
Hint for Naming Resources

When naming resources, use the format `bi-2023hs-[RESOURCE NAME]-[SHORT NAME]` where `**[SHORT NAME]**` is your account name and should be filled by you. Adjust the `**[RESOURCE NAME]**` accordingly based on the specific resource you're creating.

Exercise: Connect BigQuery to Cloud SQL on Google Cloud Platform (GCP)

In this exercise, you will establish a connection between BigQuery and Cloud SQL. You'll set up a connection to Cloud SQL, grant necessary permissions, and then test your connection.

Prerequisites

- A Google Cloud Platform account.
- A Cloud SQL instance with public IP connectivity without an authorized address. If you haven't set this up, ask your administrator or refer to Cloud SQL documentation.
- The BigQuery Connection Admin (`roles/bigquery.connectionAdmin`) IAM role granted on your project. If you don't have this permission, request it from your administrator.

Step 1: Navigate to BigQuery

1. Open the [Google Cloud Console](#).
2. Make sure you've selected the correct project from the project dropdown.
3. In the left sidebar, click on BigQuery.

Step 2: Create a Cloud SQL Connection

1. In BigQuery's Explorer pane, click on **Add data**.
2. Choose **Connections to external data sources** from the options.
3. Fill in the required details:
 - **Connection type:** Choose the source type (e.g., MySQL or Postgres).
 - **Connection ID:** Use the format `bi-2023hs-connection-[SHORT NAME]`.
 - **Data location:** Pick a BigQuery location that's in the same region as your Cloud SQL instance.

-
- **Cloud SQL instance ID:** Go to your Cloud SQL instance overview page. Here, you'll find the full name of the Cloud SQL instance in the format `project-id:location-id:instance-id`. Enter this ID.
 - **Database name:** Enter the name of the database you wish to query.
 - **Database username:** Input the username for the database.
 - **Database password:** Type in your database password.
 - To see the password, click on **Show password**.

4. Click **Create connection**.
5. Navigate to the connection by clicking **Go to connection**.
6. From the **Connection info pane**, take note of the service account ID.

Step 3: Grant Access to the Service Account

1. In the Google Cloud Console, go to the IAM & Admin section.
2. Click on **Grant Access**.
3. In the **New principals** field, input the service account name "BigQuery Connection Service Agent". If you can't find the name, refer to the **Connection info pane** from Step 2 where you noted the service account ID, and paste that ID here.
4. In the role selection field, choose **Cloud SQL**, then select **Cloud SQL Client**.
5. Confirm by clicking **Save**.

This modification provides clearer instructions for students on where to find the service account details.

Step 4: Test Your Connection

1. Go back to BigQuery.
2. Try running a query on your Cloud SQL database using the connection you just set up. You can test it with this query:

SELECT

*

FROM

```
EXTERNAL_QUERY("bi-2023hs-pect.europe-west6.postgres-bq",  
  "SELECT ROUND(AVG(salesorderdetail.unitprice), 2) as SalesAverage from  
  ↪ sales.salesorderdetail;");
```

What is the average unit price of all sales orders?

Understood! I'll make sure to incorporate that in the SQL template, making it more PostgreSQL-specific.

Step 5: Find the Highest CustomerID

Task: Identify the highest CustomerID amongst customers with the last name 'Yang'.

Resources: - **ER Diagram:** Download it from Moodle - **Query Structure:**

SELECT

[Columns you want to retrieve, e.g., t1.column_name, t2.column_name]

FROM

[Schema].[Main Table, e.g., schema_name.table1 as t1]

INNER JOIN

[Schema].[Another Table, e.g., schema_name.table2 as t2] ON

↪ t1.shared_key = t2.shared_key

WHERE

[Condition, e.g., t2.LastName = 'Yang']

ORDER BY

[Ordering criteria, e.g., t1.CustomerID DESC]

LIMIT 1;

Hints: - Remember the relationships between tables. - In PostgreSQL, tables are often accessed using the `schema_name.table_name` format. So, ensure to prepend the schema name before the table name. - The information you need is spread across two tables: one containing customer details and the other with personal information like names. You'll need to join these tables to access both the CustomerID and the last name. - **Join Hint:** Use the `INNER JOIN` clause and the `ON` keyword to specify the columns that link the two tables together. These are often shared ID or key columns. In the SQL template above, replace `schema_name.table1` and `schema_name.table2` with actual schema and table names, and `shared_key` with the actual column that links the two tables. - Use the `WHERE` clause to filter by last name. - Use the `ORDER BY` clause in conjunction with `LIMIT` to get the highest CustomerID.

Once you've constructed your query, run it in BigQuery to verify the result.

This template now emphasizes the PostgreSQL-specific schema and table naming convention, which should help guide the students more effectively.

Once you've constructed your query, run it in BigQuery to verify the result.