Brandon Coleman

Suresh Sreerameneni

ETL Project

Technical Report

* The sources of data that you will extract from.
* The type of transformation needed for this data (cleaning, joining, filtering, aggregating, etc).
* The type of final production database to load the data into (relational or non-relational).
* The final tables or collections that will be used in the production database.

Our data was extracted from crime.csv and offense\_codes.csv file provided by the Denver Crime Data dataset on Kaggle.

After analyzing crime.csv data we excluded five columns using pandas : 'GEO\_X', 'GEO\_Y', 'GEO\_LON', 'GEO\_LAT' because the dataset already contained the incident address, district\_id as well as the neighborhood name. We additionally dropped ‘OFFENSE\_TYPE\_ID’ due to redundancy.

Crime.csv was separated into three tables, for normalization, crime\_table, crime\_place\_table and crime\_time\_table.

For offense\_codes.csv we dropped three columns using pandas : 'OFFENSE\_CODE', 'OFFENSE\_TYPE\_ID', 'OFFENSE\_CATEGORY\_ID', due to redundancy. We combined ‘OFFENSE\_CODE’ and ‘OFFENSE\_CODE\_EXTENSION’ renaming the column ‘offense\_code\_ext’.

We loaded our transformed data as a relational database following are the tables :

crime\_table, crime\_place\_table, crime\_time\_table and offense\_codes.

Finally, to confirm that our data was correctly loaded into postgres, we queried our database and four tables.

As for choosing postgreSQL, apart from familiarity, postgres gives us many useful data types in addition to the most common ones. For example, in our crime\_time\_table we were able to use a TIMESTAMP datatype instead of having to separate our columns by date and time of day.