EN.685.648.81.FA23 Data Engineering Project

## By: Jackie Huang, Twesha Mitra, Brandon Morrow, Dana Andrea

<add table of contents with links>

# Project Overview

In this project, our team built and end-to-end data pipeline for covid, economic, financial, and CO\_2 emissions data. The vision for our project is a dataset that a team of data scientists can use to see how COVID influenced the economy and climate. The sources we will use are the CDC, S&P 500, FRED, and the EPA.

### CDC Data

<https://data.cdc.gov/Case-Surveillance/COVID-19-Case-Surveillance-Public-Use-Data/vbim-akqf/explore>

Our COVID data is the CDC daily reporting dataset containing public use data. This data is at the national level, and contains columns such as date reported, sex, age group, ethnicity, hospitalized, icu, and death. We will join this data set with our others on the CDC report date column. This date column ranges from 2020-06-23 to 2023-10-02.

### S&P 500 Data

TBD

### FRED

<https://fred.stlouisfed.org/>

Short for Federal Reserve Economic Data, FRED is an online database consisting of hundreds of thousands of economic data time series from scores of national, international, public, and private sources. We will use this data to historic information for key economic indicators in the US including, unemployment, GDP, Real GDP, Federal Funds Rate, and interest rates. All of this data can be extracted as a .csv file. We will want to normalize all these indicators into one table or a few tables, joining them together by timeframe.

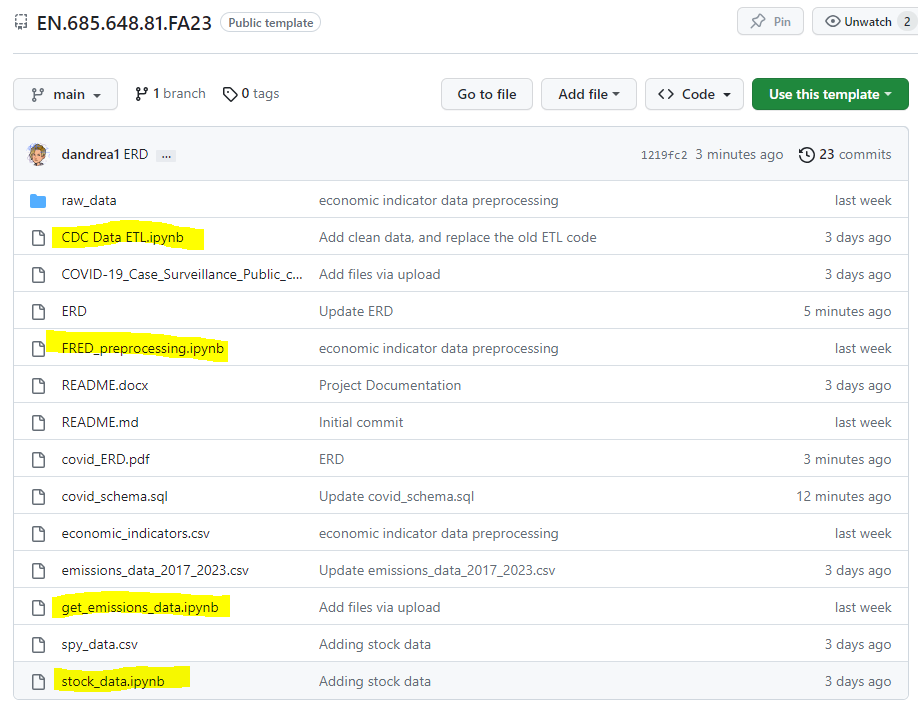
### EPA

<https://aqs.epa.gov/aqsweb/documents/data_api.html>

The EPA API provides emissions data by day for all locations in the United States. We will be combining this with COVID data on the date column to see if there was a decrease in emissions from COVID lockdowns. During our processing we take the daily national average of different air quality measures.

