perform an upsert operation on a Google BigQuery target table, you must not select streaming mode to write data to a Google BigQuery target.
- When you configure a Google BigQuery connection in simple mode, you can perform an upsert operation on columns of primitive data types only.
- If you perform an UPDATE or DELETE operation with a pre SQL or post SQL command, you must specify the following parameter in the pre SQL configuration or post SQL configuration:
  `UseLegacySQL:False`

- When you use staging mode to read data from Google BigQuery or bulk mode to write data to Google BigQuery, you must increase the maximum heap size in the **Java SDK Maximum Memory** property to 2048 MB or 4096 MB based on the amount data you want to process.
- When you use the `pmrep` command to create a Google BigQuery connection, you must specify one of the following values for the **Connection mode** option:
  - Simple
  - Hybrid
  - Complex

CHAPTER 6

Google BigQuery as CDC Target

This chapter includes the following topics:

CDC Target Overview

You can read the real-time or changed data from a Change Data Capture (CDC) source and load the data to Google BigQuery.

Create a PowerExchange for Google BigQuery connection to access Google BigQuery and write the data from a CDC source.

When the session processes the changed data from a CDC source such as PowerExchange Express CDC for Oracle, PowerExchange for Google BigQuery creates a state table and a staging table in Google BigQuery. When the changed data is received from the CDC source, PowerExchange for Google BigQuery uploads the changed data to the staging table. Then, it generates a `Job_Id` and writes the `Job_Id` to the state table along with the restart information. PowerExchange for Google BigQuery then merges the stage table with the actual target table in Google BigQuery.

Each time you run the session, PowerExchange for Google BigQuery creates the state table to store the state information. PowerExchange for Google BigQuery uses the following naming convention for the state table name:

```
state_tables_<first 20 characters of the workflow name OR application name specified in the CDC connection>_<session_ID>
```

Similarly, PowerExchange for Google BigQuery uses the following naming convention for the staging table name:

```
staging_table_cdc_<dataset_name>_<targetTable_name>_<session_name>
```

CDC Target Example

You work for a rapidly growing data science organization. Your organization develops software products to analyze financials, building financial graphs connecting people profiles, companies, jobs, advertisers, and publishers. The organization uses infrastructure based on Google Cloud Platform and stores its data in various data sources such as Oracle. 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 changed data stored in the Oracle data source to Google BigQuery, which is a highly scalable, cost-effective and fully managed enterprise data warehouse on a regular interval of time. Use Google Analytics to run high-performance analytics.

You can use a PowerExchange Express CDC for Oracle real-time connection to read changed data from the Oracle database. To write this large amount of data, you can use the PowerExchange for Google BigQuery connection.

To write changed data to a Google BigQuery object, perform the following steps:

1. Import the PowerExchange Express CDC for Oracle source object in PowerCenter Designer and create a PowerExchange for Oracle CDC real-time source connection.
2. Create or import a Google BigQuery target object and create a PowerExchange for Google BigQuery connection.
3. Create a mapping.
4. Create a session and configure the session properties.
5. Run the session.

Import Source and Target Objects

Import the PowerExchange Express CDC for Oracle source from which you want to read the data. The imported object appears in the **Source Analyzer**.

Create a connection object to connect to the PowerExchange Express CDC for Oracle source in the **Workflow Manager**. When creating a connection for the CDC source, you can configure the connection attributes, such as idle time, maximum rows per commit, and minimum rows per commit. When you configure the connection in the session properties, the PowerCenter Integration Service uses this connection to connect to the CDC source.

Similarly, create or import the Google BigQuery target object to write data. The imported object appears in the **Target Designer**. Create a connection object to connect to the Google BigQuery target object in the **Workflow Manager**

After importing the objects, create a mapping with the required transformations.

Create a Mapping

If the PowerExchange Express CDC for Oracle source contains rows flagged with insert, update, delete, upsert operations, drag the imported object to the Mapping Designer and design your logic based on source columns.

If the rows in the PowerExchange Express CDC for Oracle source are not flagged with required operations, use the Update Strategy transformation to flag the rows for insert, update, delete, or upsert operation as required.

Note: You must define a key column in the Google BigQuery target definition. While reading from a CDC source, you must define and map the key column in the CDC source to key column in the Google BigQuery target object. Else, the session fails.

Create a Session

After you create a mapping in the Designer, you create a session in the **Workflow Manager**.

In a session, you define properties that determine how the PowerCenter Integration Service extracts data from or loads data to the data source.

- Configure the following session properties in the **Properties** tab:
  - **Treat source rows as**. Select Data driven.
  - **Commit on End of File**. Deselect the check box.
  - **Commit Type**. Select Source.
  - **Recovery Strategy**. Select Resume from last checkpoint.
- Configure the following source session properties in the **Mapping** tab:
  - **Connections**. Select the PowerExchange for Oracle CDC real-time source connection to access the PowerExchange Express CDC for Oracle source.
- Configure the following target session properties in the **Mapping** tab:
  - **Connections**. Select the PowerExchange for Google BigQuery connection to access the Google BigQuery target.
  - **Write Mode**. Select CDC.

After the session is created, run the session.

Rules and guidelines for Google BigQuery CDC target

Consider the following limitations when working with a Google BigQuery target:

- When you use a Google BigQuery connection in complex mode, you cannot write changed data from a CDC source to a Google BigQuery target.
- When you capture changed data from a CDC source, you can only configure a single Google BigQuery target definition in a session. You cannot configure multiple Google BigQuery targets to write changed data from a CDC source.
- You cannot configure multiple pipelines in a workflow to write changed data from multiple CDC sources to multiple Google BigQuery targets.
- You must define a column as required in the Google BigQuery target table.
- If you define a column as required in the Google BigQuery target table, you must map a column in the CDC source to the required column in the Google BigQuery target in the mapping.

- When you map a column in the CDC source to a required column in the Google BigQuery target, you must ensure that the column in the CDC source does not contains NULL values. Otherwise, the session fails.
- You can only configure the following target session properties for CDC mode:
  - Target Dataset ID
  - Target Table Name
  - Job Poll Interval in Seconds
  - Pre SQL
  - Pre SQL Configuration
  - Post SQL
  - Post SQL Configuration
- Informatica recommends that the PowerCenter Integration Service, the CDC source and PowerExchange for CDC are configured in the same region as Google BigQuery.
- To increase performance and avoid run-time environment memory issues, increase the Java heap size in the JVM options for PowerCenter Integration Service. Set `JVMOption1` to `-Xmx1024m` in the **Custom Properties** section of the **Processes** tab of the PowerCenter Integration Service.
- To improve performance, specify a higher commit interval for the **Maximum Rows Per Commit** property in the CDC source. However, in case of failure, recovery takes more time for a higher commit interval.
- Informatica recommends to use update queries on the CDC source database only if the Google BigQuery target table is partitioned and clustered.

CHAPTER 7

Google BigQuery Pushdown Optimization

This chapter includes the following topics:

Pushdown Optimization Overview

To optimize the performance of a Google BigQuery session in PowerCenter, you can push the transformation logic to the Google BigQuery source or target database. You can use full or source pushdown optimization for the Google BigQuery session that uses the ODBC connection type.

Pushdown Optimization Functions

When you use pushdown optimization, the PowerCenter Integration Service converts the expression in the transformation or workflow link by determining equivalent operators and functions in the database.

