Description:

Data Extraction -

1. The script extracts event details such as title, venue, date, time, artists, programs, and image links from the Lucerne Festival website.
2. It iterates through event links and retrieves information for each event.
3. The data is stored in a Pandas DataFrame and exported to a CSV file named "lucerne\_festival\_data.csv".

Inserting into postgres –

1. This repository contains Python code to interact with a PostgreSQL database and perform the following tasks:
2. Create tables for storing event-related data.
3. Insert unique values into separate tables for venues, artists, programs, and titles.
4. Update a DataFrame with corresponding IDs from the database tables.
5. Optionally, insert data from the updated DataFrame into the events table.

Prerequisites

1. Python 3 installed on your system.
2. PostgreSQL database server running locally or remotely, accessible via Python's psycopg2 library.
3. Basic knowledge of SQL and Python programming.

Setup Instructions

1. Install the required Python packages using pip:

* pip install pandas psycopg2-binary

1. Ensure that your PostgreSQL server is running and accessible.

PostgreSQL Installation

1. Download PostgreSQL: Visit the official PostgreSQL website and download the appropriate installer for your operating system.
2. Install PostgreSQL: Run the downloaded installer and follow the installation wizard's instructions. Make sure to remember the password you set for the default postgres user during installation.

pgAdmin Installation

1. Download pgAdmin: Go to the pgAdmin download page and download the installer for your operating system.
2. Install pgAdmin: Run the downloaded installer and follow the installation wizard's instructions.

Connect to PostgreSQL Database

1. Launch pgAdmin: Open pgAdmin from your computer's applications menu or desktop shortcut.
2. Add Server: In pgAdmin, click on the "Add New Server" button or right-click on "Servers" in the left sidebar and select "Create" > "Server..."
3. Enter Connection Details: Fill in the following details:
4. Name: A friendly name for your server.
5. Host name/address: localhost if PostgreSQL is running on the same machine; otherwise, enter the IP address or hostname of your PostgreSQL server.
6. Port: 5432 is the default port for PostgreSQL.
7. Username: postgres (default username).
8. Password: Enter the password you set during PostgreSQL installation.
9. Save Connection: Click "Save" to save the connection settings.
10. Connect to Server: Double-click on the server you just added to connect to the PostgreSQL database.

The connection details in my code -

(user="postgres",

password="postgres",

host="localhost",

port="5432",

database="docker")

* **Additionally, I have also submitted the docker files namely Dockerfile.df and docker-compose.yml.**

**But unfortunately, I couldn’t dockerize the solution.**