ETL Project

Final Report

Team Members: Kaylie Sheehan, Vickie Hughes, Richard Butler, & Carlos Villanueva Jr

Extract

For the given ETL project, our original two sets of data came from two different databases that were provided in the website Kaggle. The first source of data is called “Datafiniti\_Fast\_Food\_Restaurants\_May19”, and it is formatted as a CSV file. In this dataset we have a list of 10,000 fast food restaurants that contain their address, city, latitude and longitude coordinates, and name. The file was published in the dataset section under the name “Fast Food Restaurants Across America”, and the original file utilized in this exercise can be obtained through the following link under the download option: :<https://www.kaggle.com/datafiniti/fast-food-restaurants>

The second source of data that was acquired for the completion of the task given is called “population\_by\_zip\_2010”, and it is formatted as a CSV file. In the dataset we are given the census data for the years 2000, and 2010 that are broken down by gender, age and location with the use of zip codes and GEOIDs. This file was found in the dataset section with the name “US Population By Zip Code”, and the original file can be extracted from the link provided below in the download section:

<https://www.kaggle.com/census/us-population-by-zip-code?select=population_by_zip_2010.csv>

Transform

The transformation process began by working on the csv file called “Datafiniti\_Fast\_Food\_Restaurants\_May19”, and the first step was to load the file and store it in a dataframe. The file contains information that is irrelevant to the desired analysis, which is the reason behind us creating a new dataframe that only contains the columns with the information needed. The columns in the new dataframe were given new names in order to make them more easily interpretable. In this dataframe, our column of interest is called zipcodes, and these were stored in the original file with the extended ZIP+4 code. However, we require these codes to be in the basic format that consists of five digits. This led us to split the column zip codes by the zip and the +4 extension code, but the +4 column was removed from the dataframe. Before going forward, the data type for zip code was object, which was changed to the int data type. At this point, transformations to the data frame from fast food restaurants were done, and it was ready for merging.

We continued our transformation process by now loading our second csv file named “population\_by\_zip\_2010”, and this was stored into a dataframe. From this dataframe we only need two columns containing population, and zip code information, which we decided to put in a new dataframe. The next step taken was to group the new dataframe by zip code. This gave us a dataframe that was ordered by the zipcode entries with the respective population for the area. Now the next step was to use sum on population by zip code in order to compress our data, and have the transformed data frame from population ready for merging.

Finally, the transformed data frame from food restaurants, and the transformed data frame from population were merged together on zip code. This new merged data frame was checked for any missing values, and those that were n/a were dropped from the final dataset. At this point, there was no further transformation needed, and the data was ready to be analyzed.

Load

For the load process we started by deciding between using PostgreSQL or MongoDB. After weighing the pros and cons of each we decided that PostgreSQL would be the appropriate choice due to our data having a tabular structure.

After deciding on PostgreSQL, we continued by creating the tables in a sql database named fastfoods. The first table we created was fastfood, it is made up of columns for FFID, DateAdded, BusinessName, Address, city, State, Zip, and Category. All these columns were text with the FFID as the primary key. The second table we created was the population table, this table included columns for id, population, minimum\_age, maximum\_age, gender, Zipcode, and geo\_id. Id was a serial and our primary key while population, minimum\_age, maximum\_age were integers and gender, Zipcode, and geo\_id was text.

After these tables were created, we took the dataframes we had transformed and then exported them into the sql database. We exported the new\_fastfood\_df to the fastfood table and the population\_data\_df into the population table.