ISSUED PARKING TICKETS ON TORONTO GREEN P PARKING SPACES

**MEMBERS**

Christopher Habib

Anna Francesca Gatus

Siddharth Krishnan

**INTRODUCTION**

The Toronto Parking Authority is a local Board of the City of Toronto which owns and operates the system of Municipal off-street parking lots ('Green P') and the on-street metered parking. Approximately 2.8 million parking tickets are issued annually across the City of Toronto. The Issued Parking Tickets dataset contains non-identifiable information relating to each parking ticket issued for each calendar year. The tickets are issued by Toronto Police Services (TPS) personnel as well as persons certified and authorized to issue tickets by TPS.

Our group chose to combine 2015 Issued Parking Tickets and Green P Parking. Final table has the following columns: Parking ID, Parking Rate, Address, Infraction Description and Set Fine Amount.

**METHODS**

Using the ETL processes, the following tasks were done:

Extract:

1. Extracted 2015 Issued Parking Tickets Data from Toronto Open Data Catalogue.

Source:

<https://www.toronto.ca/city-government/data-research-maps/open-data/open-data-catalogue/#75d14c24-3b7e-f344-4412-d8fd41f89455>

Link to three .csv files:

<https://github.com/chabib456/ETL-Project/tree/master/Resources/parking_tickets_2015>

1. Extracted 2015 Green P Parking Data from The Toronto Parking Authority Open Data Catalogue

Source:

<https://portal0.cf.opendata.inter.sandbox-toronto.ca/dataset/green-p-parking/>

Link to Json file:

<https://www.toronto.ca/ext/open_data/catalog/data_set_files/greenPParking2015.json>

Transform:

1. Issued Parking Tickets Data
   1. Used Python Pandas library to load and read the three .csv files.
   2. Used pd.concat function to combine the three DataFrame results
   3. Stored addresses by selecting location 2 column and putting it in a list
2. Green P Parking Data
   1. Used Python Json library to load and read json file
   2. Used a for loop to collect parking id, address and rate data and stored information to corresponding lists.
   3. Results were saved as DataFrame
   4. Used Python pandas library to convert lists to DataFrame
   5. Stored addresses by selecting address column and putting it in a list
3. Addresses of Issued Parking Tickets must be transformed to be identical to addresses of Green P Parking so that it can be merged.
   1. Converted address list to upper case
   2. Removed dots from address
   3. Used a for loop and if, elif, else statements to change the following
      1. east to E
      2. west to W
      3. street to ST
      4. blvd to BLVD
      5. avenue to AVE
      6. road to RD
      7. dr to DR
      8. circle to CRCL
      9. lane to LANE
      10. drive to DRIVE
   4. Stored cleaned up data to a list called streets
   5. Verified if the counts of common addresses on both datasets match
   6. Merged two DataFrames using the clean address column

Load:

1. Created SQL connection
2. Exported to MySQL. Since the final output is a DataFrame, we decided to load the data into a relational database.
3. Final table to be used in the production database has the following columns: Parking ID, Parking Rate, Address, Infraction Description and Set Fine Amount. Reason why these columns were selected is to determine possible relationship between parking rate vs infractions, and parking rate vs set fine amount. Other analysis that can be done could be: Which location has the highest infraction? What infraction is the most common? Does high parking rate causing infractions? Does high fine prevent infraction?