Data Wrangling with MongoDB

OpenStreetMap Sample Project

Author: Mario Bonilla

1 INTRODUCTION

The map was downloaded from https://mapzen.com/data/metro-extracts, with the filter "Maryland". The area was already preset (in blue in the website) so it contained the Washington D.C.- Baltimore Metro Area plus some surrounding area. More in detail, the map comprises the District of Columbia (Washington D.C.) and the Baltimore corridor. It includes as well the upper part of the Chesapeake Bay (Annapolis for example) and part of Virginia State (Vienna...etc.). Besides, it also includes a big part of the Maryland State, up to Frederick, in the east.

2 PROBLEMS ENCOUNTERED IN THE MAP

We run “01 - Audit.py” (Streets) and “02 - Audit.py” (Zip Codes) code against the map file looking for specific problems with Street names and Zip Codes formats:

- **Street names with different types of abbreviations** (Rd. -> Road, St -> Street…etc.). This names will be cleaned up to the proper full denomination. We run “01 - Audit.py” and found the following:

Abbreviations found= "Ave": "Avenue", "Blvd": "Boulevard", "Blvd.": "Boulevard", "Cir": "Circle", "Ct": "Court", "Ct.": "Court", "Dr": "Drive", "Hwy.": "Highway", "Ln": "Lane", "Ln.": "Lane, "Pkwy": "Parkway", "Pkwy.": "Parkway", "Pl": "Plaza", "Rd": "Road", "Rd.": "Road", "St": "Street", "St.": "Street", "Ter": "Terrace".

These abbreviations were cleaned up before loading the data in MongoDB.

- **Zip Codes not in the appropriate format:** they included two or more numbers separated by “;” or “:”, sometimes an “space” character. We ran “02 - Audit.py” and found the following (codes found: number of occurences):

{'20705; 20740': 1, '20776:21037': 4, '20817; 20816': 1, '20876; 20874': 1, '21043:21228': 1, '21082:21234': 8, '21093:21153': 1, '21117; 21136; 21117; 21117': 2, '21202:21230': 2, '21202:21231': 2, '21202; 21213': 2, '21208:21209': 2, '21212:21239': 4, '21221:21224': 1, '21227:21229': 2, '21239:21286': 1, '22003:22204': 4}

This Zip Codes were cleaned up to the proper full denomination. In this case we decided to convert them to 5 digits only, keeping the first 5 digits only.

3 OVERVIEW OF THE DATA

The file took uncompressed it took 2.51 GB. Due to the huge size of the file, I had to run some python code to obtain a sample of the original file; the sample weighs 260 MB, therefore meeting the project requisite of a file bigger than 50 MB.

File size:	
dc-baltimore_maryland.osm	2.51 GB (original, too big for processing)
dc-baltimore_maryland_sample.osm	266.742 KB (sample from original, working data)

The original data was sampled with the following code (attached, Code Smaller Sample OSM.py), writing every 10th top level element (Udacity).

We cleaned up the Street names and converted the map *.osm file to JSON format running the code in the attached "02 - Clean Streets and Zip Codes and Export to JSON.py" file.

Instead of using Python, we loaded the JSON file into MongoDB running the following code in a terminal:

```
C:\Windows\system32>cd c:\mongodb\bin

c:\mongodb\bin>mongoimport --db project --collection map --type json --file c:\dc.json
2015-12-16T09:03:25.707+0100    connected to: localhost
2015-12-16T09:03:28.704+0100    [###.....................] project.map  40.6 MB/247.7 MB (16.4%)
2015-12-16T09:03:31.704+0100    [#######.................] project.map  81.3 MB/247.7 MB (32.8%)
2015-12-16T09:03:34.704+0100    [###########.............] project.map  122.9 MB/247.7 MB (49.6%)
2015-12-16T09:03:37.704+0100    [###############........] project.map  165.6 MB/247.7 MB (66.9%)
2015-12-16T09:03:40.704+0100    [###################....] project.map  211.5 MB/247.7 MB (85.4%)
2015-12-16T09:03:42.671+0100    imported 1334418 documents
```

**for the import we renamed the file from "dc-baltimore_maryland_sample.osm.json" to "dc.json" for practical reasons.*

After successfully importing the data into MongoDB, we run some queries to get some general information about the DB:

Number of Documents:

```
> db.map.find().count()
```
1334418

Number of Nodes:

```
> db.map.find({"type":"node"}).count()
```
1198740

Number of Ways:

```
> db.map.find({"type":"way"}).count()
```
134993

Number of Unique Contributors:

```
> db.map.distinct("created.user").length
```
2077

4 ADDITIONAL DATA EXPLORATION WITH MONGODB

Let us find some insights about the data loaded in Mongo DB.

More in detail, we will explore which user are the main contributors, which cities are the most named in our database, the principal places for leisure and thee top amenities.

Moreover, we will go into more detail about the amenities called restaurants, and we will study some curious feature about the Italian dish pizza, and how some people denominate it fast food or cuisine and differentiate it from Italian restaurants.

