**\*\*E\*\*xtract:**

Our group selected data from the [North Carolina Board of Elections](https://dl.ncsbe.gov/?prefix=data/). Specifically, we pulled in data from the [Voter Registration](https://s3.amazonaws.com/dl.ncsbe.gov/data/ncvoter11.zip) information for Buncombe County and the [Voter History](https://s3.amazonaws.com/dl.ncsbe.gov/data/ncvhis11.zip) information for Buncombe County. These files were in csv format, with tab delineation and encoded with cp1252 encoding.

**\*\*T\*\*ransform**:

First the two csv files were read into Pandas as dataframes. Upon attempting to push this initial work to our git repository we discovered that the files were too large for github. At that point we used pandas to split each csv file into two files (four files total). These smaller files were then read into a new notebook and appended to recreate the initial dataframes. Then unnecessary columns were dropped. We dropped 4 columns (absent\_ind, name\_prefx\_cd, dist\_2\_abbrv, dist\_2\_desc) that were described in the online documentation as not being used. We also dropped the columns that described the different districts the voter was assigned to because they were larger than the district abbreviation columns and duplicated its information. We also dropped the confidential\_ind column to remove any confidential information that might have been stored in the voter registration dataframe. We pulled out personally identifying information from the voter registration file into its own dataframe and also pulled out demographic information columns to create a voter demographic dataframe. The personally identifying information was moved into its own dataframe in order to allow us to make that information more protected if needed. In order to reduce the size of our voter registration dataframe, we also created county, ethnic, race, and status code dataframes using the code keys provided on the NC Board of Elections website. The voter history file downloaded from the North Carolina Board of Elections website was loaded into its own dataframe.

**\*\*L\*\*oad:**

Using Heroku, we created a team “voter-db” application, connected to Github, in order to query on our own machines using PGAdmin. The smaller code description databases were easily inserted and populated appropriately. We encountered a number of errors and challenges when attempting to insert the three larger voter dataframes. Ultimately we discovered that the Hobby Tier of Heroku that we were using only allowed for a maximum of 10,000 rows which our dataframes far exceeded. To resolve this issue in the short term we created subset dataframes of 10,000 rows which inserted and populated as expected. To scale this project a paid Tier of Heroku would be required. The final production database is a relational database. Tables included are listed on the ERD diagram below. Ultimately (outside of this project) we envision our user interface allowing users to merge tables, query subsets of data and generate and export that data. This project has the flexibility to scale up by streamlining the ETL process, allowing for additional county and state level data to be added and may be of use to local political candidates and action groups.

