ETL Project

Group 3

Group Members: Kevin Bader, Kendall Sisk, and Josh Heisler

City Obesity and Number of McDonald's

Extract:

To begin this project, we all selected a topic and rated our favorite topic out of all three. We chose the topic of "fast food" then narrowed it more specifically to McDonald's fast-food chain. We then thought about what we wanted our data to show about McDonald's. In the end, we found a CSV file of all the United States McDonald's' locations and a CSV file by the CDC that showed obesity percentage by city. Our CSV on McDonald's (which we found on Kaggle.com) consisted of 27 columns and 29K rows. The data showed geometric coordinates, three separate columns for address, one of which showed the city, multiple columns for the hours they were open, and many more that were not important to our project. The CSV on obesity by city (which we found on the CDC's website) consisted of 21 columns and 13K rows. This data showed the year, state, city, percent obese, population, and multiple other columns were not crucial to our project.

Transform:

We started out thinking the transformation would be a piece of cake; however, there were many compatibility issues with our data that we had to fix. To start, we used the city name to match our two data sources. One of our CSV files had the city in all capital letters, and the other was in all lowercase letters. To fix this issue, we made the CDC data display all capital letters so they would match. Next, we had a lot of unnecessary columns that we needed to remove from both CSV files. We used Pandas for the removal of them. After that, we needed to convert the "population" data to be an integer since it was a string variable with commas in the CDC file. We again used Pandas to remove the commas and convert the variable to an integer. Finally, we noticed each city was printing out several times since the CDC data used specific locations in each city. To fix this issue, we grouped by city and used the aggregate function to get the average BMI in each town and each location's total population. These last changes used Pandas in Jupyter Notebook, and then we imported it from Jupyter Notebook to Postgresql.

Load:

We chose to use SQL as our database since we were using structured CSV data files. We started by creating a database in SQL. We then imported the data we had set up in Jupyter Notebook over into SQL. We then joined the "McDonald's" data with the "Obesity" data and counted the number of McDonald's there were. We then averaged the BMI and population. The reason we did an average of the "BMI" and "population" in SQL (even though we did all the converting in Jupyter Notebook) was because when we joined the data, it created duplicate rows of the same population and BMI. By averaging the same numbers for both columns, it did not affect our data in any way. In the end, we had a table that showed the number of McDonald's in each city, the city name, the city's obesity percentage, and the population of that city.