Map Area: Sunnyvale

Below are the coordinates of the Sunnyvale city open street map data I will be working on. I have downloaded the data from the Overpass API link provided in the project details.

Minimum Latitude : 37.3499 Maximum Latitude: 37.3863 Minimum Longitude: -122.0654 Maximum Longitude: -121.9782

Below is the code to understand different tags in the data and counts against each of them.

```
{'member': 7369,
 'meta': 1,
 'nd': 293169,
 'node': 250586,
 'note': 1,
 'osm': 1,
 'relation': 499,
 'tag': 96905,
 'way': 35156}
```

I will be working on getting these into a database and analyzing further.

Problems Encountered in the Map

I have used the code I worked on in the exercise. Looks like there are lower colons in 32580 tags. I want to continue the way we have split the keys into type and the key.

```
{'lower': 63283, 'lower_colon': 32580, 'other': 1042, 'problemchars': 0}
```

I have got the list of all the users who have contributed in creating this data from nodes, ways and relations.

```
set(['25or6to4',
     '42429',
     'ACE|Tony',
     'AKOK',
     'Aaron Lidman',
     'Ahlzen',
     'Aidan703',
     'Alan Pierrat',
     'Alan Vogt',
     'Aleks-Berlin',
     'Alexander Avtanski',
     'AndiG88',
     'Andrew Hazlett',
     'AndrewSnow',
     'Ankur Datta',
     'Apo42',
     'Ashish Vijayaram',
      ……
     'warutledge',
     'werner2101',
```

```
    'wheelmap_visitor',
    'whereissean',
    'woodpeck_fixbot',
    'woodpeck_repair',
    'xybot',
    'yiqingj',
    'yurasi'])
```

Cleaning Street Names

There are only 2 places where the mapping I developed in the exercise was useful. Other street names which are listed above look genuine addresses only.

```
{'1': set(['Stewart Drive Suite #1']),
 '114': set(['West Evelyn Avenue Suite #114']),
 'Ave': set(['W Washington Ave']),
 'Circle': set(['Bobolink Circle', 'Continental Circle']),
 'East': set(['Vanderbilt Court East']),
 'Oaks': set(['North Fair Oaks']),
 'Rd': set(['Wolfe Rd']),
 'West': set(['Vanderbilt Court West'])}
-----------------------------------
Wolfe Rd => Wolfe Road
W Washington Ave => W Washington Avenue
```

Cleaning Phone Numbers

I started working on cleaning phone numbers and initial analysis showed me that there are too many formats in the phone numbers. I wanted to make all the phone number formats uniform. To do the cleaning I started listing down all the exiting formats in the phone numbers and tackling each of it to make it to the uniform format (+1XXXXXXXXXX).

Most of the formats were simple string edits except the phone numbers where the alphabets are used instead of numbers. I developed a function to convert these alphabets into appropriate numbers based on the T9 format (telephone numbers used to represent set of characters).

Challenges

I have worked on the creating the csv files from the xml using the code worked on during exercise. I faced some challenges in making this code work due to multiple reasons. But I figured out how to fix it based on the error messages and print statements at every stage to understand error.

Initially I could not get the packages Cerberus and Schema installed on my environment as conda was not able to search for them. Later I searched for a while to figure out the correct name it had to install with and some documentation on what these are used for. I enjoyed reading the material and understand how the complete code is working.

I also struggled using schema.py but the udacity forums helped me look for a solution for this issue. It is great learning for the future.

I faced challenges loading file into database. Initially when I tried opening the csv files in excel it showed empty alternate rows. So I wrote the below function to confirm if there are empty rows.

After quite a bit of struggle and search online I realized it's the issue with the datatype and decoding. Then I reached out the forums to check if someone else had similar issues and found a lot of help and guidance from previous discussions. I was able to figure out my mistakes and used python connection to SQLite database and performed below queries.

Data Overview

Below are the sizes of the data I have used. Considering my laptop configuration I choose to restrict myself with the size minimum requirement.

File sizes:
Sunnyvale.osm --------53.2 MB
sunnyvale.db ---------28.5 MB
nodes.csv ------------20.0 MB
nodes_tags.csv--------500 KB
ways.csv -------------2.0 MB
ways_tags.csv---------2.73 MB
ways_nodes.csv--------6.98 MB

Once the data is loaded into the database, I was able to run few queries to analyze the data. Below are the queries and the results.

