About Wrocław

Wrocław (Wroclaw) is a city in Poland where I live. It is one of the fastest growing cities in Poland economically as well as in the size aspect: city authorities are increasing expenses on the infrastructure annually and therefore Wrocław have a lot of renovation and modernization sites. This concern also streets - as the Wroclaw citizen I have to be up-to-date with the information on the streets currently being renovated and possible traffics that could be met on the way to school or to work.

My analysis

In [5]:

```
print_size("data")
```

```
              File          Size(MB)
         ways.csv          2.283533
          test.db         34.798592
    nodes_tags.csv           3.59219
        sample.osm          0.679161
         nodes.csv         20.127293
street_mapping.csv          0.013742
       wroclaw.osm         66.331048
     ways_tags.csv          4.033549
    ways_nodes.csv          8.379125
```

NOTE: Source database ("wroclaw.osm") has more than 66 MB.

Problems encountered and solution thereto:

1. In the Polish OSM we have another tag attribute ("SYM_UL"), a unique number assigned to the each street in the particular. I have found that in my OSM there are three streets without such information, so I filled this missing data and create second table mapping street names and SYM_UL codes.
2. Due to the problem with importing data from csv files (nodes.csv, ways.csv), I have created an own importer based on the sqlite3 python library.
3. As per consistency between Python schema and SQL schema I have pointed "version" tag as the integer in Python schema.

Questions asked:

1. As the open source fan, I wonder how many users participate in the creation of the map.
2. Due to the application OSM as the GPS backend I would like to see whether it is not outdated.
3. Due to having kids and considering moving from Wroclaw - how many kindergartens are in Wrocław.

Scripts used:

scripts	What does it?
version_checker.py	
printsize.py	Printing size of the data files
tablecreator.py	Creation of table via sqlite3 library
schematest.py	It is the schema file from the course - updated with "version" input - it is now an integer
samplecreation.py	It is the file from the course
streetmapping.py	My answer on the issue with sym_ul
osmcsv.py	It is the file from the course
basecreator.py	Creating the database
csv_to_sql.py	Creating the tables in the database

Problems encountered in your map

SYM_UL for every street

We could see that most frequent tags in the nodes_tags table are following:

In [ ]:

```
%%sql
SELECT key, COUNT (key)
  FROM  nodes_tags
 GROUP BY key
 ORDER by count(key)
  DESC LIMIT 10;
```

From the above we could see that there are 4765 street tags, however "sym_ul" have 4266 positions. What is interesting in the Polish section there is a unique key added to each street ("street: sym_ul"). As you are probably an English speaking reader you may google translate this thread: https://forum.openstreetmap.org/viewtopic.php?id=59111 (https://forum.openstreetmap.org/viewtopic.php?id=59111) or here https://wiki.openstreetmap.org/wiki/Talk:Pl:Importy/Adresy (https://wiki.openstreetmap.org/wiki/Talk:Pl:Importy/Adresy). Unfortunately this tag is not documented in English. Sym_ul is the id added by users on basis of the registry by the Polish National Statistical Office. Let's check how many unique streets we have in the nodes_tags table and how many unique "sym_ul" tags we have therein too.

In [10]:

```
%%sql
SELECT COUNT(*) as "Total number fo streets"
FROM
    (SELECT value
       FROM nodes_tags
      WHERE key = "street"
      UNION SELECT value
       FROM ways_tags
      WHERE key = "street"
      GROUP BY value);
```

Done.

Out[10]:

Total number fo streets
292

In [11]:

```
%%sql
SELECT COUNT(*) AS "Total number of sym_ul tags"
 FROM
    (SELECT value
       FROM nodes_tags
      WHERE key = "street:sym_ul"
      UNION SELECT value
       FROM ways_tags
      WHERE key = "street:sym_ul"
      GROUP BY value);
```

Done.

Out[11]:

Total number of sym_ul tags
290

Then - after making investigation of the entire dataset (so also the ways_tags) with the streetmapping file we could find that the following streets have got no "sym_ul" tags:

1. Grabarska
2. Adama Mickiewicza
3. Na Niskich Łąkach

I have changed it basing on the official base https://goo.gl/qU3mXZ (https://goo.gl/qU3mXZ). I have also created the table which enable the database user to assign street codes with the street name ("`streetmapping.csv`").