We will detail the places of worship, break them down by religion and find among the Christian religion places of worship which are the most common and where to find them.

Top 5 Contributors (names):

```
> db.map.aggregate([{"$group": {"_id": "$created.user", "count":{"$sum":1}}},
{"$sort":{"count":-1}}, {"$limit":5}])
```

{ "_id" : "EP_Import", "count" : 265103 }
{ "_id" : "mpetroff-imports", "count" : 209296 }
{ "_id" : "asciiphil", "count" : 146741 }
{ "_id" : "woodpeck_fixbot", "count" : 128787 }
{ "_id" : "aude", "count" : 89313 }

Top 5 Cities Named:

Baltimore is by far the most named city, and that it is understandable because it is the biggest city in Maryland.

```
> db.map.aggregate([{"$match":{"address.city": {"$exists":1}}}, {"$group":
{"_id":"$address.city", "count":{"$sum":1}}}, {"$sort":{"count":-1}},
{$limit:5}])
```

{ "_id" : "Baltimore", "count" : 22080 }
{ "_id" : "Dundalk", "count" : 2125 }
{ "_id" : "Parkville", "count" : 2089 }
{ "_id" : "Catonsville", "count" : 1736 }
{ "_id" : "Washington", "count" : 1705 }

Top 10 Places for Leisure:

```
> db.map.aggregate([{"$match" : {"leisure" : {"$exists" : 1}}}, {"$group" :
{"_id" : "$leisure", "count" : {"$sum" : 1}}}, {"$sort" : {"count" : -1}},
{"$limit" : 10}])
```

{ "_id" : "pitch", "count" : 804 }
{ "_id" : "park", "count" : 372 }
{ "_id" : "swimming_pool", "count" : 216 }
{ "_id" : "playground", "count" : 194 }
{ "_id" : "sports_centre", "count" : 28 }
{ "_id" : "garden", "count" : 26 }
{ "_id" : "marina", "count" : 24 }
{ "_id" : "golf_pin", "count" : 22 }
{ "_id" : "track", "count" : 19 }
{ "_id" : "recreation_ground", "count" : 18 }

Top 5 Amenities:

We have here a list of the 5 of the most cited amenities, where "parking" is the most cited.

```
> db.map.aggregate([{"$match":{"amenity": {"$exists":1}}}, {"$group":
{"_id":"$amenity", "count":{"$sum":1}}}, {"$sort":{"count":-1}}, {$limit:5}])
```

{ "_id" : "parking", "count" : 1961 }
{ "_id" : "school", "count" : 431 }
{ "_id" : "place_of_worship", "count" : 431 }
{ "_id" : "restaurant", "count" : 343 }
{ "_id" : "fast_food", "count" : 172 }

Some detailed study into some of these amenities: restaurants and fast food.

Top 5 Types of Cuisine in Restaurants:

Although pizza wins, all five are very close in number of times cites.

```
> db.map.aggregate([{"$match":{"amenity": {"$exists":1},
"amenity":"restaurant", "cuisine": {"$exists":1}}}, {"$group":
{"_id":"$cuisine", "count":{"$sum":1}}}, {"$sort":{"count":-1}}, {$limit:5}]
```

{ "_id" : "pizza", "count" : 19 }
{ "_id" : "american", "count" : 18 }
{ "_id" : "italian", "count" : 17 }
{ "_id" : "thai", "count" : 16 }
{ "_id" : "seafood", "count" : 15 }

We have now this question: do some people included a pizzeria into pizza or into Italian food? Or into fast food? (as we will see next…)

Top 5 Fast Food Types:

```
> db.map.aggregate([{"$match":{"amenity": {"$exists":1},
"amenity":"fast_food", "cuisine": {"$exists":1}}}, {"$group":
{"_id":"$cuisine", "count":{"$sum":1}}}, {"$sort":{"count":-1}}, {$limit:5}])
```

{ "_id" : "burger", "count" : 35 }
{ "_id" : "sandwich", "count" : 22 }
{ "_id" : "pizza", "count" : 10 }
{ "_id" : "chicken", "count" : 9 }
{ "_id" : "mexican", "count" : 7 }

It looks like that people categorized pizza also as fast food….

Top 10 Places of Worship:

The Christian places of worship are the vast majority of the cited. We set the limit to 10 results but there are only 8 types.

```
> db.map.aggregate([{"$match":{"amenity": {"$exists":1},
"amenity":"place_of_worship", "religion": {"$exists":1}}}, {"$group":
{"_id":"$religion", "count":{"$sum":1}}}, {"$sort":{"count":-1}},
{$limit:10}])
```

{ "_id" : "christian", "count" : 412 }
{ "_id" : "jewish", "count" : 5 }
{ "_id" : "muslim", "count" : 3 }
{ "_id" : "hindu", "count" : 2 }
{ "_id" : "buddhist", "count" : 1 }
{ "_id" : "scientologist", "count" : 1 }
{ "_id" : "bahai", "count" : 1 }
{ "_id" : "unitarian_universalist", "count" : 1 }

