|  |
| --- |
| UC Berkeley |
| ETL Project |
| Data Analytics Bootcamp |

|  |
| --- |
| Sofia Mbega, Lawson James, Cody Miracle, Jahangir Dewan & Sajad Yarzada  11-14-2020 |

# **ETL Project**

## **Overview and Purpose**

The objective of the project is performing a complete ETL process using three different datasets and an API on United States unemployment rates, diversity index and median income:

1. **Extracting –** data from three different sources to show unemployment rates, diversity indexes and the median income for all US counties.
2. **Transformation –** finding the relevant variables, dropping unnecessary columns, and creating different relevant and useful tables
3. **Load –** Creating a database using MySQL and loading the Data Frames in a data base using SQLAlchemy and Postgress.

## **E- Extraction**

The following datasets were used on the US States and Counties’ unemployment rates, diversity index and median income:

* US Counties Diversity Index - <https://www.kaggle.com/mikejohnsonjr/us-counties-diversity-index>
* US States and Counties Median Income - <https://data.world/tylerudite/2015-median-income-by-county>
* US States and Counties Unemployment - <https://www.kaggle.com/jayrav13/unemployment-by-county-us>
* US Bureau of Labor Statistics - <https://www.bls.gov/bls/api_features.htm>

## **T- Transformation**

### **Diversity Index Data Set**

* The first step was to load the diversity index dataset. The “location” column is separated in to two columns to show “States” and “Counties” separately (Table 1). This will help us in merging and grouping with the other two datasets.

A picture containing calendar

Description automatically generated

***Table 1:*** *Splitting location into County and State in the diversity dataset*

* A new table is created by grouping each race by “State” and “County” (Table 2)

A screenshot of a computer

Description automatically generated

***Table 2:*** *Grouping the new data frame by “State” and “County”*

### **Unemployment Rate Dataset**

* The “State” column in the unemployment dataset used the full name of the states (Table 3). To use this column for merging with other datasets, we had to abbreviate it. A dictionary of states with their abbreviations were used to loop through the dataset. The year and month columns were deemed not useful thus, they were also dropped. (Table 4).

Table

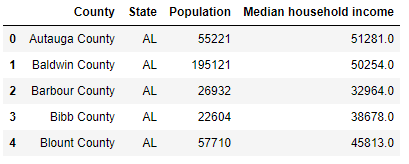
Description automatically generatedTable

Description automatically generated

***Table 3:*** *Full States Names* ***Table 4:*** *Abbreviated States*

### **Median Income Dataset**

* From the Median Income Dataset, we dropped several unnecessary columns, and renamed the “State-Code” column to “State” as that will help us in merging all three datasets in the next step. (Table5).



**Table 5:** *Unnecessary columns dropped*

### **Merging Tables**

* First, the Unemployment Rate dataset is merged with Median Income dataset on State and County. (Table 6)

Table

Description automatically generated

***Table 6:*** *Unemployment and Median Income Datasets Merged*

* Then, this new table containing unemployment rate and median income was merged with the diversity dataset again on State and County, using an inner join. (Table 7)

A screenshot of a computer

Description automatically generated

***Table 7:*** *All Three Datasets Merged*

* Finally, with all three datasets merged in to one table, the State index was reset to turn it back in to a column. (Table 8)

A screenshot of a computer

Description automatically generated

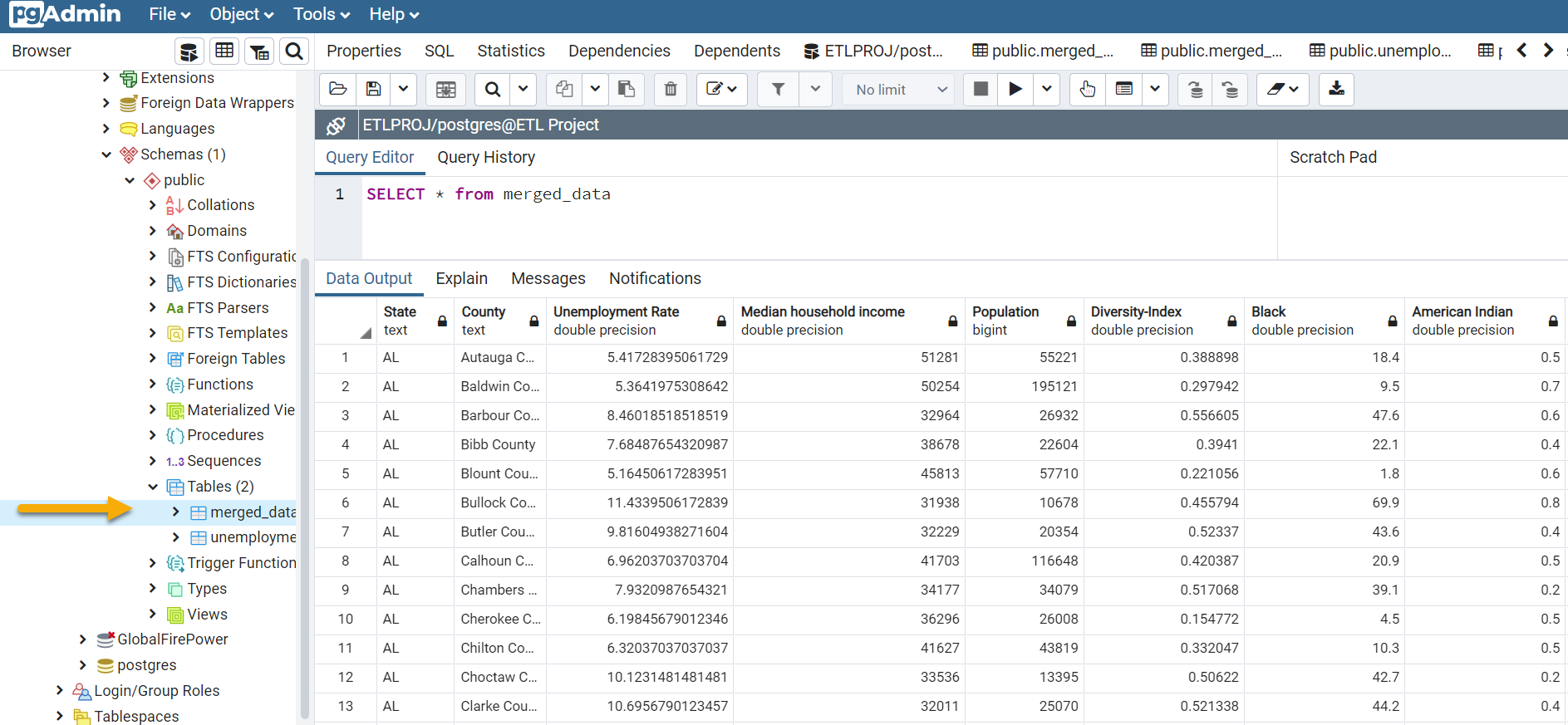
***Table 8:*** *Final table with State column as index*

