**ETL Project Report: Aggregating Data for Analyzing Weather Patterns and Speeding Violations in Chicago**

**Team members: Eric Yang, Eduardo Landeros, Ann Marie Zaya, Tracy Guo, Yuxing Wang**

**Data Sources:**

*Speeding Datasets*

<https://www.kaggle.com/chicago/chicago-red-light-and-speed-camera-data#red-light-camera-violations.csv>

*Weather Dataset*

<https://www.kaggle.com/yochanan/chicago-weather>

**Summary:**

We have found a CSV file for Chicago City speed camera locations and a CSV file for speeding violations in Chicago. Our team decided it would be interesting to link the data and come up with the number of violations at each speed camera location. It would also be interesting to plot the geo-locations on google maps to look for speeding violation hot spots. Next, we would like to see the relationship between weather and number of violations, so we looked for weather data of Chicago. We were able to find a CSV file for weather data of the Chicago Midway airport since 2008 and it can be used as a proximate of weather condition in Chicago. Linear regression could be used to analyze the relationship between weather conditions (precipitation, snowfall, temperature, visibility) and the number of speed violations in multiple locations of the city and in Chicago overall.

The goal of this project is to create a SQL table with clean data ready for above mentioned analysis.

**Technical Steps (please see the Python notebook attached with all our codes in it):**

**Extract:**

We downloaded the three CSV files from Kaggle and loaded them into one Python notebook using pandas. We browsed each of the file to check column names and data types.

**Transform:**

We cleaned up the “ADDRESS” column in the speeding camera locations dataset to be the same format as the “ADDRESS” column in the speeding violations dataset. To do this, we stripped the the “ Speed Camera” string in the speeding camera location dataset’s “ADDRESS” column. We then dropped unneeded columns in the speed camera location and speed camera violation dataset and renamed the rest of the columns to ensure the naming is concise and clear. We then moved on to the weather dataset, where we only extracted the time component (“DATE” column) of the daily weather measurement. We then merged the weather dataset with the cleaned, merged speed locations and violations dataset using inner join.

**Load:**

After cleaning the data and merging the datasets, we created a database in MySQL titled Speed\_Violations\_DB. We also created a table titled speed\_violations\_weather in Python notebook. Then, we created an engine in the Python notebook to establish a connection between the SQL database and the notebook. Next step was to push the merged dataset into the speed\_violations\_weather in the SQL database. After that in MySQL we altered the table to add a primary key and re-ordered the columns to have the columns with most useful data in the front.

Thus far, the table is ready for analysis of relationship between selected weather conditions, such as visibility and snow depth, and the number of speeding violations at each camera location.