Queries and additional ideas

query = 'select user, uid, count(*) from nodes group by uid order by count(*) desc limit 10;'

sqlquery(query)

```
[(u'andygol', 94578, 36924),
 (u'RichRico', 2219338, 31013),
 (u'karitotp', 2748195, 29270),
 (u'samely', 2512300, 24964),
 (u'n76', 318696, 21835),
 (u'dannykath', 2226712, 21037),
 (u'ediyes', 1240849, 17715),
 (u'calfarome', 2511706, 10594),
 (u'matthieun', 595221, 9457),
 (u'nikhilprabhakar', 2835928, 8185)]
```

query = 'select user, uid, count(*) from nodes group by uid order by count(*) desc limit 10;'

sqlquery(query)

```
[(u'andygol', 94578, 36924),
 (u'RichRico', 2219338, 31013),
```

```
 (u'karitotp', 2748195, 29270),
 (u'samely', 2512300, 24964),
 (u'n76', 318696, 21835),
 (u'dannykath', 2226712, 21037),
 (u'ediyes', 1240849, 17715),
 (u'calfarome', 2511706, 10594),
 (u'matthieun', 595221, 9457),
 (u'nikhilprabhakar', 2835928, 8185)]
```

query = 'select key, count(*) from nodes_tags group by key order by count(*) desc limit 10;'

sqlquery(query)

```
[(u'housenumber', 2371),
 (u'street', 2320),
 (u'city', 1516),
 (u'highway', 1458),
 (u'name', 736),
 (u'amenity', 523),
 (u'source', 354),
 (u'postcode', 325),
 (u'addr', 257),
 (u'shop', 251)]
```

query = 'select value, count(*) from nodes_tags where key = \'amenity\' group by value order by count(*) desc limit 10;'

sqlquery(query)

```
[(u'restaurant', 127),
 (u'bicycle_parking', 68),
 (u'bench', 34),
 (u'fast_food', 31),
 (u'dentist', 23),
 (u'cafe', 22),
 (u'doctors', 20),
 (u'parking', 17),
 (u'atm', 15),
 (u'post_box', 15)]
```

query = 'select value, count(*) from nodes_tags where key = \'shop\' group by value order by count(*) desc limit 10;'

sqlquery(query)

```
[(u'hairdresser', 25),
 (u'car_repair', 21),
 (u'beauty', 18),
 (u'supermarket', 15),
 (u'dry_cleaning', 11),
 (u'convenience', 10),
```

```
 (u'alcohol', 9),
 (u'laundry', 8),
 (u'clothes', 7),
 (u'jewelry', 6)]
```

query = 'select value, count(*) from nodes_tags where key = \'highway\' group by value order by count(*) desc limit 10;'

sqlquery(query)

```
[(u'stop', 504),
 (u'turning_circle', 346),
 (u'traffic_signals', 260),
 (u'crossing', 256),
 (u'bus_stop', 57),
 (u'give_way', 21),
 (u'motorway_junction', 10),
 (u'turning_loop', 3),
 (u'survey', 1)]
```

query = 'select id, count(*) from ways_nodes group by id order by count(*) desc limit 10;'

sqlquery(query)

```
[(250195022, 164),
 (38201877, 143),
 (39200808, 135),
 (49438764, 135),
 (228906414, 126),
 (49438925, 114),
 (390153611, 113),
 (390153610, 112),
 (47795407, 109),
 (474246349, 107)]
```

query = 'select id, count(*) from ways_tags group by id order by count(*) desc limit 10;'

sqlquery(query)

```
[(49046539, 21),
 (26230997, 20),
 (22372547, 19),
 (26230999, 18),
 (30931570, 18),
 (340316602, 18),
 (41914255, 17),
 (49322007, 17),
 (311341167, 17),
 (311341170, 17)]
```

query = 'select key, count(*) from ways_tags group by key order by count(*) desc limit 10;'

sqlquery(query)

```
[(u'building', 27627),
 (u'source', 6558),
 (u'highway', 5849),
 (u'street', 5341),
 (u'housenumber', 5338),
 (u'name', 3203),
 (u'maxspeed', 2614),
 (u'service', 2268),
 (u'lanes', 1964),
 (u'county', 1932)]
```

query = 'select value, count(*) from ways_tags where key = \'amenity\' group by value order by count(*) desc limit 10;'

sqlquery(query)

```
[(u'parking', 140),
 (u'restaurant', 26),
 (u'school', 26),
 (u'fuel', 23),
 (u'bank', 18),
 (u'fast_food', 17),
 (u'place_of_worship', 15),
 (u'swimming_pool', 10),
 (u'fountain', 7),
 (u'pharmacy', 5)]
```

query = 'select a.user, b.value, count(*) from nodes a join nodes_tags b on a.id = b.id \

where b.key = \'amenity\' group by b.value order by count(*) desc limit 10;'

sqlquery(query)

```
[(u'Perch338', u'restaurant', 127),
 (u'Harry Cutts', u'bicycle_parking', 68),
 (u'Minh Nguyen', u'bench', 34),
 (u'jgkamat', u'fast_food', 31),
 (u'jgkamat', u'dentist', 23),
 (u'WrErase', u'cafe', 22),
 (u'nmixter', u'doctors', 20),
 (u'Jonathan ZHAO', u'parking', 17),
 (u'matthieun', u'atm', 15),
 (u'fmarier', u'post_box', 15)]
```

