Data Pipeline Mini Project

Event Ticket System Case Study

Estimated Time: 3-5 hours

Ticket systems are designed to support online ticket sales and purchase, for events like concerts and sports matches. They allow individuals to buy tickets directly through the platform. Many popular ticket platforms also manage ticket sales for third party resellers, by ingesting sales data from these vendors.

In this project, you are provided with a ticket system database with ticket sales event table (download data here). This table tracks all ticket sales for various events, including when a third-party reseller submits their records of ticket sales for a new day. The records are submitted as a CSV file. **Your task is to:**

- **Write the table definition DDL and use it to create the sales table.**
- **Read the CSV file and load the new sales records into the ticket sales table.**

Ticket Sales Table Schema:

Column	Type
ticket_id	INT
trans_date	INT
event_id	INT
event_name	VARCHAR(50)
event_date	DATE
event_type	VARCHAR(10)
event_city	VARCHAR(20)
event_addr	VARCHAR(100)
customer_id	INT
price	DECIMAL
num_tickets	INT

Learning Objectives:

With this-mini project, you'll practice Python and SQL skills by creating a basic data pipeline. You'll learn how to use the Python database connector to perform data loading and query databases programmatically.

Step-by-Step Instructions:

Step 1. Install MySQL Python connector

mysql-connector-python is a MySQL database adapter in Python. It provides convenient APIs to load and query the tables. It also has a nice tool to load CSV files into the tables. In this step, we will need to install this Python module.

```
pip3 install mysql-connector-python
```

Step 2. Load third-party ticket sales data into MySQL database

2.1 Setup database connection

In order to make a query against the database table, we need to first connect to it. A connection can be established only when the user provides the proper target host, port, and user credentials.

```
def get_db_connection():
    connection = None
    try:
        connection = mysql.connector.connect(user='<username>',
                                             password='<password>',
                                             host='<hostname>',
                                             port='3306',
                                             database='<database name>')
    except Exception as error:
        print("Error while connecting to database for job tracker", error)

    return connection
```

2.2 Load CSV to table

You'll find the third party vendor data in the CSV file provided to you. The CSV follows the schema of the table. You will need to use the Python connector to insert each record of the CSV file into the "sales" table.

```
def load_third_party(connection, file_path_csv):
    cursor = connection.cursor()

    # [Iterate through the CSV file and execute insert statement]

    connection.commit()
    cursor.close()
    return
```

Step 3. Display statistical information

After the data is loaded into the table, you can use this data to provide recommendations to the user. For instance, recommending popular events by finding the most top-selling tickets for the past month.

3.1 Query the table and get the selected records

```
def query_popular_tickets(connection):
    # Get the most popular ticket in the past month
    sql_statement = "<your sql statement>"
    cursor = connection.cursor()
    cursor.execute(sql_statement)
    records = cursor.fetchall()
    cursor.close()
    return records
```

3.2 Display the result

The records you just retrieved are formatted as a list of tuples. You need to convert the format to display the on-screen results in a more user-friendly format. Please use this as an example.

```
Here are the most popular tickets in the past month:
- The North American International Auto Show
- Carlisle Ford Nationals
- Washington Spirits vs Sky Blue FC
```

Instruction for Submission:

- Push the Python code to your GitHub repo.
- Add a readme file to include steps to run your code and verify the result. Your mentor should be able to run it by following your instructions.
- Two examples of readme file:
    - Example 1
    - Example 2
- Attach the command line execution log for the successful job run. You can capture it in a text file.