# ETL: Data Preprocessing

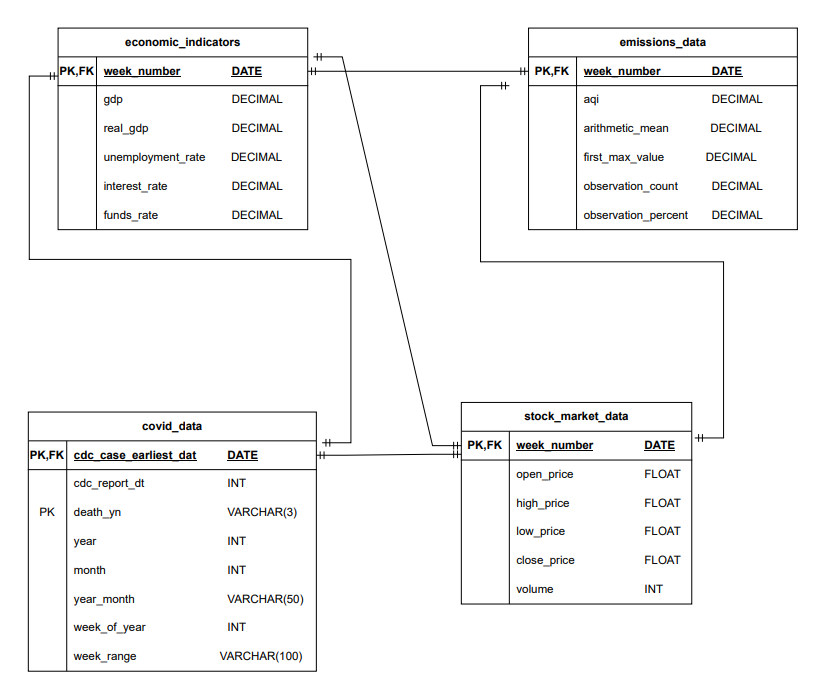
Each team member extracted a different data source and cleansed it to be ready to insert into the Postgres database. The jupyter notebooks are highlighted here:



In addition to modifying the correct data types, addressing null values, etc. All data was aggregated or disaggregated to the week level to be fit for our database.

# ERD Diagram & Data Dictionary

Our ERD looks like this:

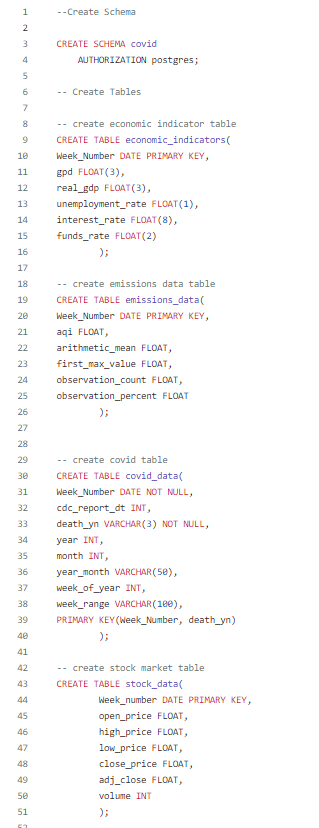


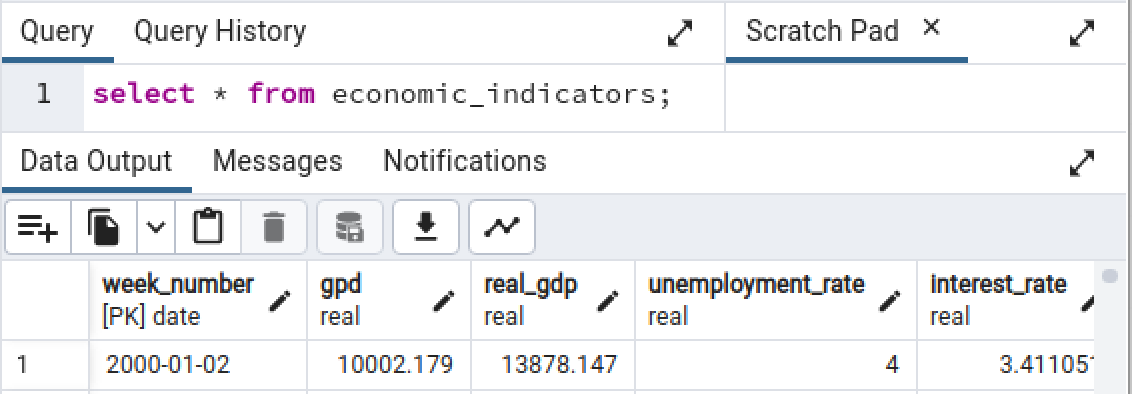
This is the data dictionary for our ERD.