PowerExchange for Google BigQuery supports the following pushdown functions in a Google BigQuery 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

Function	Pushdown	Function	Pushdown	Function	Pushdown
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()	X
CHOOSE()	-	LEAST()	-	RTRIM()	X
CHR()	X	LENGTH()	X	SET_DATE_PART()	-
CHRCODE()	-	LN()	-	SIGN()	-
COMPRESS()	-	LOG()	-	SIN()	X
CONCAT()	X	LOOKUP	X	SINH()	-
COS()	X	LOWER()	X	SOUNDEX()	-
COSH()	-	LPAD()	X	SQRT()	X
COUNT()	X	LTRIM()	X	STDDEV()	-
CRC32()	-	MAKE_DATE_TIME()	-	SUBSTR()	X
CUME()	-	MAX()	X	SUM()	X
DATE_COMPARE()	X	MD5()	-	SYSDATE()	X
DATE_DIFF()	X	MEDIAN()	-	SYSTIMESTAMP()	X
DECODE()	X	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

Function	Pushdown	Function	Pushdown	Function	Pushdown
IIF()	X	RAND()	-	TRUNC(NUMBER)	X
IN()	X	RATE()	-	UPPER()	X
INDEXOF()	-	REG_EXTRACT()	-	VARIANCE()	-

Note: Columns marked with an X indicate that the PowerCenter Integration Service can push the function to the Google BigQuery database by using source-side or full pushdown optimization. Columns marked with a dash (-) symbol indicate that the PowerCenter Integration Service cannot push the function to the database.

Rules and Guidelines for Functions in Pushdown Optimization

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

- 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 IS_DATE() or LAST_DAY() function to the Google BigQuery database, you must define the arguments of the Date data type.
- To push the 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) function to the Google BigQuery database, you must define the arguments of the Timestamp data type in the following format:
  - D
  - DD
  - DDD
  - DY
  - HH
  - HH24
  - MI
  - SS
  - MS
- To push the TRUNC(DATE) function to the Google BigQuery database, you must define the arguments of the Timestamp data type in the following format:
  - Y
  - YY

- YYY
- YYYY
- MM
- MON
- D
- DD
- DDD
- DY
- HH
- HH24
- MI
- SS
- MS
- US

- To push the TO_CHAR(DATE) function to the Google BigQuery database, you must define the arguments of the Timestamp data type in the following format:
  - YYYY
  - MM
  - DD
  - HH24
  - MI
  - MS
  - -
  - /
  - .
  - ;
  - :
  - "text"
- When you define arguments of the Timestamp data type in the TO_DATE() function, you must use the following format:
  - YYYY
  - MM
  - DD
  - HH24
  - MI
  - SS
  - MS
  - US
  - -
  - /

- .
- ;
- :

- When you define arguments of the Timestamp data type in the GET_DATE_PART() function, you must use the following format:
  - Y
  - YY
  - YYY
  - YYYY
  - MM
  - MON
  - D
  - DD
  - DDD
  - DY
  - HH
  - HH24
  - MI
  - SS
  - MS
  - US
- When you push the TO_DATE() function to the Google BigQuery database, you must map the output to a Timestamp column in the Google BigQuery table.
- When you push the SYSTIMESTAMP() function to the Google BigQuery database, do not specify any format. The Google BigQuery database returns the complete timestamp.
- Ensure that you do not specify an in-out parameter of Date or Time data type. Otherwise, the session fails.
- If the schema or table name starts with a number, then you must enclose the schema name or table name within single quotation marks.
- When you configure the Lookup Source Filter or Lookup SQL Override property in a Lookup transformation, you must select the **Allow Temporary View for Pushdown** property under the session properties.
- When you configure the **Allow Temporary View for Pushdown** in the session, you must have the bigquery.tables.create and bigquery.tables.delete to permissions create and drop views.

Pushdown Optimization Transformations, Operators, and Data Types

When you use pushdown optimization, the PowerCenter Integration Service converts the expression in the transformation or workflow link by determining equivalent operators and functions in the database.

PowerExchange for Google BigQuery supports the following transformations for pushdown optimization:

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	Source, Full
Update Strategy	Source, Full

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

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

PowerExchange for Google BigQuery supports the following data types for pushdown optimization:

Google BigQuery Data Type	Transformation Data Type
DATE	Date.
DATETIME	DateTime
FLOAT	Double
INT	BigInt
NUMERIC	Decimal
STRING	String

Google BigQuery Data Type	Transformation Data Type
BOOLEAN	Integer
BYTE	Byte
DATE	DateTime
DATETIME	DateTime Applicable only for full pushdown.
FLOAT	Double

Google BigQuery Data Type	Transformation Data Type
INTEGER	BigInt
NUMERIC	Decimal Applicable only for full pushdown. **Note:** Google BigQuery ODBC connection supports maximum precision of 28 and maximum scale of 9.
RECORD	String Applicable only for full pushdown.
STRING	String
TIME	DateTime
TIMESTAMP	DateTime

Add the EXTODBC.DLL Entry

Before you use the Google BigQuery ODBC connection and configure pushdown optimization, you must add the required ODBCDLL entry in the `powrmart.ini` file of the PowerCenter Client.

1. Navigate to the following directory:

   ```
   <Informatica installation directory>\clients\PowerCenterClient\client\bin
   ```

2. In the `powrmart.ini` file, add the following entry:

   ```
   Bigquery=EXTODBC.DLL
   ```

Google BigQuery Pushdown through ODBC Connection

Use an ODBC connection to enable full or source pushdown optimization when you want to read data from a Google BigQuery source and write to an Google BigQuery target. The mapping logic is processed entirely in Google BigQuery.

When you run a task configured for pushdown optimization, the task converts the transformation logic to an SQL statement. The task sends the SQL statement to the database and the database executes the SQL statement.

Example

You work for a rapidly growing data science organization. Your organization develops software products to analyze financials, building financial graphs connecting people profiles, companies, jobs, advertisers, and publishers. The organization uses infrastructure based on Google Cloud Platform and stores its data in Google BigQuery, a petabyte scale data warehouse. 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 to the business intelligence service. You can use PowerExchange for Google BigQuery to read data from Google BigQuery. To read this large amount of data, you can use source pushdown for the ODBC connection type. Using the ODBC connection type with pushdown optimization enhances the performance.

Prequisites

When you run a session 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 session, you must have the bigquery.tables.create and bigquery.tables.delete to permissions 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 the Google BigQuery ODBC driver to enhance the session 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, set the value of the **Pushdown Optimization** property as **Full** in the **Performance** section on the **Properties** tab of the session.

You cannot configure target-side pushdown optimization by using Google BigQuery ODBC driver.

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

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 PowerCenter Integration Service 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.

   For more information about downloading the Informatica Google BigQuery ODBC (64-bit) driver, `Register.bat` file, and the `DriverInstallHelper.jar` file, 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`