query = 'select key, count(*) from nodes_tags group by key order by count(*) desc limit 20;'

sqlquery(query)

```
[(u'housenumber', 2371),
```

```
 (u'street', 2320),
 (u'city', 1516),
 (u'highway', 1458),
 (u'name', 736),
 (u'amenity', 523),
 (u'source', 354),
 (u'postcode', 325),
 (u'addr', 257),
 (u'shop', 251),
 (u'crossing', 240),
 (u'direction', 240),
 (u'country', 193),
 (u'state', 181),
 (u'phone', 171),
 (u'cuisine', 154),
 (u'created_by', 133),
 (u'surface', 130),
 (u'website', 129),
 (u'date', 128)]
```

query = 'select a.user, b.value, count(*) from nodes a join nodes_tags b on a.id = b.id \

where b.key = \'cuisine\' group by b.value order by count(*) desc limit 10;'

sqlquery(query)

```
[(u'Perch338', u'indian', 25),
 (u'mvexel', u'chinese', 15),
 (u'jgkamat', u'sandwich', 15),
 (u'joeclmbr', u'pizza', 14),
 (u'n76', u'mexican', 9),
 (u'n76', u'thai', 8),
 (u'Minh Nguyen', u'korean', 7),
 (u'Minh Nguyen', u'burger', 5),
 (u'WrErase', u'coffee_shop', 5),
 (u'Minh Nguyen', u'japanese', 5)]
```

query = 'SELECT tags.value, COUNT(*) as count FROM (SELECT * FROM nodes_tags UNION ALL SELECT * FROM ways_tags) tags \

WHERE tags.key= \'postcode\' GROUP BY tags.value ORDER BY count DESC;'

sqlquery(query)

```
[(u'94087', 250),
 (u'94086', 231),
 (u'95051', 137),
 (u'94085', 25),
 (u'94040', 18),
 (u'94086-6406', 5),
 (u'95054', 5),
 (u'94041', 3),
```

```
 (u'94807', 2),
 (u'94087-2248', 1),
 (u'94087\u200e', 1),
 (u'94088-3453', 1),
 (u'94088-3707', 1),
 (u'95050', 1),
 (u'95086', 1),
 (u'CA 94086', 1)]
```

query = 'SELECT COUNT(*) FROM nodes;'

sqlquery(query)

```
[(250586,)]
```

query = 'SELECT COUNT(*) FROM ways;'

sqlquery(query)

```
[(35156,)]
```

query = 'SELECT COUNT(DISTINCT(e.uid)) FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e'

sqlquery(query)

```
[(324,)]
```

query = 'SELECT e.user, COUNT(*) as num \

FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e \

GROUP BY e.user \

ORDER BY num DESC \

LIMIT 10;'

sqlquery(query)

```
[(u'andygol', 41907),
 (u'RichRico', 34222),
 (u'karitotp', 31478),
 (u'samely', 28559),
 (u'n76', 27198),
 (u'dannykath', 23103),
 (u'ediyes', 19664),
 (u'calfarome', 11449),
 (u'matthieun', 11309),
```

```
 (u'nikhilprabhakar', 9047)]
```

query = 'SELECT COUNT(*) FROM (SELECT e.user, COUNT(*) as num FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) e\

GROUP BY e.user HAVING num=1) u;'

sqlquery(query)

```
[(85,)]
```

query = 'SELECT nodes_tags.value, COUNT(*) as num FROM nodes_tags JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value=\'place_of_worship\') i \

ON nodes_tags.id=i.id WHERE nodes_tags.key=\'religion\' GROUP BY nodes_tags.value ORDER BY num DESC;'

sqlquery(query)

```
[(u'christian', 11), (u'unitarian_universalist', 1)]
```

Other Ideas

- Based on the information I have as I am local resident of Sunnvale, I see there are a lot of restaurants missing. If we can get a reference to all the restaurants and addresses from Yelp or other restaurant review sites and add to the database it would be more complete.
- I am not sure at this moment if Yelp provides API or data to public to analyze but I believe Yelp has a lot of information on most of the businesses around. It also has ratings for each business.
- It would be great if we can capture the ratings as well, as part of tags for the nodes.
- Ratings and business information will be valuable information for this data I believe.

Benefits

- Yelp information would add value in terms of providing one stop information for all the decision making related to restaurants and other businesses.
- Ratings are simple information on a scale of 5 so I hope there won't be more issues with data quality.
- Issues though could be if we want to add some more inference based on the reviews on yelp, it will be having too much text to be cleaned and programmatic cleaning for the reviews is tough.

Conclusion

It looks like the Sunnyvale area is incomplete, though I believe it has been well cleaned by the users before I began cleaning itself. I feel lot of information can be added to this data. I live in Sunnyvale and I observed few of the famous restaurants are missing and I was able to search for my home address and it was available in there. I tried to analyze what ever information I could imagine we can get from this data.