**Project 3: OpenStreetMap Data Wrangling**

***Submitted by: Naga Ramesh Kamisetti***

**Overview:**

This project deals with Data wrangling, and using SQL queries to mine data.

**Introduction:**

OpenStreetMap Data is accessible to everyone. Anyone can edit the data in the website which is both the advantage and disadvantage. There is a high chance of human errors in the map files. I used the API to get the map details for College Station-Navasota area, Texas.

OpenStreetMap link: [link](https://www.openstreetmap.org/#map=5/51.500/-0.100)

**API link:** [API LINK](http://overpass-api.de/query_form.html)

**Example Query Syntax:** (node(51.249,7.148,51.251,7.152);<;);out meta;

latitude and longitude in the order of S, W, N, E

**wiki:** [link](http://wiki.openstreetmap.org/wiki/Main_Page)

General File structure:

The file format is osm xml. The entire map area is defined by different type of [Elements](http://wiki.openstreetmap.org/wiki/Elements) like, [nodes](http://wiki.openstreetmap.org/wiki/Node) , [ways](http://wiki.openstreetmap.org/wiki/Way) , [relations](http://wiki.openstreetmap.org/wiki/Relation) , [tags](http://wiki.openstreetmap.org/wiki/Tags) .

**Problems encountered in the Map:**

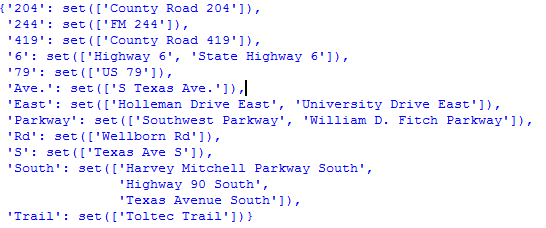
***Inconsistent Postal codes:***

While almost all postal codes have general five-digit postal code, in some places there is a prefix of “TX” representing Texas. This is cleaned using the help of **update\_postalcodes** function in the **auditing.py** file.



These are represented here as key- value pair where key represents the postal code and value represents the number of times that particular code appeared in the file.

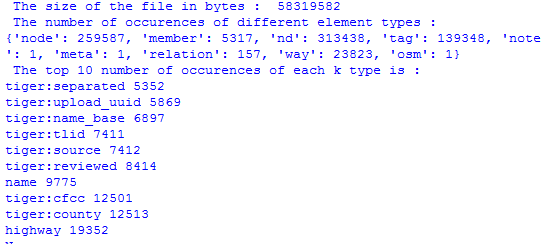
***Inconsistent street types:***

Abbreviations were used for some of the street names like “Ave.” for “Avenue”. This problem is fixed by using the **update\_name** function.

Using the update function, the results are like:



**Overview Statistics about the Dataset:**





These values matched exactly with the values obtained after exporting to csv’s. For example, the number of node elements are 259587 which exactly matched with number of rows in the **nodes.csv** file.

SQL Queries on the tables created using the csv files.

1. Number of Node elements:

**SELECT COUNT(id) FROM nodes;**

Ans: 259587

1. Number of way elements:

**SELECT COUNT(id) FROM ways;**

Ans: 23823

1. Number of Unique users who contributed for node elements:

**SELECT COUNT(DISTINCT uid) FROM nodes;**

Ans: 259

1. Number of Unique users who contributed for way elements:

**SELECT COUNT(DISTINCT uid) FROM ways;**

Ans: 214

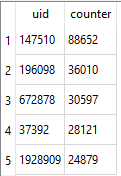
1. Number of users contributing only once in node elements:

**SELECT COUNT(counter) FROM (SELECT uid, count(uid) as counter FROM nodes GROUP BY uid ORDER BY counter) sub WHERE counter = 1;**

Ans: 41

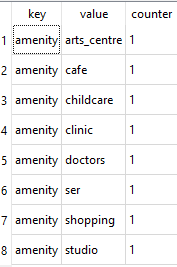
1. Top 5 contributing users:

**SELECT uid, count(uid) as counter FROM nodes GROUP BY uid ORDER BY counter desc LIMIT 5;**



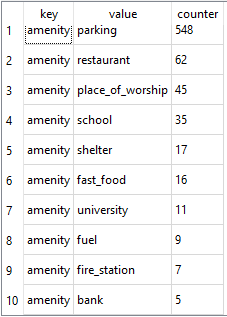
1. List of Amenities that appeared only once:

**SELECT key, value, counter FROM (SELECT key, value , count(value) as counter FROM waytags WHERE key = "amenity" GROUP BY value ORDER BY counter desc) sub WHERE counter = 1;**



1. What are the most common amenities:

**SELECT key, value , COUNT(value) as counter FROM waytags WHERE key = "amenity" GROUP BY value ORDER BY counter desc LIMIT 10;**



1. What is the most common city?

**SELECT key,value, count(value) as counter FROM waytags WHERE key ='city' GROUP BY value ORDER BY counter desc LIMIT 1;**

Ans: Navasota – 157 times

1. The way id which is defined by more number of nodes:

**SELECT id,max(maximumnodes) FROM(SELECT id,max(position) as maximumnodes FROM waynodes GROUP BY id) sub ;**



According to the Openstreetmap wiki, maximum of 2000 nodes were used to define a single way.

1. How many Users edited both node elements and way elements?

**SELECT count(DISTINCT (nodes.uid)) FROM nodes INNER JOIN ways ON nodes.uid = ways.uid;**

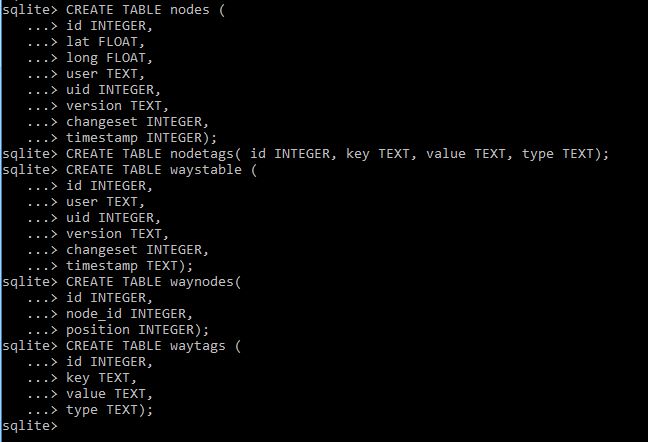
Ans: 180

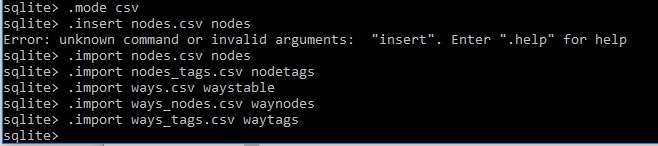
**Additional Improvements:**

The city name can be audited the same way as street names. There is a suffix “Tx” in some city names. This helps in further clean data that can be used for further exploratory analysis. As the latitude and longitude are known, data from other sources like GOOGLE MAPS can be used to find the validity, and in some cases both the data can be compared to prepare better maps. But, the problem is that Google maps data is not available for public freely. One can use this data to create database of building names, type of amenities which can be used for creating a library of the city data. One can use this data to know the different type of amenities in a street and find out how suitable living in that street for a particular person based on his/her needs.

**Additional Information:**

**Preparing the SQL tables:**

****

****

**I used DB browser for sqlite for performing the SQL queries.**