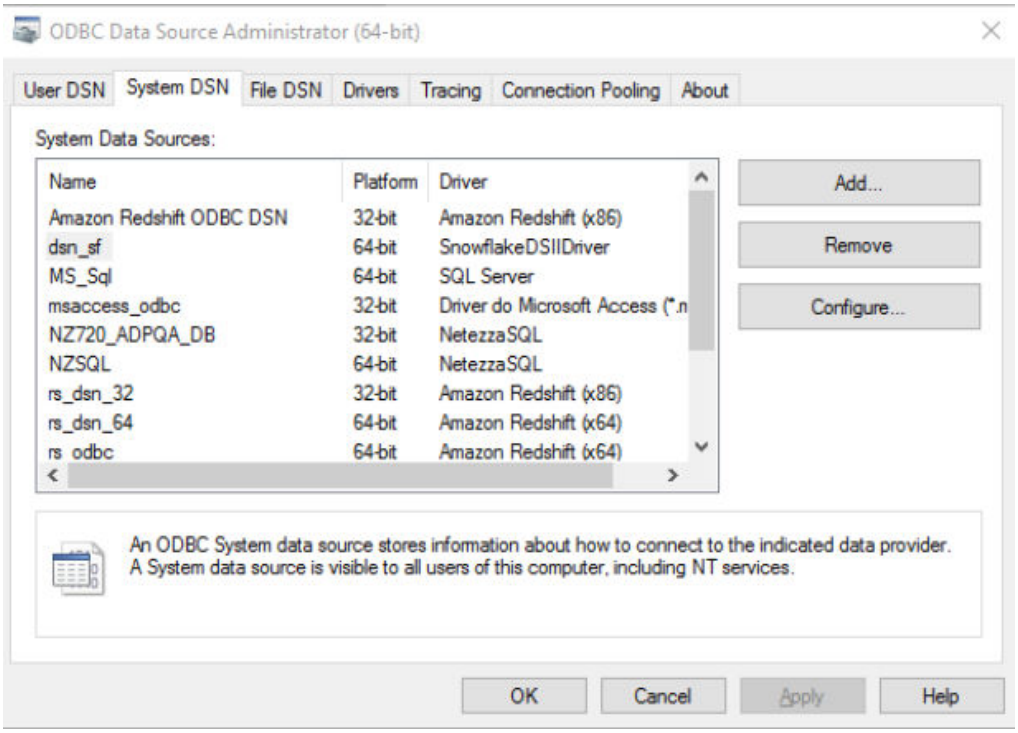
9. Specify the following directory for the **Input Directory Containing Installation Files** and press **Enter**:

   `<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.

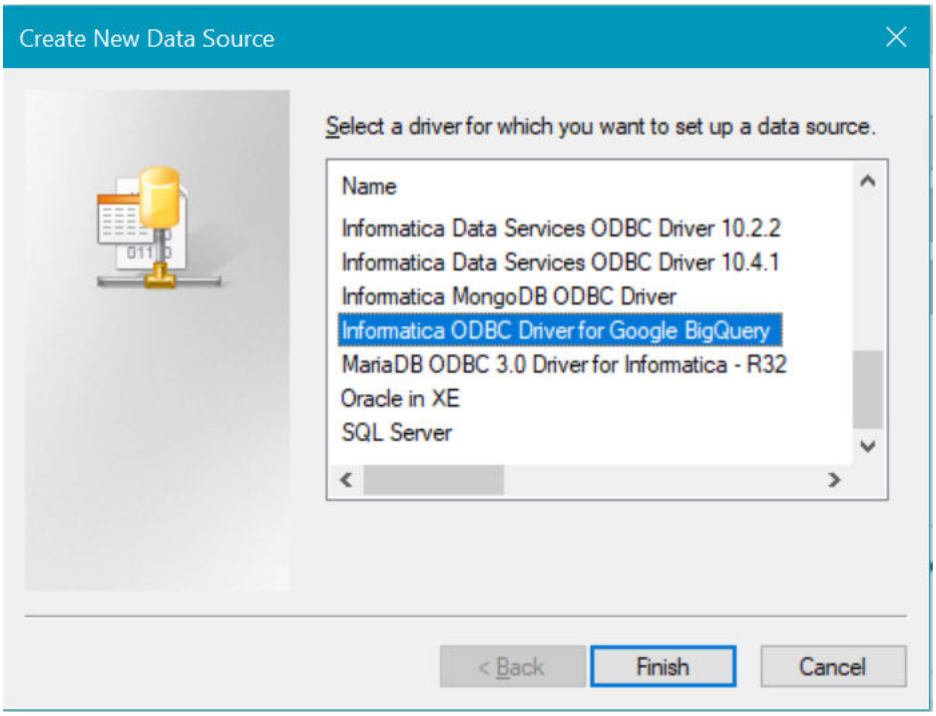
    The **System DSN** tab appears.
    The following image shows the System DSN tab on the ODBC Data Source Administrator (64-bit) dialog box:



14. Click **Add**.

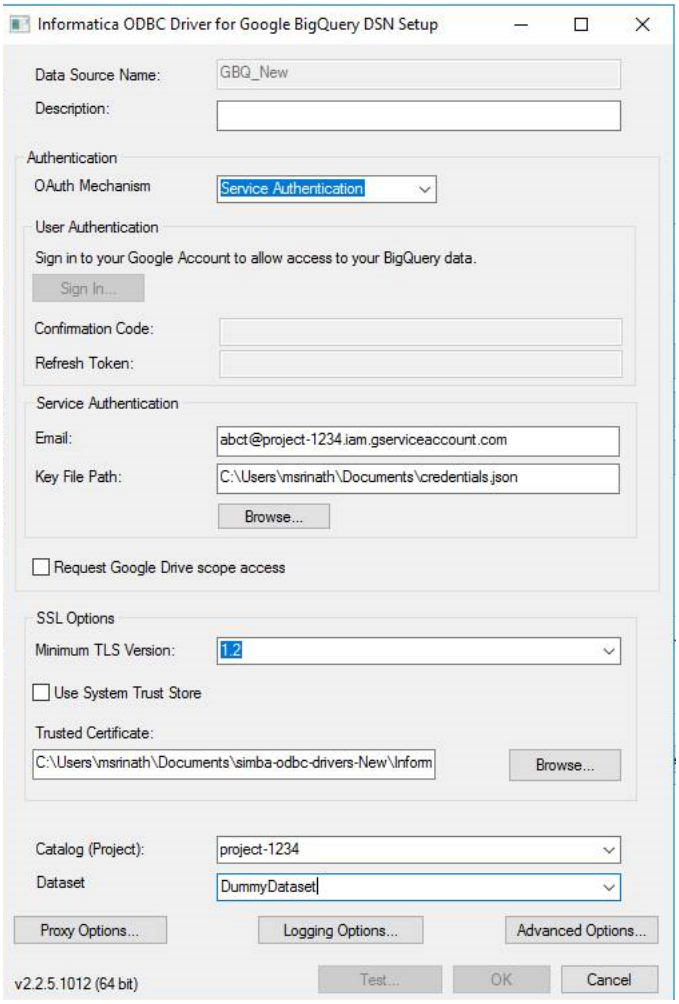
    The **Create New Data Source** dialog appears.

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.
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.

Property	Description
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.

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 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 PowerCenter Integration Service 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.

   For more information about downloading the Informatica Google BigQuery ODBC (64-bit) driver, `Register.bat` file, and the `DriverInstallHelper.jar` file, 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**.

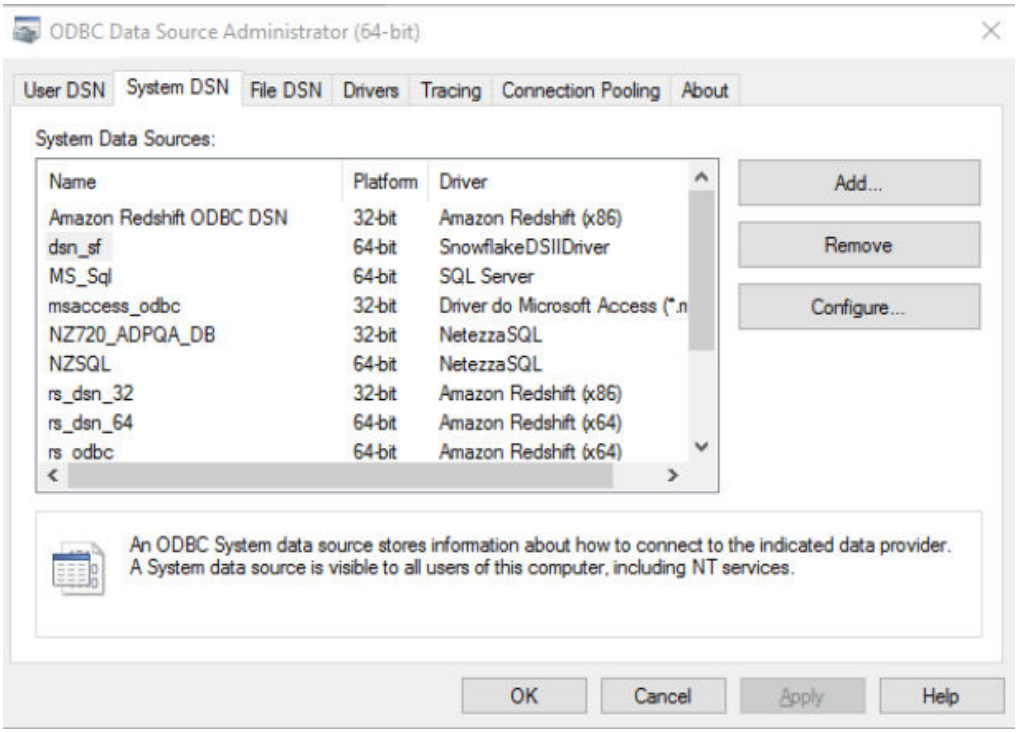
12. Click **Data Sources (ODBC)**.

