Wrangle OpenStreetMap Data Using SQL

Part 1 - Map Overview and motivation

I live and work in Bangalore (Bengaluru) for the last 10 Years . That ignited my curioisty to audit Bangalore OSM maps

https://s3.amazonaws.com/metro-extracts.mapzen.com/bengaluru_india.osm.bz2 (https://s3.amazonaws.com/metro-extracts.mapzen.com/bengaluru_india.osm.bz2)

In [1]:
```
OSM_FILE='bengaluru_india.osm'
```

Checking the total number of different tags in the OSM file

In [2]:
```
import MapParser
MapParser.test()
```

```
{'bounds': 1,
 'member': 6491,
 'nd': 3570439,
 'node': 2877979,
 'osm': 1,
 'relation': 963,
 'tag': 815458,
 'way': 659891}
```

Checking the user statistics

In [3]:
```
import users

users.test()
```

```
Total number of users:
1932
```

Part 2- Problems encountered in the map

Street Names

The main problem I could find with the street names was the improper street types. Even though this idea was explored during the course I could see that how much data were misrepresented because of the incorrect street types. For e.g Rd. for Road and St. for Street .

For audit the street type I have used the regular expression re.compile(r'\b\S+.?$', re.IGNORECASE) to pull out the last name from the street name and to add it to a dictionary with street type and frequency of it's occurence in the data set . Using this I could see that what were the errors in the data and how often it was misrespresented .

To correct the street type I have used a mapping table- dictionary .

Mapping dictionary

In [4]:
```
mapping = { "St": "Street",
            "St.": "Street",
           "street":"Street",
            "Rd.": "Road",
            "Circle)": "Circle",
            "Cit" : "City",
             "City," : "City",
             "Rd"  : "Road",
             "Road": "Road",
             "Rd." : "Road",
             "stn" : "Station",
             "galli" : "Gali",
            "Layou " :"Layout",
           "layout":"Layout",
           "Layout, ": "Layout",
           "Layout,.": "Layout",
           "main":"Main",
           "MAIN":"Main",
           "Naga":"Nagar",
           "Nagar, ":"Nagar"

           }
```

Postal codes / Zip Codes

Postal codes are one more value which I feel I can audit and correct. The reason for it was twofold.

1. In a country like India – because of the sheer number of duplicity of street name , postal code stands out as a real differentiator of a location and therefore supremely important for the user experience and accuracy of the addresses.
2. Postal code values follows a pre fixed pattern. For e.g. for Bangalore area the postal code values start with 560 and has 6 digits. This kind of structure makes it easy to audit and correct the values programmatically

Logic used for correction

1. Used the regex re.compile(r'560\d\d\d', re.IGNORECASE) to confirm the structure - the postal code values start with 560 and has 6 digits.
2. Removed the whitespace from the zip codes
3. Replaced O's with 0's .
4. Checked whether alphabet are present in the value.
5. Handled cases where the zip code is less than 6 digits .

Part 3- Overview of data

File sizes

banglore.db.....: 352M
bengaluru_india.osm.....: 618M
nodes.csv.....: 232M
nodes_tags.csv.....: 3M
ways.csv.....: 38M
ways_nodes.csv.....: 85M
ways_tags.csv.....: 23M

Number Of Nodes

SELECT COUNT(*) FROM nodes ;

There are 2877979 nodes in the data

Number Of Ways

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

There are 659891 ways

Number of unique users

```
SELECT COUNT(DISTINCT(u.uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) u;*
```

There are 1923 unique users . Please note that there is a difference in the number of users calculated in the first part of this report.

Number of schools in the city

```
SELECT COUNT() AS num FROM nodes_tags WHERE nodes_tags.value = 'school';
```

There are 416 schools in the city

Number of restuarants in the city

```
SELECT COUNT() AS num FROM nodes_tags WHERE nodes_tags.value = 'restaurant';
```

There are 1681 restaurants in the city

Part 4- Additional Analysis

Types of amnenties

```
SELECT value, COUNT(*) AS num
FROM nodes_tags
WHERE nodes_tags.key = 'amenity'
GROUP BY value
ORDER BY num DESC;
```

restaurant|1676
atm|790
bank|727
place_of_worship|694
pharmacy|544
fast_food|503
hospital|451
school|371
cafe|348
fuel|281

Here we can see that restaurant tops the amenity list . One more thing to notice is that Bangalore has around 348 cafes too .

Most popular cusisines in the city

SELECT value, COUNT(*) AS num FROM nodes_tags WHERE nodes_tags.key = 'cuisine' GROUP BY value ORDER BY num DESC;

regional|368
indian|290
pizza|89
vegetarian|88
chinese|79
ice_cream|55
coffee_shop|51
burger|46
international|32
italian|29

We can easily notice the cosmopolitan nature of Bangalore by the look of the cuisines. We have several pizzerias , coffee shops,cinese,italian cusines which marks for a city with well developed palatte.

Places of worship by religion

```
SELECT nodes_tags.value, COUNT(*) AS num
FROM nodes_tags
INNER JOIN
(SELECT DISTINCT(id)
FROM nodes_tags
WHERE value='place_of_worship') place
ON nodes_tags.id=place.id
WHERE nodes_tags.key='religion'
GROUP BY nodes_tags.value
ORDER BY num DESC;"
```

```
hindu|436
christian|80
muslim|58
jain|3
Sri_Anjaneya_Swamy_Temple|1
buddhist|1
```

This result is expected and is inline with the demographic numbers

Part 5- Improvement suggestions

1. Gamification to improve editing quality .

Pros

1. In a city like Bangalore where the IT literate people are more we can easily improve the quality of the data by setting up gamification challenge . We can also involve hobbyist groups like Free Software movement to organize hackathons and knwledge sessions about the structure of OSM .
2. Gamification and social proof can bring a lot of people getting intrested in the project. Wikipedia already proved that people would be ready and able to work on complex, time consuming tasks if they believe in the cause

Cons

1. We need to have proper audting in place for people intentionally muddying up the data .

2. Data restriction at the entry level

Pros

At this age of machine learning , AI and superior front end technologies mass editing but high impact tools like OSM maps should have more stringent and intelligent data validation process at place . For example it is known that the structure of the postal code is easy to follow thus by easy to reinforce a validation.

Cons

Data validation should not be so restrictive that it would discourage major chunk of the users from editing it .

Part 6 - Reference

https://github.com/lifengleaf/OpenStreetMap-Project-Udacity (https://github.com/lifengleaf/OpenStreetMap-Project-Udacity)
https://github.com/avs20/DWmongoDB (https://github.com/avs20/DWmongoDB)
https://discussions.udacity.com/t/display-files-and-their-sizes-in-directory/186741/7 (https://discussions.udacity.com/t/display-files-and-their-sizes-in-directory/186741/7)