OpenStreetMap Data Case Study

Map Area

neal by Ellicott City, Howard County, Maryland, United States of America

- https://www.openstreetmap.org/relation/133607 (https://www.openstreetmap.org/relation/133607)

In 2008, I lived in the city, so I want to analyized the city data. I acutally grap the city.. but I think the my data grap the region more than Ellicott City like Howard County and so....

Problems Encountered in the Map

After I get the file data of Ellicoot City, I put in the file to my previous data.py files to make CSV files to use in sql and find some problems to use this file.

- street names problem
  - there are some street names which didn't audited using data.py
- zipcode problem
  - the data has some different name like(postcode, postal_code)
  - there are some diffenrent value(21244)
- overlap key name problem
  - there are some different key name have same attributes (county_name = county, county_num = county_id) I check several times about this key.

street names problem

To standarize street names, 1. I have to know what kind of street names the data has, and 2. update data.py to adjust it.

using below code, I got some street names in this file.

In [ ]:
```
OSM_PATH = "/home/shlee/Desktop/Programming/ellicott city.xml"

def get_street_name(files):
    data = []
    context = ET.iterparse(files, events=('start', 'end'))
    _, root = next(context)
    for event, elem in context:
        if elem.tag == 'way':
            for a in elem.iter("tag"):
                if a.attrib['k'] == 'tiger:name_type':
                    data.append(a.attrib['v'])
    return set(data)

data = set(['Pike', 'Walk', 'Run', 'Trl', 'Ave:Rd', 'Ln', 'Pl:Rd', 'Path', 'Pky', 'St', 'Rd'
, 'Cir', 'Dr:Rd', 'Way', 'Aly', 'Ave', 'Ter', 'Row', 'Dr', 'Pl', 'Ct', 'Blvd', 'Ave:Dr', 'Ci
r:Ct', 'Ave; Rd', 'Dr:St', 'Dr:Ln:Rd', 'Dr:Ln'])
```

using the data, I insert below code to data.py to make aduited csv file.

In [ ]:
```
def update(name, mapping):
    try:
        name = mapping[name]
        return name
    except:
        return name
```

zipcode problem

there are two column postcode, postal_code, I have to merge it.

In [ ]:
```
select key, count(*)
from ways_tags
where key LIKE 'post%'
group by key
order by count(*) desc;

   key           value
"postcode"     "24010"
"postal_code"  "176"
"postal"          "7"
```

I decide to update 'postal' and 'postal_code' to 'postcode'

In [ ]:
```
update ways_tags
set key = "postcode"
where key = "postal_code" or key = "postal";
```

check the result

"postcode" "24193"

next, I check wrong postcode

In [ ]:
```
select key, value , count(*)
from ways_tags
group by value
having key = "postcode" ;

"postcode"      "21207" "4934"
"postcode"      "21229" "2370"
"postcode"      "21043" "2163"
"postcode"      "21244" "911"
"postcode"      "21042" "830"
"postcode"      "21227" "718"
"postcode"      "21163" "204"
"postcode"      "21045" "116"
"postcode"      "21075" "105"
"postcode"      "21044" "74"
"postcode"      "21104" "69"
"postcode"      "21029" "48"
"postcode"      "21794" "14"
"postcode"      "21216" "2"
"postcode"      "21784" "2"
"postcode"      "21042-6298"    "1"
"postcode"      "21092" "1"
"postcode"      "21212" "1"
"postcode"      "21214" "1"
"postcode"      "21215" "1"
```

I doubt the postcode which count is under 20

In [ ]:
```
"postcode"      "21794" "14"            -> right
"postcode"      "21216" "2"             -> 21207
"postcode"      "21784" "2"             -> right
"postcode"      "21042-6298"    "1"     -> 21042
"postcode"      "21092" "1"             -> 21029
"postcode"      "21212" "1"             -> 21207
"postcode"      "21214" "1"             -> 21207
"postcode"      "21215" "1"             -> 21207
```

i update the things using update fuction.

overlapped keys

some way's key are County, county, county_name // county_id, county_num in same meaning

In [ ]:
```
update ways_tags
set key = "county"
where key = 'County' or key = 'county_name';

update ways_tags
set key = "county_id"
where  key = 'county_num';

using this code, aduited the data
```

check overlapped city name

In [ ]:
```
select distinct value
from nodes_tags
where key = 'city'
order by value ;

"Baltimore"
"Carroll"
"Catonsville"
"Clarksville"
"Columbia"
"Elkridge"
"Ellicott City"
"Gwynn Oak"            -> one is wrong
"Gywnn Oak"            ->
"Halethorpe"
"Highlandtown"
"Marriottsville"
"Oella"
"Windsor Mill"
"Woodstock"
```

using update, aduit all. Also, i check ways_tags too.