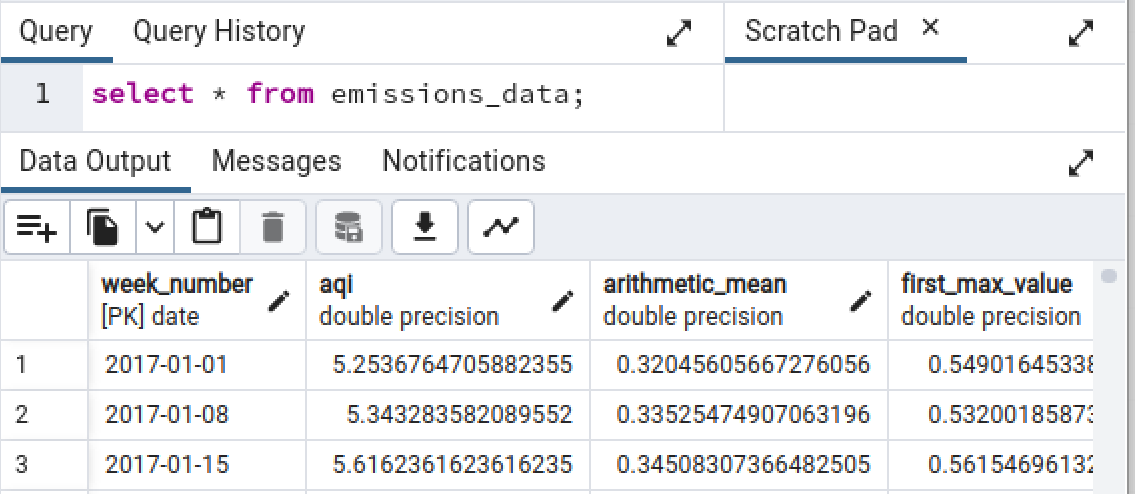
* Table: Economic Indicators
  + week\_number: represents the week start day
  + gdp: gross domestic product in billions of dollars for the USA
  + real\_gdp: real gross domestic product in billions of dollars for the USA
  + unemployment\_rate: rate of unemployment in the USA as a percent
  + interest\_rate: rate of interest (cost of borrowing money) in the USA as a percent
  + funds\_rate: federal fund rate in the USA as a percent
* Table: Emissions
  + tbd
* Table: Covid
  + tbd
* Table: S&P 500
  + tbd