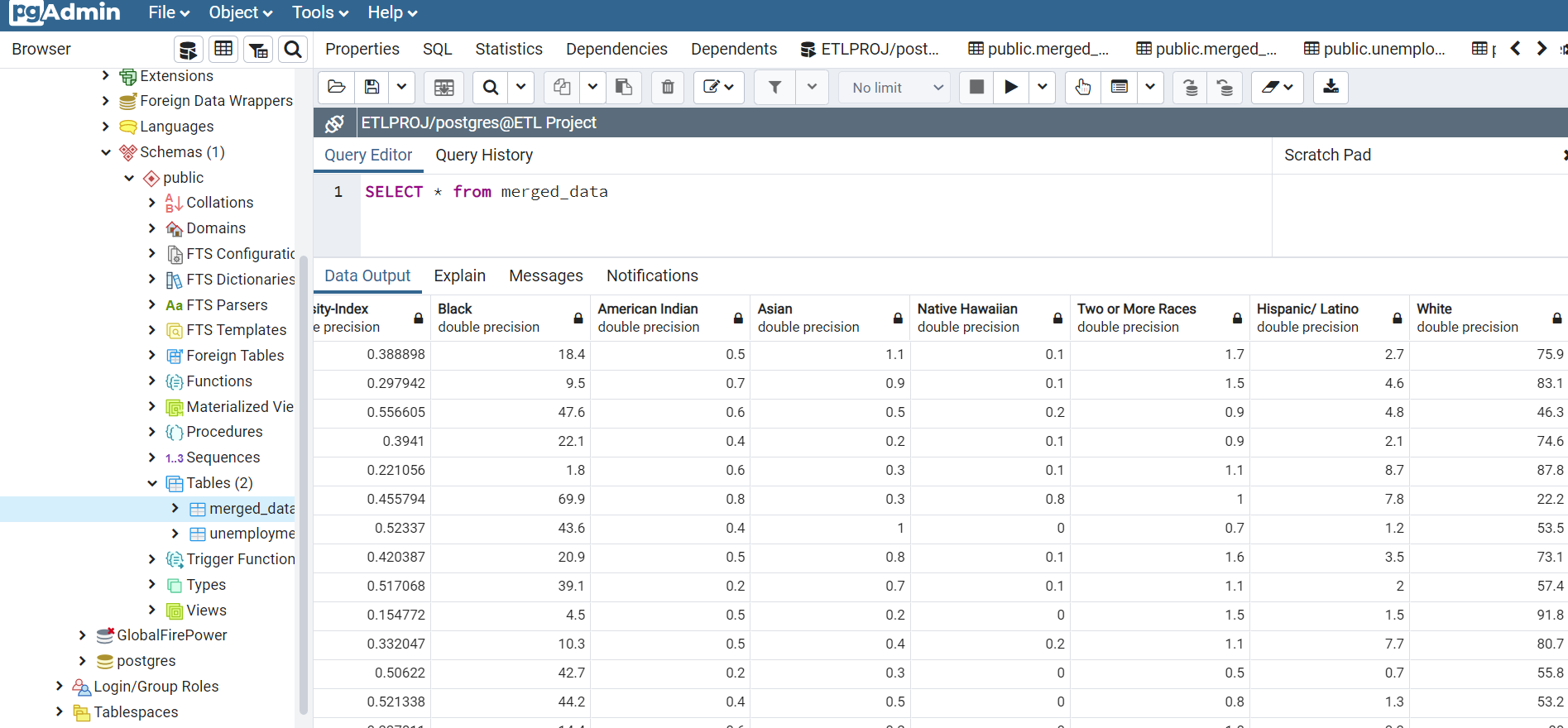
## **L – Load**

After the transformation stage in which we cleaned the data and created several usable tables. A database was created to match the columns from our final data frame in pandas using SQL and Postgress. Data was loaded to the Postgres using the following processing logic:

* Created the db connection through sqlalchemy
* Imported the create\_engine function
* Authenticated access to the db using my username and pwd
* If the table existed, it was dropped
* Created the table
* Committed the creation
* Inserted the merged data to the table

The data frames were then connected to the database using SQLAlchemy to load the results.





***BONUS***

Load the unemployment JSON to a postgres db.