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

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

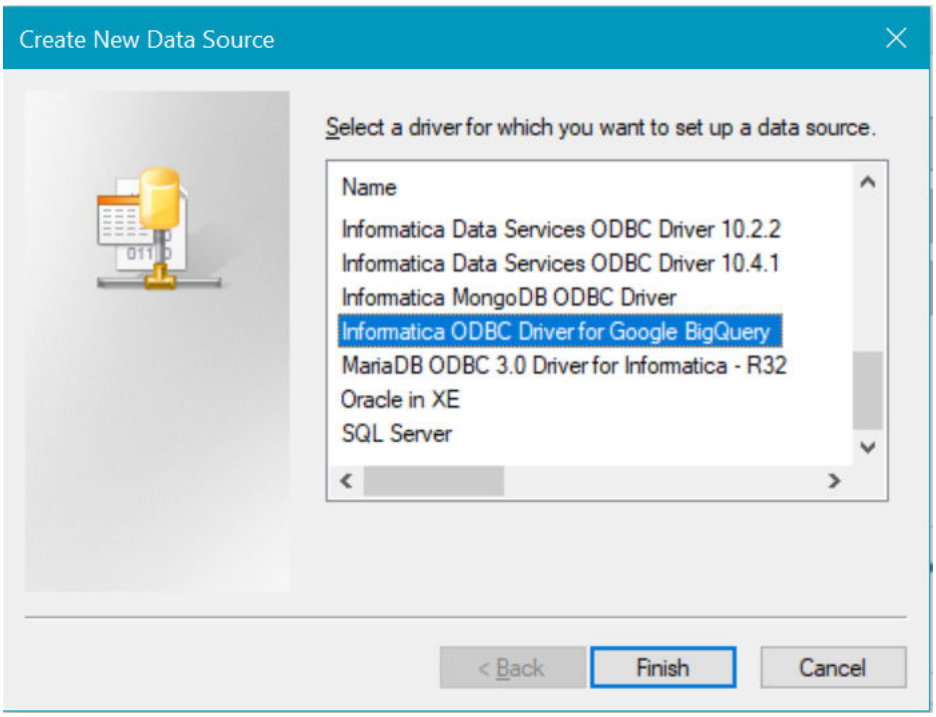
    The **System DSN** tab appears.

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



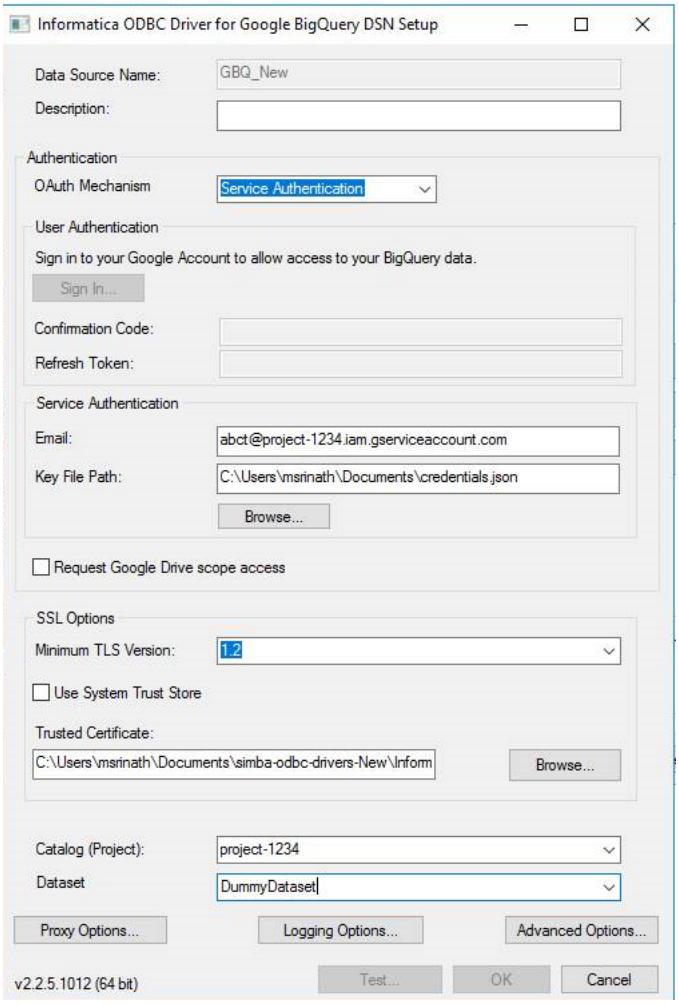
14. Click **Add**.

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.

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.

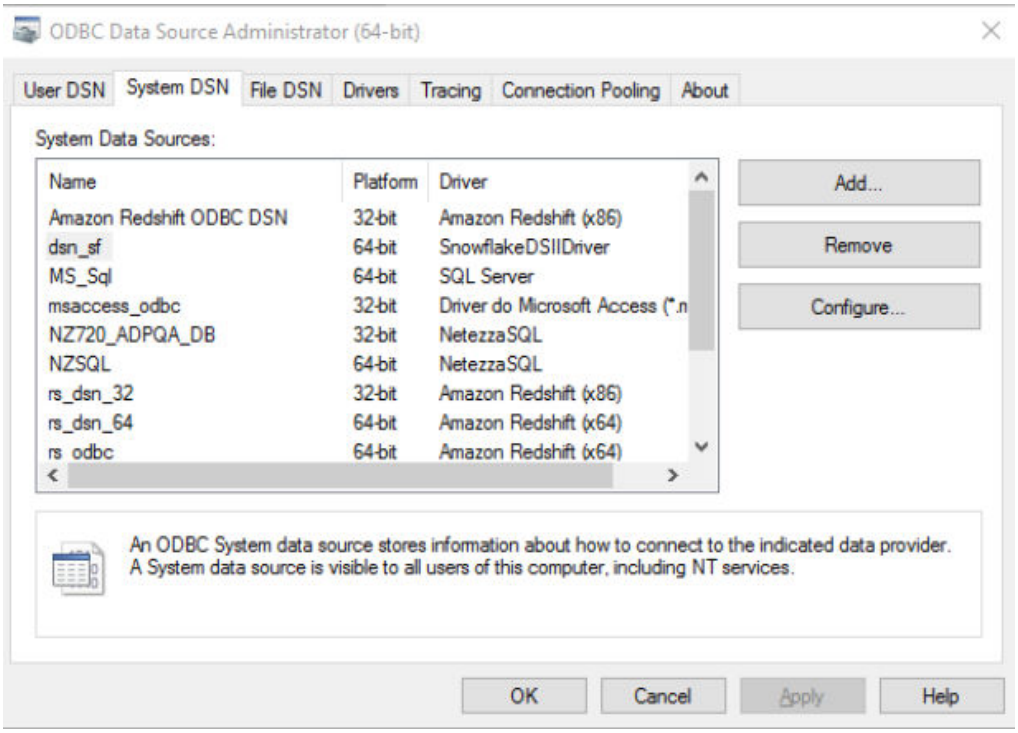
Configuring the Google BigQuery ODBC Driver on Windows

To establish an ODBC connection to connect to Google BigQuery on Windows, install the 64-bit Google BigQuery ODBC driver on the machine where the PowerCenter Integration Server runs. Install the 32-bit ODBC driver on the machine where you install the PowerCenter Client.
After you install the driver, configure the Data Source Name (DSN) in the ODBC Data Source Administrator.