Data importer based on the sqlite3 python library - "`csv_to_sql.py`"

After importing data to csv I have found some issues:

```
nodes.csv:253769: INSERT failed: UNIQUE constraint failed: nodes.id
```

I have checked it manually while uploading each node separately from node.csv into the sqlite3. Please find below an exemplary chunk of the code:

After applying the script "`csv_to_sql.py`" I have not encountered any errors. **In the day-to-day work as the data scientist / analyst when he connects from one source to another (in our case from csv to sql) some errors can occur that could be omitted by another API.**

Here we have omitted this error through applying the Python script what spare our time

"version" tag as the integer - `versionchecker.py`

From the original schema we have:

Name	Type of data SQL	Type of data python
id	integer	int [primary key]
lat	REAL	float
lon	REAL	float
user	TEXT	int
uid	INTEGER	int
version	INTEGER	int
changeset	INTEGER	int
timestamp	TEXT	string

In the file schema we have however:

```
'version': {'required': True, 'type': 'string'},
```

What is inconsistent with the integer type set in the SQL schema. Thus, Python is able to convert it to the integer so I have updated schema with:

```
'version': {'required': True, 'type': 'integer'},
```

Overview of the Data

Counting nodes, ways and users

In [ ]:

```
SELECT COUNT(*) AS "NO of Nodes" FROM (SELECT DISTINCT id FROM nodes);
SELECT COUNT(*) AS "NO of Ways" FROM (SELECT DISTINCT id FROM ways);
```

Number of ways is 39294. Number of nodes is 253769.

In [111]:

```
CREATE VIEW users AS
SELECT user AS user
FROM nodes
UNION ALL
SELECT user as user FROM ways;

SELECT DISTINCT user, COUNT(*) FROM users
 GROUP BY user
 ORDER BY count(*)
 DESC LIMIT 3;
```

```
Done.
Done.
```

Out[111]:

user	COUNT(*)
rowers2	112861
lms	74278
maraf24	70277

There are three guys that created more than 250000 elements! Three guys from **368** people who create the map (below see the SQL query).

In [ ]:

```
SELECT count(user) as "No. of users" FROM (SELECT DISTINCT * FROM users);
```

It is 88% ((112861 + 74278 + 70277)/293063) of the elements! The dataset should be verified due to the fact that only three users created almost 88% of the map.

It could influence potentially negatively on the quality of the map - when the map is not verified by more users it could cause error to GPS system using OSM.

Counting number of kindergartens / schools

As I have mentioned due to the having a kid, it is important to assure him a proper education and the best opportunities to learn as well as to play. Let's check types of kindergartens and schools in the neighbourhood.

In [ ]:

```
SELECT COUNT(*)
  FROM nodes_tags
 WHERE key = "amenity"
   AND value LIKE "kindergarten"
    OR value LIKE "school"
    OR value LIKE "playground";
```

This query shows that only 23 positions fulfills query criteria, what is however too small as for Wrocław. According to this site (https://www.wroclaw.pl/files/edu_szkoly/33Przedszkola.xlsx), the total number of public kindergartens is nearly **100**. Similarly, number of schools amounts to almost **100** (check here (https://www.wroclaw.pl/files/edu_szkoly/33Szkoly%20Podstawowe.xlsx)).

Result: The dataset is not reliable in the searching such objects like kindergartens or schools. It cannot be also excluded that other types of locations could be different from the reality.

Time of actualization

Every map used in GPS should be up-to-date. Let's check how many times the users intervene and update the way tags as regards as Wrocław:

In [ ]:

```
SELECT strftime("%Y", timestamp) AS Year, COUNT(*) AS "Number of timestamps"
 FROM ways
GROUP BY strftime("%Y",timestamp);
```

Between 2008 and 2013 there were not many interventions made by the users, however from 2014 the base seems to be updated. The threshold is at 2017. You could see it in the file named graph.jpg in the data folder. Most of the changes have been made in 2017. However the tendency is broken in January 2018 what could put in the question the accuracy and validity of data.

Conclusion: You need to be careful using the OSM of Wroclaw!

Other ideas about the dataset

To sum up the following analysis:

1. The map was created by three users in almost 88%. It puts the quality of the map **in the question**, assuming that the higher number of users participating in creation of the map, the higher level of verification of the information contained therein. **Therefore**, improvement requires an engagement of the other open source volunteers who will verify and change the map, with the benefit for the passive users - for instance, the map will be a good a reliable source for the parent as the information about the number of schools or kindergartens. Other locations should be also always verified. **However**, it could be very hard to monitor the state of locations, ways and nodes in Wrocław in real time. The gathering much larger number of users could also be difficult to achieve due to the organizational problems (it’s a so called human factor).
2. What has been analyzed through queries above - "SYM_UL" tag could also cause some discrepancies for non-Polish GPS Systems. However, as this tag is congruent with other Polish systems of information provided by authorities, the benefits of using it for Polish users could be great. But open source community should prepare some uniform standard about the "SYM_UL" tag in order to eliminate possible system integration problems between Polish OSM and other GPS System.
3. The problem with eliminating discrepancies between real state of ways / locations and Wrocław and OSM map is that you cannot make it on your own without help of others. However, some verification level you could achieve through checking the OSM content with Google Maps. This requires creation of relevant scripts enabling you to automate scraping information about locations and ways from Google Maps and verification of this information with the data from OSM.