**ETL Project**

Technical Report

Seth Newcomb, Ryan Zimmerlee, Bryan Wood, Qin Gui

**Purpose:**

This document describes our process of extracting, transforming, and loading data into a database.

**Project Summary:**

We chose to utilize sources related to 2016 Election and demographic Census data in order to build a relationnal database that could be used to perform analysis and identify trends pertaining to the 2016 Election.

**Data extraction:**

Our data consisted of two main sources:

1. 2016 Election result data broken down by FIPS (Federal Information Processing Standard) code across the country. This data included total voter counts for each presidential party, total votes, state, county name and FIPS code.
2. Census Data retrieved by utilizing the Census API. Areas we found to be interesting related to each geographic location included; median age, population and education level.

We noted that the Census data included the fields “State Code” and “County Code”. These two values combined resulted in the FIPS code for that area which is ultimately how we could relate the two datasets on a one-to-one relationship basis. The 2016 Election Data was in CSV format, while the Census API data returned in JSON format.

For the election data, our approach included utilizing python and reading the CSV into a pandas data-frame. For the Census Data information, we researched the documentation to find related codes to pass through the request to retrieve “Median Age”, “Population”, “No education”, “High School”, “GED”, “Associates”, “Bachelors”, “Masters”, ”Professional Degree”, ”Doctorate”.

**Transformation:**

Our transformation and data cleansing process included reviewing the 2016 election data csv file and noting that for some of the locations, they had FIPS codes of 4 digits instead of 5. We defined a function that added a 0 to the front of those values lower than 10000. The 5-digit code better resembled a true “Zip Code” and matched up better with the Census Data. From there we created a new column in the Election Data data-frame which contained the 5-digit code and we then dropped the old code column. We also noted that for Alaska (state code “AK”), all lines repeated the exact same information. Without knowing for sure, our best guess is that detailed data was not provided for Alaska so they were just carried over each location. For that reason, we decided to remove rows that referenced the state code “AK” to have a true dataset since we were unable to distinguish differences in election data for the specific areas in Alaska.

For the census data , we removed spaces in the column headers and replaced them with underscores to translate efficiently into a SQL database. We also renamed the columns, and organized them to display the information more effectively. The Census data split out each location by a two digit “State Code” and a three digit “County Code”. We combined the two columns to create a similar five digit “Fips” column and dropped the old two columns.

**Load:**

For our loading process we chose to create a MySQL database utilizing MySQL workbench. Since these two datasets were relational this made the most sense. Once the schema was create in MySQL workbench we were able to utilize the pandas dataframe.to\_SQL method and created two tables, one for each dataset. The primary key being a unique id for each record in the tables, and the FIPS code being the Foreign key between the two tables.