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

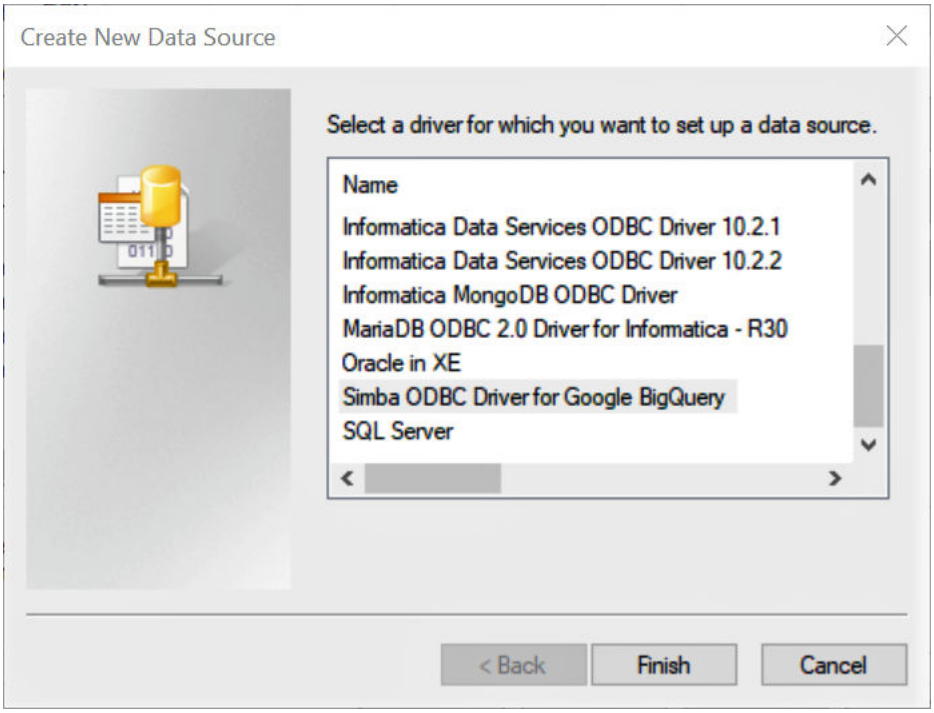
   The **ODBC Data Source Administrator** dialog box appears.
4. Click the **System DSN** tab.

   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**.

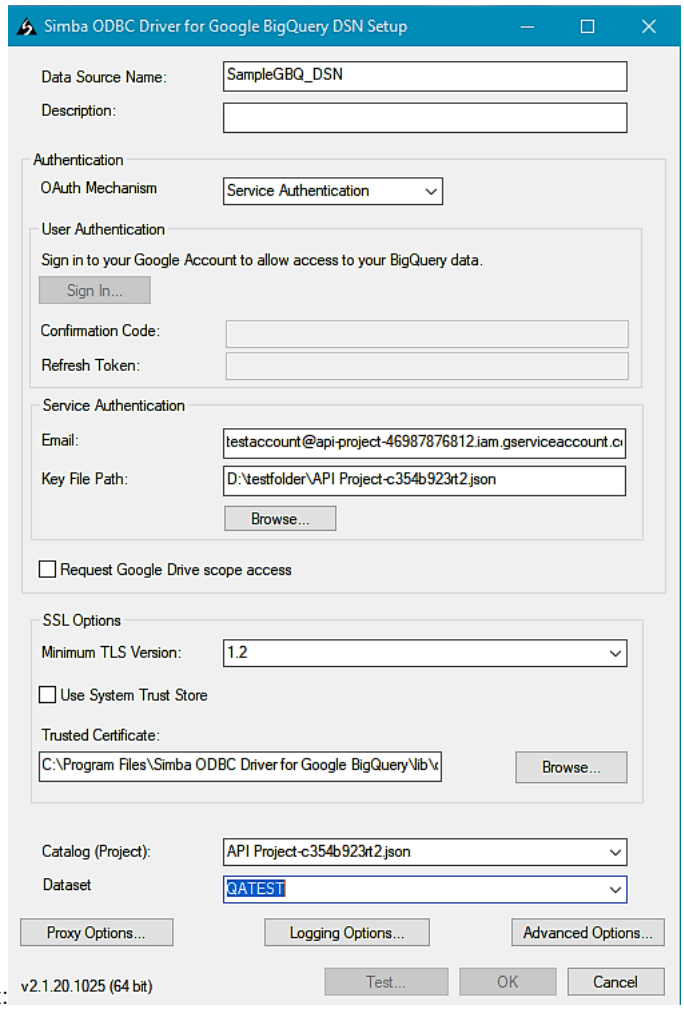
The **Create New Data Source** dialog box appears. The following image shows the Create New Data Source dialog box where you can select the Google BigQuery data source:



6. Select **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. Configure the connection properties.

   The following table describes the connection properties that you must configure:

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.

Property	Description
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.

    Note: Simba ODBC Driver for Google BigQuery does not support Legacy SQL.
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

Configuring the Google BigQuery ODBC Driver on Linux

To establish an ODBC connection to connect to Google BigQuery on Linux, you must download the Google BigQuery ODBC (64-bit) driver on the Linux machine that runs the PowerCenter Integration Service and configure the ODBC connection.

1. Install the driver on the machine where the PowerCenter Integration Service runs.
2. 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=<Driver 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
```

The following table describes the 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. **Note:** Simba ODBC Driver for Google BigQuery does not support Legacy SQL.
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 **Email**
	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.

Property	Description
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.

3. Set the ODBCINI environment variable to point to the directory that contains the `odbc.ini` file. For example:

