Map Area: Hyderabad

I am interested to find out map information of my hometown, more specifically the places, where I did my schooling, and take this opportunity to contribute to OpenStreetmap

http://overpass-api.de/api/map?bbox=78.5104,17.3358,78.5902,17.4077

I took a part of my city database, the complete city database was quite huge, so I took a part out of my city, mainly the place where I grew up

Problems Encountered in the Map

These were the following Observations:

1) There is one Problemchar key, Renuka Ellama Temple and its key is place_of_worship. It is likely that it should be reversed

2) Handling Incorrect postal codes: some places have 5 digit codes, but hyderabad city has 6 digit postal codes

3) We observe in some places users use, Inconsistent keys, postal_code and in some other places they use postcode. Objective is to be consistent across the keys denoting postcodes

4) Checking for all keys, and identifying duplicate keys

5) Identifying and Cleaning Improper/abbreviated values, Like 'Restrictio' instead of 'Restriction'

6) The last part was the most challenging, how to store characters in native language 'telugu', in Sqlite Database and csv files.

<u>1) Problematic keys</u>

In node_tags, we have a key: Renuka Ellama Temple and its value: place_of_worship. Ideally, key-value pair should be reversed

Changing it to key = place_of_worship, value = Renuka Ellamma Temple

DELETE * FROM node_tags WHERE KEY = 'Renuka Ellamma Temple';

INSERT INTO node_tags

VALUES (1504756065,

'place_of_worship',

'Renuka Ellamma Temple',

'regular');

2) Handling incorrect Postal Codes Values

```
SELECT tags.key,tags.value
FROM (SELECT * FROM node_tags UNION ALL
    SELECT * FROM way_tags) tags
WHERE tags.key LIKE '%post%'
GROUP BY tags.key,tags.value;
```

Results:

postal_code|500013

postal_code|500659

postcode|500013

postcode|50003

postcode|500035

postcode|500036

postcode|500038

postcode|500039

postcode|500044

postcode|500060

postcode|500068

postcode|500074

We notice there is a wrong postcode 50003, we need to change this to the right value,

Identifying the node_id value for this record

```
select * from node_tags where key = 'postcode' and value = 50003;
```

2445154240|postcode|50003|addr

select * from nodes where id = 2445154240;

2445154240|17.3641703|78.5244243|kpworld|1733609|2|17663299|2013-09-03T23:45:37Z

Based on the lat and long co-ordinates, the right pin code for that location turns out to be 500060

Hence, updating the values

update node_tags

set value = 500060

where id = 2445154240

and value = 50003;

3) Inconsistent Postal keys:

Use consistent keys, we observe in some places users use, postal_code and in some other places they use postcode

Maintain consistency across the key denoting postcodes

update node_tags

set key = 'postcode'

where key = 'postal_code';

update way_tags

set key = 'postcode'

where key = 'postal_code';

4) Checking for all Node_tags keys, and identifying duplicate keys

Idenitfy all keys in Node_tags, Way_tags and ensure there are no duplicate keys, we want to ensure consistency in keys. Identify all keys in Node_tags

select distinct key from node_tags;

AND_a_nosr_p, name, postcode, source, operator, platforms, railway, ref, created_by, amenity, shop, religion, power, street, highway, place, cuisine

, internet_access, barrier, local_ref, name_1, shelter, city ,housename, building, alt_name, atm, designation, leisure, housenumber ,man_made, todo ,district, tourism, website, wikipedia, historic, ele, note, phone, office, opening_hours, access, bitcoin, horse, motor_vehicle, ja, junction, smoking , crossing, place_of_worship

Checking for these key values (Objective is to determine if they are duplicate)

i) ref, local_ref: ref, local_ref convey different information

ii) name, name_1, alt_name: name_1 (nodes: 1594518972, 2412783672) is giving more detail about the name

'alt_name' (nodes: 1598149459, 2183029318): is the same with alt_name, it gives more information about the name

iii) Checking for key 'ja':

This record whose key = 'ja', it's value seems erraneous 'ワランガル', It is not local script, so have deleted that record

delete from node_tags where key = 'ja'

5) Repeating the above steps for way_tags:

Idenitfy all keys in Way_tags, Way_tags and ensure there are no duplicate keys, we want to ensure consistency in keys

select distinct key from way_tags;

highway, oneway, ref, old, name, source, lanes, junction, natural, waterway, bridge, layer, surface, leisure, sport, electrified, frequency , gauge, passenger_lines, railway, voltage, maxspeed, amenity, created_by, admin_level, boundary, AND:importance_level, AND_a_nosr_r, fixme, landuse, building, man_made, housename, postcode, street, website, note, city, office, emergency, power, access, motor_vehicle, bicycle, cycleway, housenumber, protect_class, te, phone, name_1, religion, foot, horse, incline, full_id, osm_id, osm_type, operator, wikipedia, cuisine, shop, construction, ele, levels, intermittent, water, cables, area, opening_hours, atm, fee, park_ride, supervised, barrier, service , tourism, type, trail_visibility, width, phone_1, restrictio, tunnel

i) change the key restrictio to restriction

update way_tags

set key = 'restriction'

where key = 'restrictio'

ii) Checking for keys: phone_1, phone

select * from way_tags where id = (select id from way_tags where key = 'phone_1');

phone_1 captures additional information

iii) Checking for street names

select count(*) from way_tags where key = 'street';

Surprisingly, street names are largely good, except for one, where abbreviations were used.

