OpenStreetMap Data Case Study

Map Area

İzmir, Turkey

- https://mapzen.com/data/metro-extracts/metro/izmir_turkey/

This map is of my hometown. therefore I’m interested in this map.

Problems Encountered in the Map

I noticed some main problems with the data, which I will discuss in the following order:

- Street address inconsistencies
  “İzkent” : “Izkent”
  “İzkent,” : “Izkent”
  “İzmir” : “Izmir”
  “izmir” : “Izmir”
- Abbreviations
  “sk.” : “Street”
  “Sk.” : “Street”
  “Sk” : “Street”

“sk” : “Street”

“Sok.” : “Street”

“Cd” : “Avenue”

“Cd,”: “Avenue”

“cd” : “Avenue”

“Cd.” : “Avenue”

“cd.” : “Avenue”

“Cad.” : “Avenue”

“Cad” : “Avenue”

“Blv.” : “Boulevard”

“Bulv.”: “Boulevard”

“Mh” : “Neighborhood”

“mh”: “Neighborhood”

“Mah.”: “Neighborhood”

“Mh.,” : “Neighborhood”

“Mah,” : “Neighborhood”

- Misspelling

“Kâhya” : “Kahya”

“Şirinyer” : “Sirinyer”

- Turkish names

“Sokak” : “Street”

“sokak” : “Street”

“Sokağı” : “Street”

“Sokak,” : “Street”

“Caddesi” : “Avenue”

“caddesi” : “Avenue”

“Havalimanı” : “Airport”

“Havalımanı” : “Airport”

“havalimanı” : “Airport”

“Liman” : “Port”

“liman”: “Port”

“Bulvar” : “Boulevard”

“Bulvari” : “Boulevard”

“bulvarı”: “Boulevard”

“Bulvarı”: “Boulevard”

“mahallesi”: “Neighborhood”

“Mahallesi”: “Neighborhood”

“Yerleşkesi” : “Campus”

“İzkent” : “Izkent”

“İzkent,” : “Izkent”

“Meydanı” : “Square”

“Meydan” : “Square”

“Şirinkapı” : “Sirinkapi”

“İstikbal” : “Istikbal”

“Gaziosmanpaşa” : “Gaziosmanpasa”

“sahil” : “Coast”

“NilüferSokak” : “Nilufer Street”

“Alışveriş Merkezi”: “Shopping Center”

“Paşa” : “Pasa”

“Şehitleri” : “Sehitleri”

“Çevre Yolu” : “Highway”

“Üniversite” : “University”

Above is the old name corrected with the better name. Using clean_data.py, I updated the names.

File sizes:

```
osm/izmir_turkey.osm:               50,6 MB
nodes_csv:                          19,2 MB
nodes_tags.csv:                     414 KB
ways_csv:                           2,2 MB
ways_nodes.csv:                     7,1 MB
ways_tags.csv:                      2,2 MB
izmir.db:                           26,7 MB
```

Number of nodes

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

```
Number of nodes:  ['232635']
```

Number of ways

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

```
Number of ways:  ['36274']
```

Number of unique users:

```
SELECT COUNT(DISTINCT(a.uid))
```

```
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) a;
```

```
Number of unique users:  458
```

Top 10 contributing users

```
SELECT a.user, COUNT(*) as num
FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) a
GROUP BY a.user
ORDER BY COUNT(*) DESC
LIMIT 10;
```

```
kshuseyin           41551
VinothS             30555
vignesh anand       30106
erkinalp            18318
hari t              18303
katpatuka           17519
sheik farid         14682
surendran1          7733
niranjana           6621
Nesim               6316
```

Number of users appearing only once (having 1 post)

```
SELECT COUNT(*)
FROM
    (SELECT a.user, COUNT(*) as num
     FROM (SELECT user FROM nodes UNION ALL SELECT user FROM ways) a
     GROUP BY a.user
     HAVING num=1)  b;
```

```
Number of users appearing only once:  ['114']
```

First contribition date

```
 SELECT timestamp FROM Nodes UNION SELECT timestamp From Ways
        ORDER BY timestamp desc
        LIMIT 1;
```

```
First contribution :  ['2007-03-27T16:06:35Z']
```