```
setenv ODBCINI <$ODBC_HOME>/odbc.ini
```

4. Restart the PowerCenter Integration Service.

Pushdown Optimization Configuration Tasks

You can implement pushdown optimization to improve the performance of a PowerCenter session that uses a Google BigQuery mapping. To configure a pushdown optimization in a Google BigQuery session, perform the following tasks:

1. Import the Google BigQuery ODBC source table or tables.
2. Create a Google BigQuery ODBC connection.
3. Create a Google BigQuery mapping.
4. Create a Google BigQuery session.
5. Configure a session for full pushdown optimization.

Import a Google BigQuery Source Definition

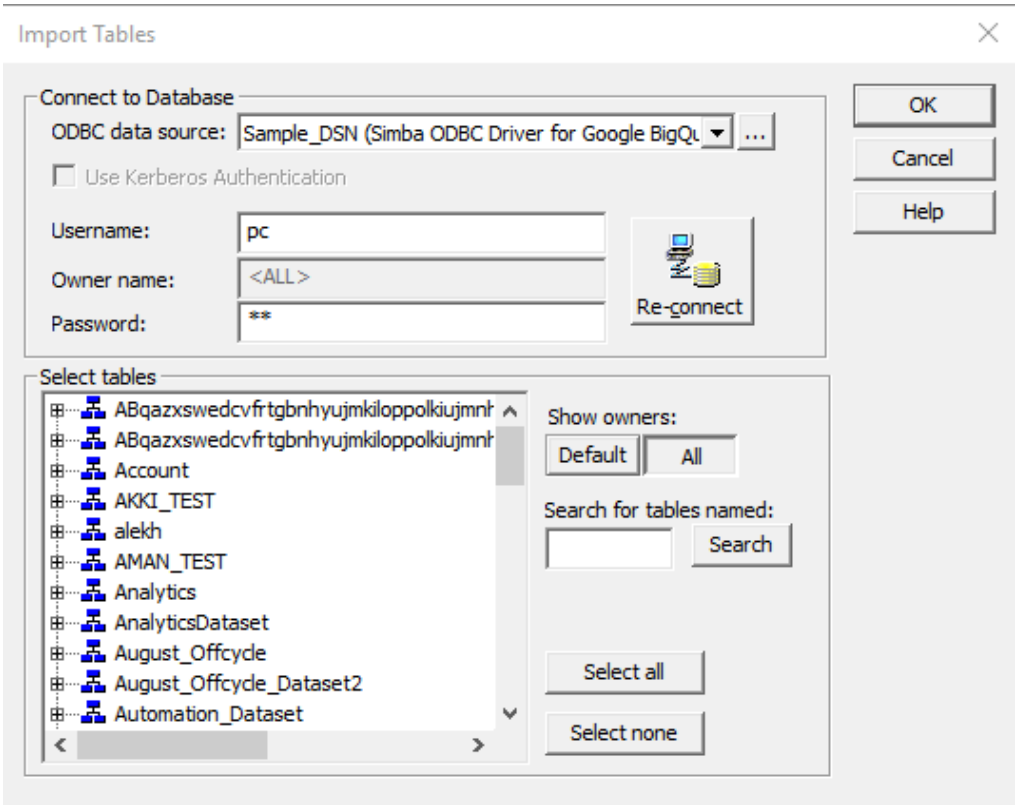
To create a Google BigQuery source definition, use the Source Analyzer to import source metadata with the Google BigQuery ODBC data source.

1. In the Source Analyzer, click **Sources** > **Import from Database.**
2. Select the Google BigQuery data source used to connect to the source database.

   If you need to create or modify a Google BigQuery data source, click the **Browse** button to open the ODBC Administrator. Create the Google BigQuery data source and click **OK.** Select the new Google BigQuery data source.
3. Enter the user name and password to connect to Google BigQuery.

   Note: Specify a dummy username and password.
4. Click **Connect.**

The following image shows the **Import Tables** dialog box where you can import a Google BigQuery source definition:



5. Select a dataset and scroll down through the list of tables to find the source you want to import. Select the Google BigQuery table or tables you want to import.

   You can hold down the **Shift** key to select a block of sources within one folder or hold down the **Ctrl** key to make non-consecutive selections within a folder. You can also select all tables within a dataset by selecting the dataset and clicking **Select All**. Use the **Select None** button to clear all highlighted selections.

6. Click **OK.**

The source definition appears in the Source Analyzer. In the Navigator, the source definition appears in the Sources node of the active repository folder under the source database name.

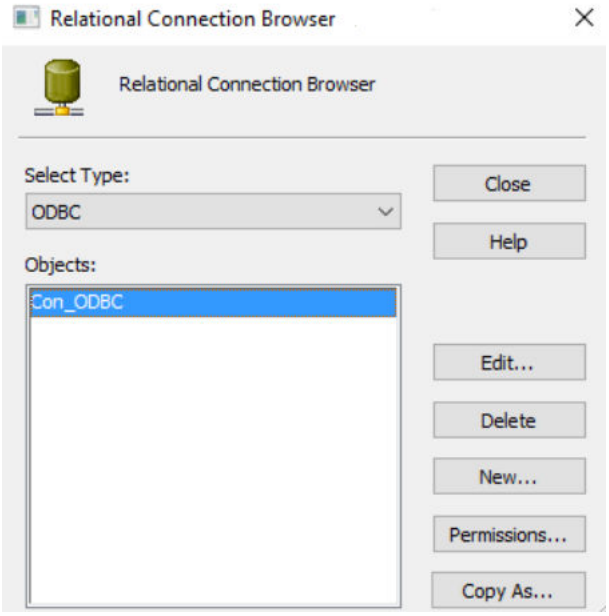
Create a Google BigQuery ODBC Connection

Use the configured Google BigQuery ODBC driver to create an ODBC connection to connect to Google BigQuery from PowerCenter.

1. In the Workflow Manager, click **Connections**.
2. Select **Relational** from the list.

   The **Relational Connection Browser** box appears.

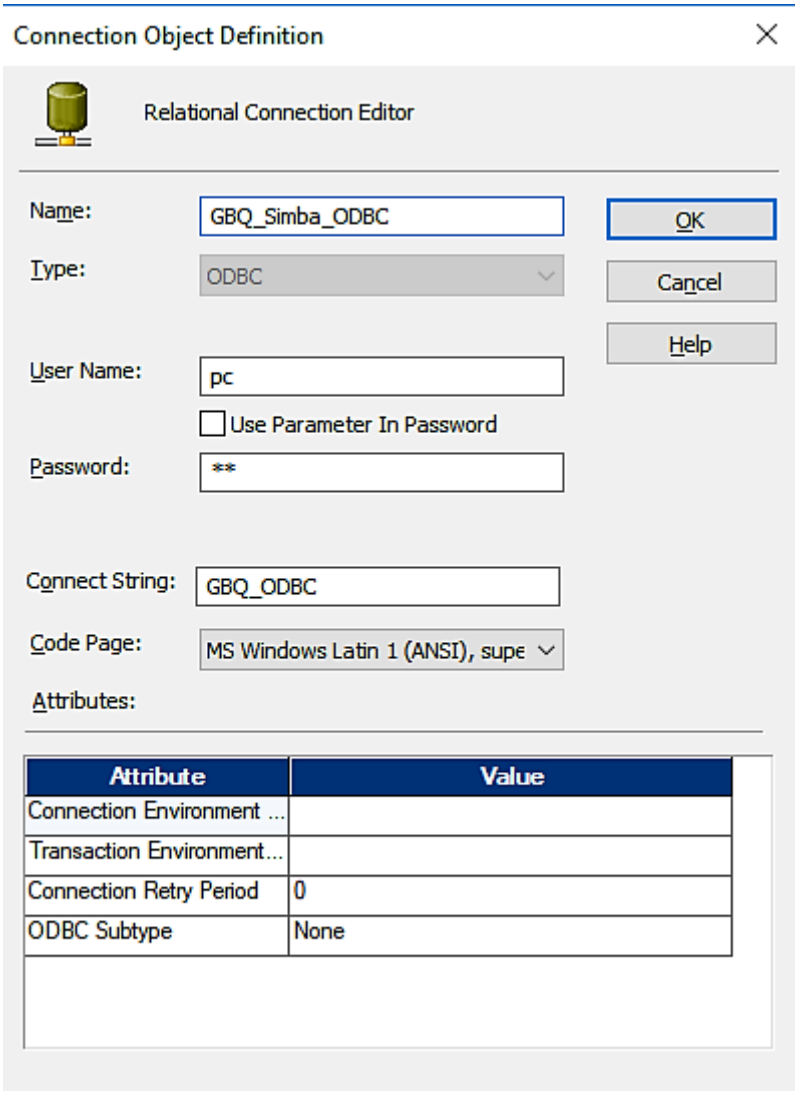
The following image shows the **Relational Connection Browser** box:



3. Select **Type** as **ODBC**.
4. Click **New**.

   The **Connection Object Definition** box appears.

The following image shows the **Connection Object Definition** dialog box where you configure the connection properties:



5. Configure the relational connection properties.

   The following table lists the relational connection properties that you must configure:

