**Task Title:** Develop a Simple ETL Pipeline using Python and PostgreSQL

**Objective:**The goal of this task is to build a basic ETL (Extract, Transform, Load) pipeline that extracts data from a file, transforms it into a desired format, and loads it into a PostgreSQL database hosted locally or ideally on AWS RDS. This will demonstrate your ability to work with Python, SQL, and optionally AWS infrastructure.

**Deliverables:**

**A Python script that performs the ETL process (Extract, Transform, Load).**

Link:

**Medium Blog Link**: https://medium.com/@waqasdost/from-data-to-maps-etl-pipeline-with-python-postgresql-and-aws-rds-d68577212da5

**The SQL script used to create the PostgreSQL table.**

Link:

**A short write-up (1-2 paragraphs) explaining the decisions you made during the task (e.g., extracting state code from address, any challenges faced).**

In this ETL pipeline task, my key focus was on extracting, transforming, and loading data related to addresses into PostgreSQL and AWS RDS databases, followed by visualization using Folium. During the transformation phase, I encountered the challenge of parsing addresses that were not uniformly structured. To address this, I split the addresses into components such as house number, city, state code, and zip code using string manipulation. I also ensured that rows with missing address, latitude, and longitude values were removed, to avoid errors during data insertion into PostgreSQL.

A notable decision was to leverage PostgreSQL for local storage and then move the cleaned data to an AWS RDS instance, simulating a cloud-based production environment. This required setting up secure connection parameters and ensuring that the AWS RDS database mirrored the local PostgreSQL structure for seamless migration of data. For visualization, I used Folium to generate an interactive map by fetching the latitude and longitude data from the database, with markers placed based on the extracted addresses.

**Clear instructions on how to run the script and connect to the PostgreSQL instance.**

* Set up the Python Environment
* Install the required libraries
* Installed PostgreSQL Database
* Create PostgreSQL Database and Table named Addresses.
* Update the connection details (host, user, password) in the script to match your PostgreSQL settings
* Update the connection details (host, user, password) in the script to match your AWS RDS PostgreSQL settings (End point, Master db, db, password)
* Performed ETL using Pandas
* Execute the SQL query to count addresses by state and save it as a CSV file using psycopg2
* Visualize the addresses on a map using Folium

**Commit the complete code on a GitHub repository and share its link.**

Link:

**Complete Code must be functional on your Laptop.**

Yes, all the code and is functional on my laptop with Jupyter notebook, locally installed PostgreSQL and AWS RDS EC2 PostgreSQL free tier.