'Vidyanagar Railway Stn. Road'

SELECT REPLACE(value,'Stn.','Station') from way_tags where key = 'street' and value like '%Stn.%';

Update way_tags

set value = REPLACE(value,'Stn.','Station')

where key = 'street' and value like '%Stn.%';

iv)Checking what key 'te' represents

select value from way_tags where key = 'te'

this time it represents ??????? . The value is lost in encoding,

It's actual value is మూసి : "musi" in telugu, so need to store values in Unicode-8

I have posted a question on stackoverflow asking the same.

(http://stackoverflow.com/questions/37218970/sqlite-csv-unicode-encoding-error) but have not found any replies yet

Would like to know the answer.

Data Overview and Additional Ideas

File sizes

```
hyderabad_india.osm ......... 73.717 MB
final_project.db ........... 57.007 MB
nodes.csv ............... 29.518 MB
nodes_tags.csv ........ 60 KB
ways.csv ............... 5.275 MB
```

```
ways_tags.csv ......... 3.147 MB
ways_nodes.cv ......... 11.246 MB
```

Number of nodes

325191

Number of ways

79292

Number of unique users

```
select count(*)
from (select user from Nodes
union
select user from Ways
);
```

159

Top 10 contributing users

```
select users.user, count(*)
from (select user
from Nodes
UNION ALL
select user
from Ways) users
group by users.user
order by count(*) desc
limit 10;
```

AnushaPyata 69471

Bindhu 53173

Harisha 51810

Pallavi1 50354

Kranthikumar 47644

HarishVarma 17471

Shiva05 17266

Nagarjunreddy 14862

SENTHILN 9958

FlankerH 8659

Number of users appearing only once (having 1 post)

```
select count(*)
from (select users.user, count(*)
from (select user
from Nodes
UNION ALL
select user
from Ways) users
group by users.user
having count(*) = 1);
```

37

Other ideas about the datasets

1) The extracted data appears largely clean, but a consistency in keys could make analysis much easier. Tag Keys like name_1, alt_name, ele, ja, te, phone_1 were used, but users did not specify what they meant.
   We could force users to enter a description of keys that are not a part of database. This will help figuring out the meaning of the tags. Since today the keys list is exhaustive, there will

few cases where user adds new keys. And by making entering description of new keys mandatory it will be easier to make sense of data. However, potential challenge of making description mandatory is that it can turn user off from entering the key altogether. Ensuring that in suggestions potential keys props up when user enters keys, could help entry simple. Stackoverflow does this brilliantly well

2) Existing users can invite new users, where both the user sending the request and the one who receives the request get rewards. The rewards can be points, badges. This way we can have more users contributing to map data thus covering more area. One challenge can be to measure the impact of non-monetary reward points like points, badges. UBER benefitted from introducing a similar system, where both the inviter and the invitee received monetary benefits. In our case, the success of this idea depends on which reward system can be as successful/impactful as monetary rewards. Will just points, badges as rewards work well, is the challenge
3) If Users can receive notifications about new data entries based on their location we can request them to verify this new data. Again introducing a reward system, where we incentivize user with accurate data and a penalise user with incorrect entries, can be a good way to ensure clean data entries. This will motivate users to enter correct values. One potential challenge could be if everyone starts to modify the new data as they wish, then it leads to unreliable data. Making sure that there are moderators or only users with a certain number of points be eligible for making changes could be one way. Again stackoverflow does this well
4) Simplifying the process of data entry by taking a picture of the image and then all the relevant features be extracted from the image could be a long term vision of openstreet map. If things are implemented correctly, it can really simplify the whole process of entering the data. Imagine if a user takes a photo, the relevant features like name of the entity, address, phone number if available are extracted from the image. This basic information coupled with GPS co-ordinates can be used to search for the exact object in the internet. If exact matches are found then it can automatically fill in other tag keys. The major challenge is it is very difficult to take a extract necessary information like Name, Address, phone no. from a normal image as images have lot of noise and it is difficult to extract useful information from the noise. But for now, keeping a database of images along with user entered data could come along way in the future. We can use it like tagging images with user defined key-values. Atleast, This could lead to huge collection of Image database and can collaborate with google Images for feature extraction
5) Top User (AnushaPyata) has contributed 17.31%

   Top 5 users, have contributed 67.35%,

   Top 10 users have contributed 84.22%

   As the contributions have been fairly even from the top 5 contributors, it appears they have may have been inspired by gamification strategies, while the rest may not be motivated as the gap between top 5 and the rest is huge.

Additional Data Exploration

Top 10 appearing amenities

```
select T.value, count(*)
from (
select value from node_tags where key = 'amenity'
Union all
select value from way_tags where key = 'amenity'
) T
group by T.value
order by count(*) desc
limit 10;
```

```
place_of_worship,38
fuel,33
bank,25
atm,22
parking,16
restaurant,14
cinema,12
hospital,12
college,11
school,10
```

Biggest religion

```
select T.value, count(*)
from (
select value from node_tags where key = 'religion'
Union all
```

```
select value from way_tags where key = 'religion'
) T
group by T.value
order by count(*) desc
limit 3;
```

hindu,32

christian,2

muslim,2

Most popular cuisines

```
select T.value, count(*)
from (
select value from node_tags where key = 'cuisine'
Union all
select value from way_tags where key = 'cuisine'
) T
group by T.value
order by count(*) desc
limit 5;
```

regional,4

chicken,1

Data sample is too less to infer much

Conclusion

Due to the limitations of my laptop's processor, I had to take a relatively small area of Hyderabad,

The database appears to be fairly clean, much improvement could be expected in the number of keys, and eliminating duplicate keys

Was expecting some users to enter values Local script, but mostly English was used.