Relational Connection Property	Description
Name	Name of the connection.
Type	The connection type. The value is set by default. You cannot edit this value.
User Name	Username to connect to the Simba Google BigQuery ODBC data source. Specify a dummy username.

Relational Connection Property	Description
Password	Password to connect to the Simba Google BigQuery ODBC data source. Specify a dummy password.
Connect String	Name of the ODBC data source that you want to use to connect to Google BigQuery.
Code Page	The code page that the PowerCenter Integration Service uses to read or write data.
Attributes	The **ODBC Subtype** attribute value. Select **None**.

6. Click **OK**.

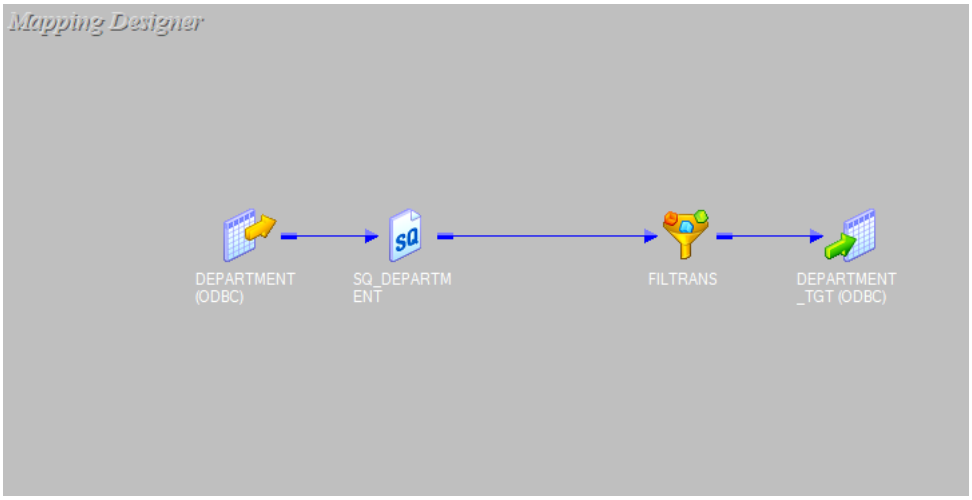
Create a Google BigQuery Mapping

Create a mapping to represent how the PowerCenter Integration Service extracts, transforms, and loads data to Google BigQuery.

For example, create a mapping to read the product details from the Google BigQuery source, apply a filter transformation to filter data based on the department, and write the records to a Google BigQuery target for data analysis.

1. Import a Google BigQuery source and target definition into the PowerCenter repository.
2. Create a Google BigQuery mapping with a Google BigQuery source and target.

   The following image shows a Google BigQuery mapping:



   The Google BigQuery mapping contains the following objects:

   - A Google BigQuery source definition that contains the product department details, such as the Department ID, Department Name, Product ID, Product Description, Quantity, and Cost.
   - An Filter transformation to filter product details based on the Department ID.
   - A Google BigQuery target definition.

Create a Google BigQuery Session

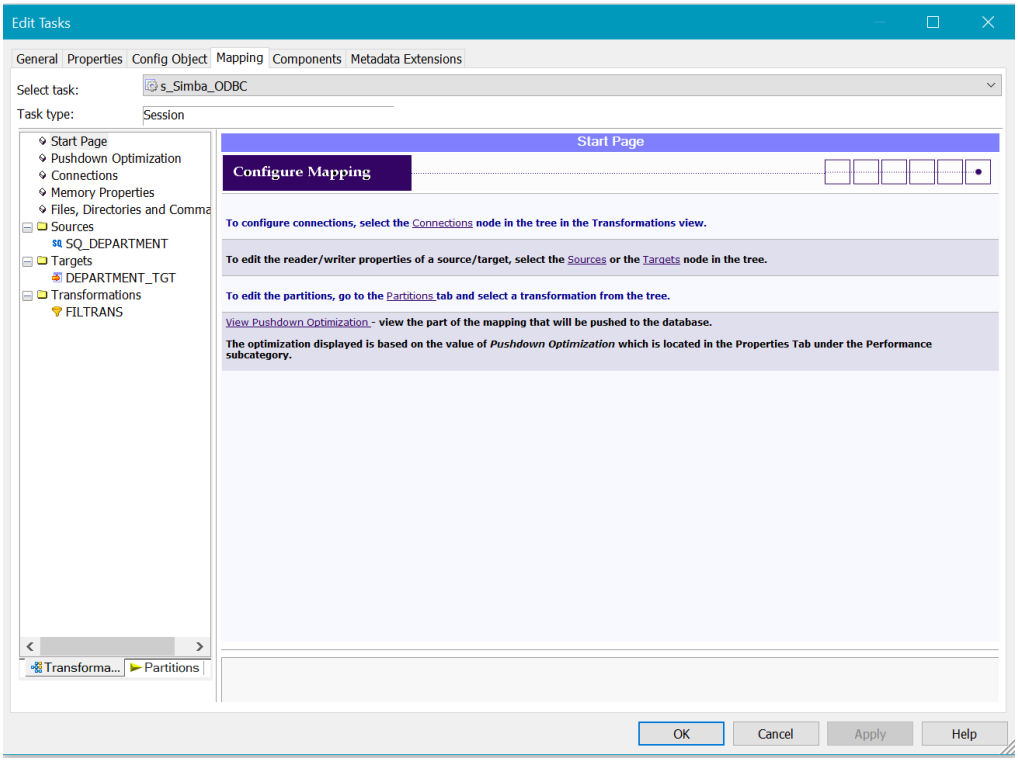
Use the **Workflow Manager** to create a Google BigQuery session and use the configured mapping in the session.

1. Click **Tools** > **Task Developer**.
2. Click **Tasks** > **Create**.
3. Select **Session** as the task type to create.
4. Enter the session name and click **Create**.

   The **Mappings** dialog box appears.
5. Select the Google BigQuery mapping for which you want to configure pushdown optimization, and click **OK**.

   The **Workflow Manager** creates a reusable Session task in the Task Developer workspace.
6. Click **Done** in the **Create Task** dialog box.
7. In the workspace, double-click the session you created to open the session properties.
8. On the **Mapping** tab, select **Sources** or **Targets** in the Transformations pane on the left.

   The following image shows the **Mapping** tab for the session:



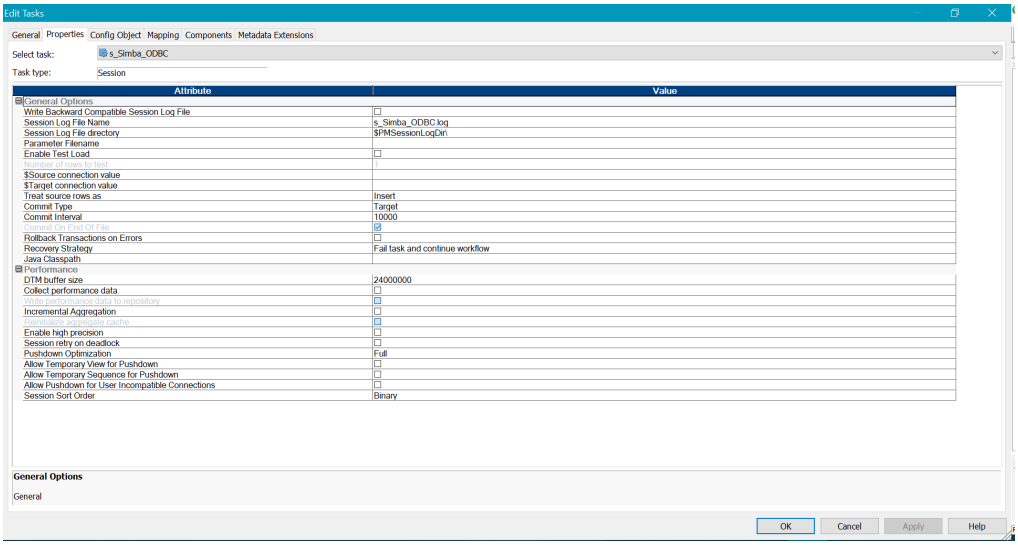
9. In the Connections section, select the configured Google BigQuery ODBC connection.
10. Click **OK**.

Configuring a Google BigQuery Session for Pushdown Optimization

To optimize the performance of the Google BigQuery session, configure full pushdown optimization in the session properties.

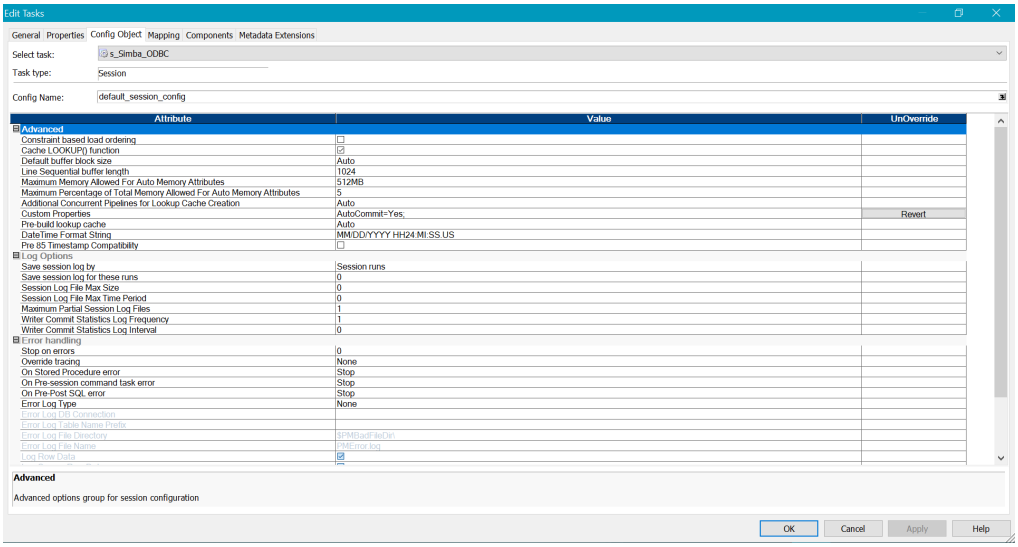
