# OpenStreetMap Project

# Data Wrangling with MongoDB

***Abdelrahman Saud***

Map Area: Riyadh, Saudi Arabia

After downloading Riyadh city from mapzen and running it against provisional data.py file
I noticed seven problems, which will be discussed in the following section.

## 1. Problems Encountered

**Dummy data**
after running auditing script over tags revealed dummy tags like "Fixme"
with values like "to be updated" or "confirm this is a gas station"
I imported the node normally ignoring these tags

**Multiple languages for city values**
while auditing the tag city, I found Riyadh typed in arabic and english
I updated all the values to "Riyadh" for consistency

**Street Abbreviations**
during the auditing of street names, I found street and road were sometimes abbreviated
and sometimes the full word were used, I updated all values to the full word

**Invalid values for state**
after running the auditing script, I found that the tag state contained country names
I ignored this tag because the country is already known and also sometimes there's country tag present

**Duplicate suburb values**
audient the suburb tag, I found that it contained duplicate values from street tag
I ignored this tag

**P.O Box appended to postal code**
some postal code tags also contained P.O numbers, I splitted them in their proper fields

**Invalid country tag values**
while most of country values were correct "SA" some had the value "IT"
I updated those values to "SA"

## 2. Data Overview

This section contains basic statistics about the dataset and the MongoDB queries used to gather them.

**#File sizes**

Riyadh.osm ..... 71 MB
Riyadh.osm.json .... 106 MB

**#Number of documents**
> db.Riyadh.find().count()
383863

**# Number of nodes**
> db.Riyadh.find({"type":"node"}).count()
311945

**# Number of ways**
> db.Riyadh.find({"type":"way"}).count()
71918

**#Number of unique users**

```
> db.Riyadh.distinct("created.user").length
272
```

**#Top contributing users**

user count

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

{ "_id" : "Seandebasti", "count" : 175880 } 45.8%
{ "_id" : "Rub21", "count" : 45039 } 11.7%
{ "_id" : "bauma", "count" : 30650 } 7.98%
{ "_id" : "Cicerone", "count" : 27864 } 7.26%
{ "_id" : "hsarslan", "count" : 17839 } 4.65%

## 3. Additional Ideas

While auditing the data some counts of tags suggested that the data entered is less than the node and way count

**#Documents with tag city count and values**

```
  "Neusiedl am See": 1,
  "riyadh": 5,
  "RIYADH": 1,
  "Carrara": 1,
  "Riyadh": 66,
  "\u0627\u0644\u0631\u064a\u0627\u0636": 5
```

**#Documents with tag Country count and values**

```
  "SA": 7,
  "IT": 1
```

So I ran the below queries to investigate further

**#count of name in specific languages**
**count of arabic names**

```
>
db.Riyadh.aggregate([{"$match":{"name:ar":{"$exists":1}}},{"$group":{"_id":null,"count":{"$sum":1
}}}])
```

{ "_id" : null, "count" : 507 }

**count of english names**

>
db.Riyadh.aggregate([{"$match":{"name:en":{"$exists":1}}},{"$group":{"_id":null,"count":{"$sum":1}}}])
{ "_id" : null, "count" : 824 }

**count of german names**

>
db.Riyadh.aggregate([{"$match":{"name:de":{"$exists":1}}},{"$group":{"_id":null,"count":{"$sum":1}}}])
{ "_id" : null, "count" : 420 }

Although the counts are far less than the number of documents, having those close values I suspected that
they should not be added and they might actually be the same documents with different tags

ar & en
409

ar & de
360

en & de
389

ar & en & de
359

This proved that great portion of them are indeed same documents
Checking the count of documents having street value

db.Riyadh.aggregate([{"$match":{"address.street":{"$exists":1}}},{"$group":{"_id":null,"count":{"$sum":1}}}])
{ "_id" : null, "count" : 93 }

It was not surprising to find that there are no amenity in all documents, given the counts mentioned above

>
db.Riyadh.aggregate([{"$match":{"amenity":{"$exists":1}}},{"$group":{"_id":null,"count":{"$sum":1}}}])
>

It is also revealed that most of the documents didn't have address at all

db.Riyadh.aggregate([{"$match":{"address":{"$exists":0}}},{"$group":{"_id":null,"count":{"$sum":1}}}])
{ "_id" : null, "count" : 383753 }

db.Riyadh.aggregate([{"$match":{"address":{"$exists":1}}},{"$group":{"_id":null,"count":{"$sum":1}}}])
{ "_id" : null, "count" : 110 }

## Conclusion

After the review of the data, it is obvious that it is far from complete, having that number of nodes with gps data
while no tags for addresses, amenities, or even name for the node.

To improve this dataset there are several options

1. 1- Imputing missing data
   This option is inappropriate because most of the data has only name of the user and the location, there's nothing else to be used like address or even amenity.
2. 2- Gamification would be a good choice here to increase the participation and gather more data, however this would cost developing the application to be used in that process, but on the other hand the data would be accurate and up-to-date
3. 3- Cross-referencing missing data from Google Maps API, this option has lower costs than the previous one, but it might have its limitation due to the way google gathers the data, it might have outdated or even missing data