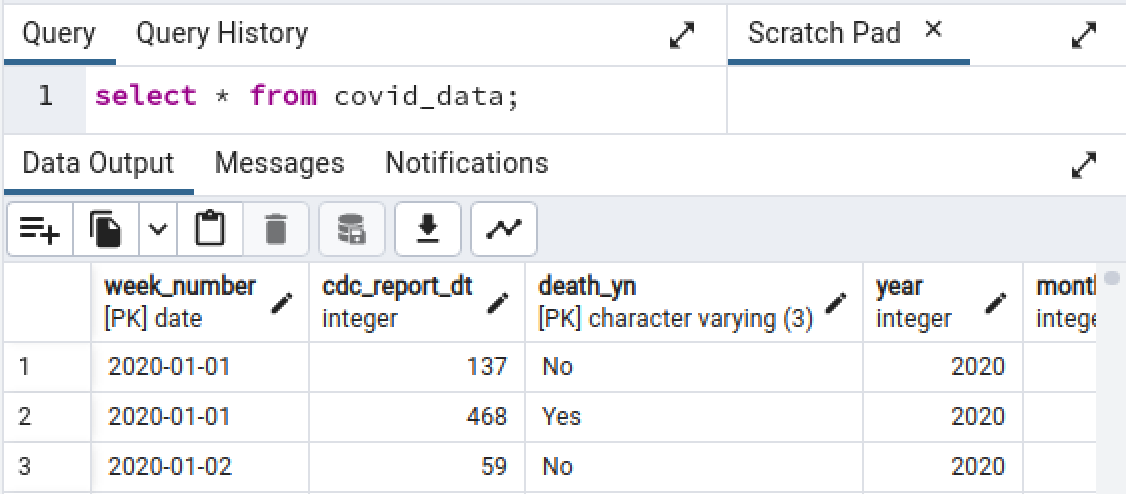
# Create Schema, Tables, and Load Data

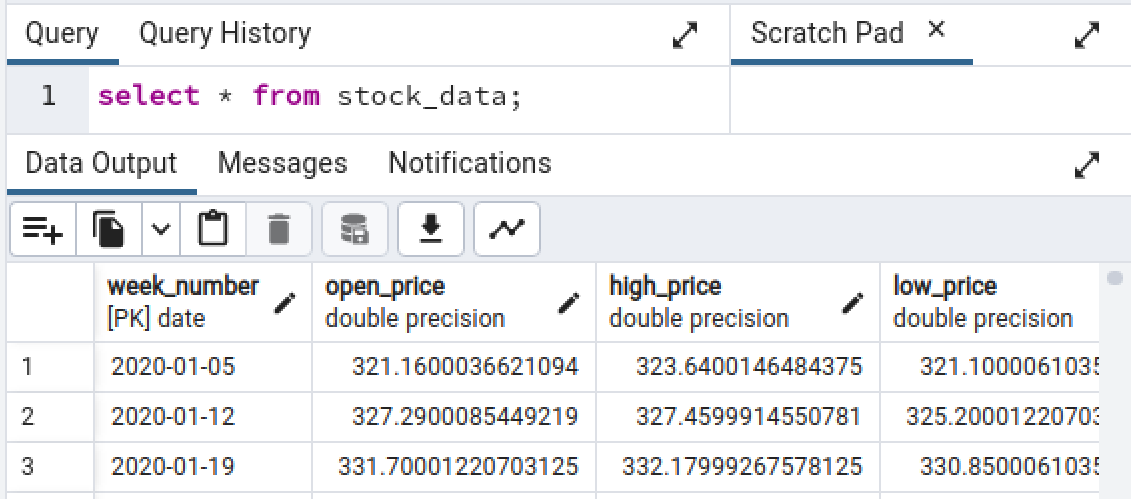
The schema and tables are created in the [covid\_schema.sql](https://github.com/dandrea1/EN.685.648.81.FA23/blob/main/covid_schema.sql) file. Also, the cleaned .csv data is loaded into the tables using the COPY method.











# Automation

For our automation we built an Airflow DAG that will run Python scripts that will update our data weekly. The EPA and stock data are available through API so these scripts pull the past week of data from the API and will insert it into our covid database. The CDC and Economic data are set up so that if a csv is dropped into the airflow scripts folder then the previous weeks data will be pulled from the csv and inserted into the database. The scripts for CDC and Economic data detect if there is a csv file to pull from, and if there is no file it simply prints “No updates for CDC data” or “No updates for Economic Data”.

The DAG is as follows:

A diagram of a software process

Description automatically generated

This DAG executes weekly. All four data sources pull data from the previous week and then update the database. A Flask API is then started, and a report is generated in the API. After this script executes the Flask API remains open, and reports on each of the four data sources can be pulled. For example, <http://localhost:8001/api/covid> returns the covid data from the database in json format.

# API

Our Flask API allows access to the covid schema contents, and has GET functions for each of the four data sources. This API runs on localhost 8001, and remains running after completion of the DAG, or can be activated manually to get data from the covid schema.

# Step by Step instructions to run the pipeline

1. Open and login to the class Virtual Machine.
2. Download the project zip\_file to your Downloads: **<name of file>**
3. Extract the files.
4. Open Postgres and copy paste the [covid\_schema.sql](https://github.com/dandrea1/EN.685.648.81.FA23/blob/main/covid_schema.sql) file into the query box.
5. Run the query. The schema and database should be created and the data loaded (see above screenshots).
   1. *Note to the reader: If the .sql file does not run due to permissions error. Follow the solutions in this stack overflow:* [*https://stackoverflow.com/questions/50273803/postgresql-could-not-open-file-for-reading-permission-denied-sql-state-42501*](https://stackoverflow.com/questions/50273803/postgresql-could-not-open-file-for-reading-permission-denied-sql-state-42501)
6. tbd

# 