Additional Ideas

Additional Data Exploration

Top 10 appearing amenities

```
SELECT value, COUNT(*) as num
FROM nodes_tags
WHERE key='amenity'
GROUP BY value
ORDER BY num DESC
LIMIT 10;
```

```
Pharmacy            116
Bank                79
Restaurant          76
Atm                 64
School              50
Fuel                48
Cafe                44
Bus_station         33
Place_Of_Worship    30
Taxi                22
```

Biggest religion (no surprise here)

```
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
LIMIT 1;
```

```
Biggest religion :  ['Muslim 26']
```

Most popular cuisines

```
SELECT nodes_tags.value, COUNT(*) as num
FROM nodes_tags
    JOIN (SELECT DISTINCT(id) FROM nodes_tags WHERE value
='Restaurant') i
    ON nodes_tags.id=i.id
WHERE nodes_tags.key='cuisine'
GROUP BY nodes_tags.value
ORDER BY num DESC;
```

```
Turkish                                 6
Regional                                2
Seafood                                 2
Barbecue;Sausage;Chicken;Tea;Grill      1
Dessert                                 1
Fish;Steak_House;Grill;Kebab;Seafood    1
Kebab;Turkish                           1
```

```
Pizza,_Pasta,_Insalata                  1
Steak_House;Grill                       1
```

Most popular bank

```
SELECT nodes_tags.value, COUNT(*) as num
        FROM nodes_tags
            JOIN (SELECT DISTINCT(id) FROM nodes_tags WHE
RE value like '%Bank%') i
            ON nodes_tags.id=i.id
        WHERE nodes_tags.key='name'
        GROUP BY nodes_tags.value
        ORDER BY num DESC
        LIMIT 10;
```

```
İş Bank                 16
Akbank                  17
Finansbank              8
Vakıfbank               8
Garanti Bank            6
Ziraat Bank             6
Denizbank               4
Halkbank                4
TEB                     4
İngbank                 4
```

Conclusion

The İzmir OpenStreetMap dataset is a big enough and quite quite and this dataset has many typo errors which caused by human. Considering there’re many of contributors for this map, there is a great numbers of human errors in this project. The dataset contains very old information which is now incomparable to that of Google Maps. We clean this data set ,but We could not completely clean the data. But it was sufficiently cleaned for this project.

The following can be done to Control typo errors:

- We should make some rules to input data ,that users input. This rules may be restrict users input in their native language.
- We should create a more robust script to clean the data regularly on a bi-weekly basis.
- The text entered in the user’s native language can be translated into English.
- Competitions can be organized to encourage users who make the most contributions and at least misspellings. Every month, the top 10 users can be encouraged by showing them in social media. Forexample Certain gamification elements such as rewards, badges, or a leaderboard.
- People living in the city are searching for places that are popular in the city. We can increase the number of users by adding information about tourism regions. So more people can correct the map.

İzmir area is incomplete. The dataset contains very less amount of additional information such as amenities, religion, cuisines , bank, popular places and other useful information. but Via SQL query, I learned a few new things about my hometown.

Files

- `README.md` : this file
- `OpenStreetMapReport.pdf` : pdf format of md file
- `nodes.csv` : nodes csv
- `nodes_tags.csv` : nodes tags csv
- `ways.csv` : ways csv
- `ways_tags.csv` : ways tags csv
- `ways_nodes.csv` : ways nodes csv
- `izmir.db` : Database
- `izmir_sample.osm` : sample data of the OSM file
- `clean_data.py` : Incorrect street, city name find and update their names
- `create_csv.py` : build CSV files from OSM
- `create_database.py` : create database of the CSV files
- `query_executer.py` : execute given sql file
- `sql/select_way_size.sql` : query of Number of ways
- `sql/select_nodes_size.sql` : query of Number of nodes
- `sql/unique_users.sql` : query of Number of unique users
- `sql/contributing_user.sql` : query of Top 10 contributing users
- `sql/users_appearing_only_once.sql` : query of Number of

users appearing only once

- `sql/biggest_religion.sql` : query of Biggest religion
- `sql/popular_cuisines.sql` : query of Most popular cuisines
- `sql/first_contribution.sql` : query of First contribution date
- `sql/popular_bank.sql` : query of Most popular bank
- `sql/amenity.sql` : query of Most popular amenity

Note

This application builds with Python 3.6