Some detail about the Christians:

```
> db.map.aggregate([{"$match" : {"amenity" : "place_of_worship"}},
{"$group" : {"_id" : {"religion" : "christian", "denomination" :
"$denomination"}, "count" : {"$sum" : 1}}}, {"$sort" : {"count" : -1}}])
```

{ "_id" : { "religion" : "christian" }, "count" : 207 }
{ "_id" : { "religion" : "christian", "denomination" : "baptist" }, "count" : 73 }
{ "_id" : { "religion" : "christian", "denomination" : "methodist" }, "count" : 59 }
{ "_id" : { "religion" : "christian", "denomination" : "catholic" }, "count" : 22 }
{ "_id" : { "religion" : "christian", "denomination" : "lutheran" }, "count" : 19 }
{ "_id" : { "religion" : "christian", "denomination" : "presbyterian" }, "count" : 18 }
{ "_id" : { "religion" : "christian", "denomination" : "roman_catholic" }, "count" : 7 }
{ "_id" : { "religion" : "christian", "denomination" : "episcopal" }, "count" : 5 }
{ "_id" : { "religion" : "christian", "denomination" : "jehovahs_witness" }, "count" : 3 }
{ "_id" : { "religion" : "christian", "denomination" : "pentecostal" }, "count" : 3 }
{ "_id" : { "religion" : "christian", "denomination" : "orthodox" }, "count" : 3 }
{ "_id" : { "religion" : "christian", "denomination" : "anglican" }, "count" : 2 }
{ "_id" : { "religion" : "christian", "denomination" : "assembly_of_god" }, "count" : 1 }
{ "_id" : { "religion" : "christian", "denomination" : "evangelical" }, "count" : 1 }
{ "_id" : { "religion" : "christian", "denomination" : "Progressive National Baptist" }, "count" : 1 }
{ "_id" : { "religion" : "christian", "denomination" : "United Methodist" }, "count" : 1 }
{ "_id" : { "religion" : "christian", "denomination" : "friends" }, "count" : 1 }
{ "_id" : { "religion" : "christian", "denomination" : "Episcopal" }, "count" : 1 }
{ "_id" : { "religion" : "christian", "denomination" : "african_methodist_episcopal" }, "count" : 1 }
{ "_id" : { "religion" : "christian", "denomination" : "non-denominational" }, "count" : 1 }
{ "_id" : { "religion" : "christian", "denomination" : "seventh_day_adventist" }, "count" : 1 }
{ "_id" : { "religion" : "christian", "denomination" : "united_methodist" }, "count" : 1 }

And, among the Christian group, where are located the most cited places of worship? (Baptist)

```
 > db.map.aggregate([{"$match" : {"amenity" : "place_of_worship"}},
{"$group" : {"_id" : {"religion" : "christian", "denomination" : "baptist",
"address.city" : "$address.city"}, "count" : {"$sum" : 1}}}, {"$sort" :
{"count" : -1}}, {$limit:5}])
```

```
{ "_id" : { "religion" : "christian", "denomination" : "baptist",
"address.city" : "Baltimore" }, "count" : 13 }
{ "_id" : { "religion" : "christian", "denomination" : "baptist",
"address.city" : "Middle River" }, "count" : 3 }
{ "_id" : { "religion" : "christian", "denomination" : "baptist",
"address.city" : "Washington" }, "count" : 3 }
{ "_id" : { "religion" : "christian", "denomination" : "baptist",
"address.city" : "Bethesda" }, "count" : 2 }
{ "_id" : { "religion" : "christian", "denomination" : "baptist",
"address.city" : "Essex" }, "count" : 2 }
```

Baltimore has the biggest number of Christian Baptist places of worship.

5 OTHER IDEAS ABOUT THE DATASET

I found while working on the data that some of it was not standardized: for example, when auditing the street names, we found that some contributors had entered different denominations for the same "reality". That happened also for the Postal Codes, although in a different way.

In this particular case, it was cleaned up but the results will not be incorporated in the Mapzen database so others will not find them.

This could be improved in two ways:

- first, set a standard of codes so everybody knows the exact rules about, for example, how to name a Street. Some filters including those set of codes could be used to validate the data the contributors enter in the database.
- Second, a more flexible system to aggregate the cleaned (and validated) data into the database could also help. In this exercise we are not cleaning a big portion of the map, but all these "micro-cleanings" added up could make in the long run a great difference.

There is another problem, missing data in the database: this is an issue that will never end because there will be always more data, more and broader detail, to be added. Still, some minimum values should be mandatory, although some may argue that even below the minimum is better that nothing.

Apart from that, some other techniques could be used like imputing missing values like getting other values from within the same node and/or cross validating incorrect or missing data from other databases.

Usually the Government and some companies (for instance, Google) have that information (or at least some of it) available for free or at a very low cost, but other companies might have proprietary data that they do not want to share with others because they are making profit from it; in some cases, the best and more accurate data is just not freely available except if some high fees are paid to those companies (for example navigation system companies like Garmin, TomTom and others).