1. Open the session properties for the session that you created.
2. On the **Properties** tab, access the **Performance** section and select the value of the **Pushdown Optimization** as **Full**.

   The following figure shows the selected full pushdown optimization option:



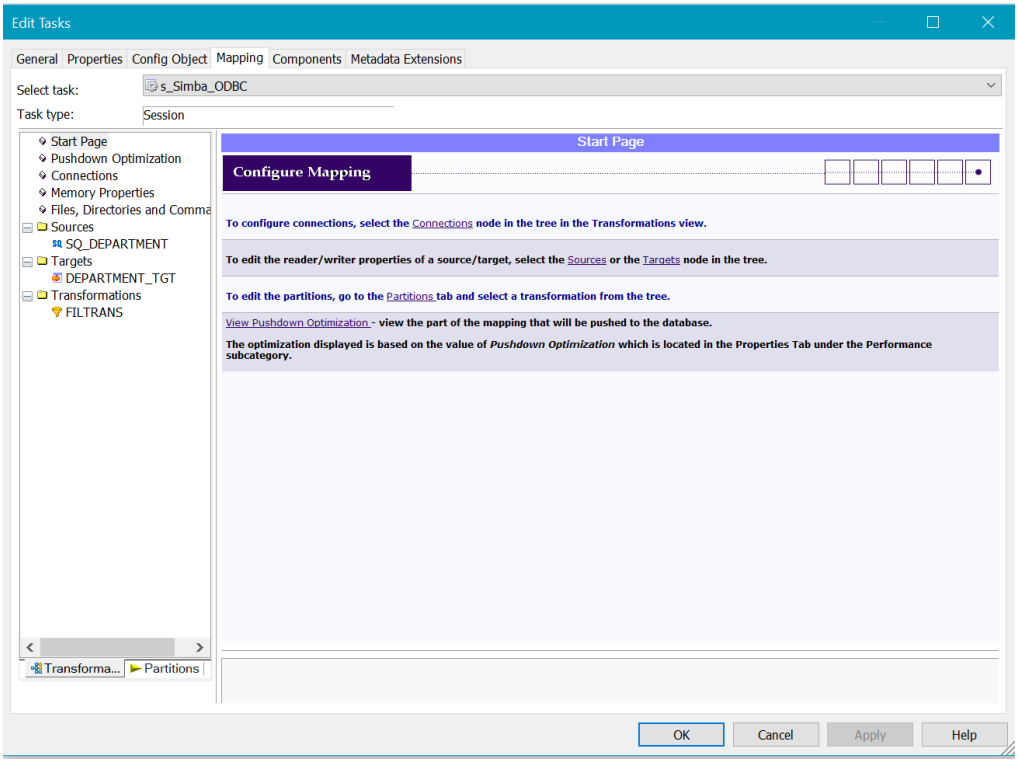
3. On the **Config Object** tab, set **AutoCommit=Yes** in the **Custom Properties**.

   The following figure shows the **Config Object** tab:



4. Click the **Mapping** tab in the session properties.

The following figure shows the **Mapping** tab:



5. Click **View Pushdown Optimization** to view the full pushdown optimization logic.

   The **Pushdown Optimizer Viewer** window appears and displays the pushdown groups and the SQL that is generated to perform the transformation logic. It displays messages related to each pushdown group. It also displays numbered flags to indicate the transformations in each pushdown group.

APPENDIX A

Google BigQuery Data Type Reference

This appendix includes the following topics:

Data Type Reference Overview

PowerCenter uses the following data types in Google BigQuery mappings:

- Google BigQuery native data types. Google BigQuery data types appear in Google BigQuery definitions in a mapping.
- Transformation data types. Set of data types that appear in the transformations. They are internal data types based on ANSI SQL-92 generic data types, which the PowerCenter Integration Service uses to move data across platforms. They appear in all transformations in a mapping.

When the PowerCenter Integration Service reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When the PowerCenter Integration Service writes to a target, it converts the transformation data types to the comparable native data types.

Google BigQuery and Transformation Data Types

The following table lists the Google BigQuery data types that PowerCenter supports and the corresponding transformation data types:

Google BigQuery Data Type	Transformation Data Type	Range and Description for the Transformation Data Type
BOOLEAN	Nstring	1 to 104,857,600 characters
BYTE	Byte	1 to 104,857,600 bytes

Google BigQuery Data Type	Transformation Data Type	Range and Description for the Transformation Data Type
DATE	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
DATETIME	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
FLOAT	Double	Precision 15
INTEGER	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0
RECORD	Nstring	1 to 104,857,600 characters
NUMERIC	Decimal	Default precision 38, scale 9 PowerExchange for Google BigQuery supports precision upto 28 and scale upto 9. Decrease the precision to 28 to avoid any data type mismatch. **Note:** You must enable high-precision in the session properties to use 28-bit precision. If you do not enable high precision, data is truncated.
STRING	Nstring	1 to 104,857,600 characters
TIME	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)
TIMESTAMP	Date/Time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. (precision to the nanosecond)

INDEX

W