OpenStreetMap Project

Data Wrangling with MongoDB

Nicole Mister

Map Area: DC-Baltimore, MD, United States

<https://s3.amazonaws.com/metro-extracts.mapzen.com/dc-baltimore_maryland.osm.bz2>

# Problems Encountered in the Map

I did not find any issues with the data itself. That is, I was not able to find issues with the data I was able to process. I was not able to clean the entire DC-Baltimore OSM file. The OSM file is about 2.8 gigabytes. Even after running for several hours, the cleaning code failed to complete. Instead, I took a sample of every 50,000th element using the given sampling code. The sample took over an hour to process. I ran the cleaning code on that sample. The queries below are on that sample.

Below are the audits I ran to check the data validity. From the sampling, the postcodes and streets were cleaned correctly.

# Find all distinct post codes

print db.balt.distinct('address.postcode')

[u'21128', u'21222', u'21133', u'20020', u'20019', u'21093', u'21236', u'21204', u'21244', u'21218', u'21217', u'21227', u'21214']

# Find all distinct streets.

print db.balt.distinct('address.street')

[u'Anns Garden Way', u'Eastern Avenue', u'Oak Trace Way', u'Savannah Terrace Southeast', u'Grant Street Northeast', u'Fountain Hill Drive', u'Fox Brier Lane', u'West Joppa Road', u'Belclare Road', u'Fairbrook Road', u'Cator Avenue', u'Harlem Avenue', u'Benson Avenue', u'Glenoak Avenue']

Another issue, I did not find any data regarding amenities, which may have provided the opportunity for more interesting queries.

# Find all distinct ammenties

print db.balt.distinct('ammenity')

[]

# Data Overview

Below are the dataset statistics and MongoDB queries.

# Number of documents

> coll.find().count()

283

# Filesize

> coll.dataSize()

67689

# Number of nodes

db.balt.find({'type': 'node'}).count()

253

# Number of ways

db.balt.find({'type': 'ways'}).count()

0

#Number of unique users

print len(db.balt.distinct("created.user"))

63

# Top 5 contributing users

import pprint

cursor = db.balt.aggregate([

{'$group': {'\_id': "$created.user", 'count': {'$sum': 1}}},

{'$sort': {'count': -1}},

{'$limit': 5}

])

for doc in cursor:

pprint.pprint(doc)

{u'\_id': u'EP\_Import', u'count': 56}

{u'\_id': u'mpetroff-imports', u'count': 41}

{u'\_id': u'asciiphil', u'count': 28}

{u'\_id': u'woodpeck\_fixbot', u'count': 25}

{u'\_id': u'aude', u'count': 18}

# Types and counts of buildings

import pprint

cursor = db.balt.aggregate([

{'$group': {'\_id': "$building", 'count': {'$sum': 1}}},

{'$sort': {'count': -1}}

])

for doc in cursor:

pprint.pprint(doc)

{u'\_id': None, u'count': 266}

{u'\_id': u'yes', u'count': 9}

{u'\_id': u'residential', u'count': 5}

{u'\_id': u'detached', u'count': 2}

{u'\_id': u'house', u'count': 1}

# Additional Ideas

The DC-Baltimore area is missing data, but the data that it contains is clean. With more data, users would be able perform more queries and draw more interesting conclusions. For example, the addition of the amenities field would allow users to query what type of amenities are most popular in what postcodes.