ETL Project Report

***Extract:***

We searched for data containing the U.S. populations and found the site: [State Population Totals: 2010-2019 (census.gov)](https://www.census.gov/data/tables/time-series/demo/popest/2010s-state-total.html#par_textimage) which contained datasets [Index of /programs-surveys/popest/datasets/2010-2020/national/totals (census.gov)](https://www2.census.gov/programs-surveys/popest/datasets/2010-2020/national/totals/) and downloaded the *nst-est2020.csv* which contained a worksheet State Population for the U.S., state, and regional populations from 2010 thru 2020.

Next, we searched for data related to U.S. senators and found the site: <https://ballotpedia.org/List_of_current_members_of_the_U.S._Congress>. We scraped the html data from the website that contained the senator content we were looking for.

***Transform:***

Read the csv into a Pandas data frame and reviewed the columns. Captured the columns that would become part of the final csv. Renamed the columns. Checked the count, data types, and dropped extraneous rows. Reindexed and renumbered columns. Rearranged and renamed the columns to allow for a smooth transition into the new database. Returned a clean csv file to the Resources folder.

We then scraped the U.S. Senators from the Ballotpedia website by storing the URL in a Pandas data frame. Grabbed the senators table from the website and stored it into a data frame. Viewed the table and decided to split the content of the office column on the spaces between words. Created two new columns and then split the state column which created an additional column. Dropped the unnecessary columns and renamed the columns for smooth transition into the database. Joined the two data frames and reviewed the content. Converted the senator data frame into a cleaned csv.

***Load:***

Using PgAdmin, we created the senator\_db relational database and used SQL to create the tables and set the primary keys. Then, we went back to the Jupyter Notebook and added code to connect to the database and populated the tables. Lastly, we performed a few queries to validate the functionality of the newly created database.

***Lessons Learned:***

During our creation process we found that the Jupyter kernel needed to be refreshed often. We found that it was essential that the table and csv columns match exactly; specifically, the column heads needed to all be lowercase. We were presented with difficulty at first when adding the primary key, but we came to realize the order in which we added the primary key to the code made a difference. We worked as a team to resolve issues as they were discovered, given more time, we would work to further enhance our database and tables. Although we were unable to produce a foreign key given the project time constraints, nevertheless, we have created a solid and fully functional database upon which queries can be made.

***Future Database Analysis:***

As suggested, we believe that our database is fluid and with more time and resources we understand that with more data added, additional queries could be made upon our database. Some examples areadding columns to our tables such as *the year senators assumed office, political affiliation of the state, and what term a senator was in.* These additional data points would allow us to perform aggregate analysis such as the average number of terms of senators, most freshmen and senior senator, and possibly the average changeover timeframe between new senators being elected after a term or the incumbent remaining in office based off them coming from a red or a blue state.