ETL Challenge Report

(Francis Odo)

Background

Most business entities, government institutions and agencies maintain some type of database in their day-to-day operations. Depending on business type, industry category, as well as other need factors, the nature of data collection varies widely. There is the necessity to proactively keep up with the accumulation and maintenance brought about by business operations and other associated factors. One of the ways of achieving this task is the Extract Transformation and Load (ETL) process. This project takes us through the process from beginning to the end.

Objective

The objective is to automate the process of Extract Transformation and Load (ETL) in fulfilling the need of an ideal business entity wanting to maintain an information database. The process takes in a set of files in raw data format, then:

a) Extract - Take in the information in the best available format(file) and condition
b) Transformation - Perform various types of cleaning, rearrangement and updates
c) Load - Load the data into a functional database platform in a clean and easy to understand format.

Development Environment

Code Development – Jupyter Notebook

Programming Language – Python

Libraries – Pandas

Database – PostgresDB

ORM - Sql Alchemy

Code File - Module_8_Challenge.ipynb

Config File – config.ny

Code Plan

(1) Import all the necessary dependencies and set up file paths to raw datafile location
(2) Define engine and database connection string
(3) Create a main function in Python using “def etl_pipe_func(file 1, file 2, file 3)”.
The three raw files are:
(a) wikipedia.movies.json
(b) movies_metadata.csv
(c) ratings.csv

(4) Open and read the files with the appropriate and applicable Pandas method. Verify that data is being read correctly within the function created for automation purposes.
(5) Perform transformation which includes cleaning, avoiding duplicates, merging columns, regular expression and other necessary steps.
(6) Create and configure the "movies_data" database in PostgreSQL.
(7) Remove existing data from the table
(8) Load the clean and transformed data into PostgreSQL database. Apply **try-except** to catch errors in the process.

Approach and Assumptions

a) Conditions of raw data can vary in so many ways and may require different steps/type of cleaning. The approach is to demonstrate the fundamental principle and methods of cleaning in limited fashion with emphasis on automating the process.

b) One key assumption is that Opening and Reading files, Transforming/Cleaning and Loading to the Database steps are primary essential processes.

c) Data cleaning varies depending on how dirty the data may be. There is no limit to cleaning as much as it adds improvement.

Risks

1. Ratings file is large. Loading the file into the database is best done in chunks. Data integrity could be a challenge
2. The process is memory intensive, and tends to be slow sometimes. Loading process may halt and the process will have to be restarted.
3. Ensure that PostgresDB is up and running.

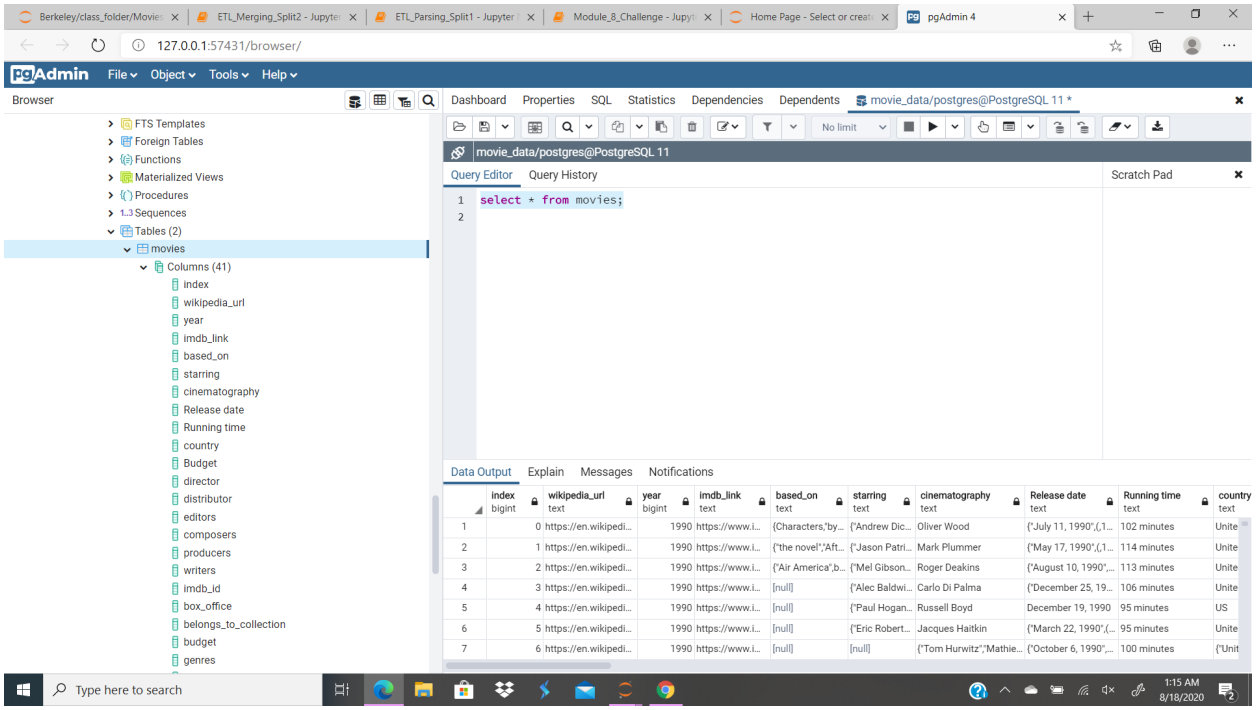
Conclusion

There are other ways of creating data pipelines that involve other methods and applications. ETL is fairly straight-forward and can handle average sized data successfully and conveniently. However, these methods are outside the scope of this project.

This application can be tailored or modified for any data environment to fulfill the basic needs. The main advantage here is to eliminate repetitive processes in maintaining data.

Verification of tables being created and populated in PostgresDB.

Movies Table:



Ratings Table: