BigLake and Multi-Source Data Management Assignment

Introduction

This assignment will help you practice advanced data management using Google Cloud services, including Bigtable and BigQuery, in a BigLake environment. You will learn how to use Python to create external tables, query data across multiple data sources, and manage data access.

All tasks should be completed using the Python SDK for Google Cloud services.

Dataset Information

For this assignment, you will connect to a simulated Bigtable instance and a Cloud Storage CSV file. The data will represent user activities across different services.

1. Bigtable Instance: ”student_instance”
2. Bigtable Table: ”student_data”
3. Cloud Storage Bucket: ”your-bucket-name”
4. Cloud Storage File: ”activity_log.csv”

Tasks

Task 1: Bigtable Table Creation and Data Insertion

Objective: Create a Bigtable instance and table using Python and insert data into it.

Instructions:

1. Set Up Environment: Install the necessary Google Cloud libraries (pip install google-cloud-bigtable)

2. Create a Bigtable Instance and Table:

- Create an instance named ”student_instance”.
- In this instance, create a table named ”student_data” with two column families:”user_info” (Row Key & user_info) and ”activity_log” (activity & activity_log).

3. Insert Data into Bigtable:

Write a Python script to insert the following sample data into ”student_data”:

Row Key	**user_info**	**activity**	**activity_log**
User1	**Alice**	**login**	**2024-01-01T12:00:00Z**
User2	**Bob**	**purchase**	**2024-01-02T08:30:00Z**
User3	**Cindy**	**login**	**2024-01-03T15:45:00Z**
User4	**David**	**purchase**	**2024-01-05T17:36:00Z**

Hints:

- https://cloud.google.com/bigtable/docs/reference/libraries

Task 2: External Table Creation in BigQuery (Bigtable Source)

Objective: Create an external table in BigQuery that connects to your Bigtable data.

- **Instructions:**

1. Set Up Environment: Install the BigQuery Python library (pip install google-cloud-bigquery)

2. Create BigQuery External Table:

- Use Python to create an external table in BigQuery named ”bigtable_external”.
- This table should map to the ”student_data” table in your Bigtable instance.
- Map the following columns:
  - ”user_info” as ”user_name”
  - ”activity” as ”activity_type”
  - ”activity_log” as”activity_time”

Hints:

- https://cloud.google.com/bigquery/docs/create-bigtable-external-table (10% Bonus Points)

Task 3: External Table Creation in BigQuery (Cloud Storage Source)

Objective: Create an external table in BigQuery that connects to a Cloud Storage CSV file.

Instructions:

1. Upload CSV File:

- Upload a sample CSV file named ”activity_log.csv” to your Cloud Storage bucket.

2. Create BigQuery External Table:

- In BigQuery, create an external table named ”cloud_storage_external” pointing to the CSV file.
- Ensure the table structure includes ”user_name”, ”activity_type”, and ”activity_time”.

Hints:

- https://cloud.google.com/bigquery/docs/external-data-cloud-storage

Task 4: Query Multi-Source Data in BigQuery Using Python

Objective: Query data across the Bigtable and Cloud Storage external tables in BigQuery.

Instructions:

1. Write a Query to Join Data:

- Write a Python script to query data from ”bigtable_external” and ”cloud_storage_external” tables.

- Use a ”JOIN” (or LEFT JOIN or INNER JOIN) statement to combine user activity logs from Bigtable with activity timestamps from the Cloud Storage CSV file.

2. Query Criteria:

- Select entries where the activity type is ”login”.
- Order results by ”activity_time” in descending order.

Hints:

- Use ”client.query()” to execute the query and ”to_dataframe()” to convert results to a DataFrame.
- Clean NULL value or use UNION for task 5

Task 5: Data Analysis with Window Functions

Objective: Perform a data analysis on the joined/unioned dataset using window functions.

Instructions:

1. Write and Execute Query:

- Write a query to rank each user by the number of ”login” activities.
- Display the top 2 users with the highest login count.

Hints:

- Use ”RANK() OVER()” to rank users based on login activity count.

Task 6: Data Access Management

Objective: Manage BigLake data access by setting permissions on the BigQuery dataset.

Instructions:

1. Grant Access to Another User:

- Use Python to grant the ”bigquery.dataViewer” role on your BigQuery dataset to the TA (113356042@g.nccu.edu.tw).

Hints:

- https://cloud.google.com/bigquery/docs/control-access-to-resources-iam#python
- AccessEntry

Submission Instructions

- Please complete and share your assignment with the TA by November 13, 2024, at 11:59 PM.
- Please submit your Python file that includes solutions for all six tasks described in the assignment. Make sure your code is well-structured and properly commented to explain the functionality implemented in each task.

- Please name your Python file using your student ID (e.g., studentID.py). Submit your Python file on MOODLE by November 13, 2024, at 11:59 PM.