Data Overview and Additional Ideas

This section contains basic statistics about the dataset, the sqlite used to gather them, and some additional ideas about the data in context.

File sizes

- ellicoot_city.osm ..... 140.7 MB
- ellicott.db ..... 100.7 MB
- nodes.csv ..... 52.7 MB
- nodes_tags.csv ..... 3.2 MB
- ways.csv ..... 3.9 MB
- ways_tags.csv ..... 9.7 MB
- ways_nodes.cv ..... 16.6 MB

Number of nodes

In [ ]:
```
SELECT COUNT(*) FROM nodes;

622975
```

Number of ways

In [ ]:
```
SELECT COUNT(*) FROM ways;

64332
```

Number of unique users

In [ ]:
```
sqlite> SELECT COUNT(DISTINCT(e.uid))
FROM (SELECT uid FROM nodes UNION ALL SELECT uid FROM ways) e;

214
```

Top 10 contributing users

In [ ]:
```
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;

"asciiphil"     "257122"
"EP_Import"     "256060"
"Sarr_Cat"      "67397"
"mpetroff-imports"      "35430"
"RoadGeek_MD99" "33599"
"ElliottPlack"  "18640"
"AdamJPaul"     "3705"
"mdroads"       "2059"
"kriscarle"     "1936"
"aude"  "1037"
```

Number of users appearing only once (having 1 post)

In [ ]:
```
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;

53
```

Number of city and county

In [ ]:
```
select count( distinct value )
from
(
select distinct value
from nodes_tags
where key = 'city' or key = 'county'

union all
select distinct value
from ways_tags
where key = 'city' or key = 'county'
)
order by value desc

18
```

Top 10 appearing amenities

In [ ]:
```
"restaurant"    "92"
"school"        "46"
"emergency_phone"       "32"
"place_of_worship"      "31"
"bench" "27"
"fast_food"     "26"
"cafe"  "23"
"charging_station"      "18"
"bank"  "14"
"shelter"       "13"
```

this is really interesting for me cuz when i was there, i didn't find restorant to eat nice food so i had to drive to goldencorral for 30minutes.

also, there are 4 clinics which is really few! i remember that my cousin broke his hand while shavelling snow, and i wanted to find hospital and that's horrible (because i don't get hospital which his insurance is accepted)

addition: find top user's main area

asciiphil make about 37% of this area... so i want to check this guy's main area

In [ ]:
```
select c.key, c.value
from
(select nodes_tags.key, nodes_tags.value
from nodes join nodes_tags on nodes.id = nodes_tags.id
where nodes.user = 'asciiphil'
union all
select ways_tags.key, ways_tags.value
from ways join ways_tags on ways.id = ways_tags.id
where ways.user = 'asciiphil' ) c
where c.key = 'city'

"city"  "Arbutus"
"city"  "Arbutus"
```

surprisly, this guy have just 2 city tag.

In [ ]:
```
select c.value, count(c.value)
from
(select nodes_tags.key, nodes_tags.value
from nodes join nodes_tags on nodes.id = nodes_tags.id
where nodes.user = 'asciiphil'
union all
select ways_tags.key, ways_tags.value
from ways join ways_tags on ways.id = ways_tags.id
where ways.user = 'asciiphil' ) c
where c.key = 'county'
group by c.value

"Anne Arundel, MD"      "8"
"Baltimore, MD" "1998"
"Howard, MD"    "3186"
```

this guy insert Baltimore and howard county data. i think his playground is howard and baltimore.

conclusion

I think this area's map is almost finished. But the area need bus station(which is really really terrible) information too! people who are making this map usually use different street, county, city name. so to be more precise map, openstreetmap can set a module to standardize address (like amazon address module).