NoSQL Project

Eline Rion

December 15, 2016

Contents

Introduction

"What is the best part of New-York to live in ?" is the question we are going to answer in this project. We will use the help of a website https://data.cityofnewyork.us which stores different types of data about the city of New-York. The project will be articulated in five different steps :

- choose the datasets to use
- choose the database to work with
- work on the datasets, if needed, to modify them
- load the datasets into the database
- query the database

We'll present in detail each of these steps in this report. First we'll give a defition of the "best part of town" then explain which datasets we chose and why and finally explaining the procedure to get to the result which in our case will be the best borough to live in. This third part is going to be detailled later.

I. Definition of the best part of town to live in

Let first precise that we are looking for an appartment for a young lady that has just arrived in town. This laday, like everyone else, has features that can define herself. Let's call her Sarah. Sarah has a thing for food, let say that she lives for food. Recently she learnt herself into cooking and we cannot say that she is succeeding : she managed to start fire in her kitchen several times already. In her new appartment she trully want to keep practicing but as her new job is about to take almost all of her time she plans on going to restaurant several times a week.
Even if going to the restaurant does not bother her she tends to be sightly agoraphobic.
She is also someone that makes plans on her future : she is certain she will find the love of her life in New-York and start growing a family with him. That's why she also has some criteria considering her future hild/children : she wants them to have the best education in maths.

Considering this definition of Sarah we can summarize here criterium to find the best appartment : * near firehouse in case she tryes cooking again * near good restaurants because she definitely is not a good cooker * the levek of maths in children should not be bad * she would also prefer a place not too crowded.

II. Links to the database

To answer Sarah's request we selected 4 datasets :

- *FDNY Firehouse Listing* : listing of all NYC fire houses with addresses. To be sure that in case she starts fire in her kitchen a fireman could be there pretty fast.
- *New York City Population By Neighborhood Tabulation Areas* : population Numbers By New York City Neighborhood Tabulation Areas. To find a neighbourhood not too crowded.
- *Math Test Results 2006-2012 - Borough - Gender* : latest available data and trends in the state assessment results of math for grades 3 through 8. Data are disaggregated by borough and gender. To confirm the maths level in her area
- *DOHMH New York City Restaurant Inspection Results* : this dataset provides restaurant inspections, violations, grades and adjudication information. To have the list of all the restaurants in her area and be sure that she won't get food poisoning going there.

You can find the links to these datasets on the *links.txt* files.

III. Procedure to get the answer

1. Choice of the database

We decide for this project to use **Neo4j**. This choice is first personnal : I already used MongoDB during my internship last summer so I would not learn much using it. Then between Neo4j and Cassandra, as all my datasets will be linked with each other through the borough, Neo4j seemed to be a good option. Moreover as we already spent time in class studying *Cypher* to interact with Neo4j then I could reuse what we learnt.

2. Preparation of the datasets

After downloading the four datasets from the website as CSV (Comma Separated Value) we won't directly use them. We first need to modify them in order to select for example only the columns we need. Moreover we had to prepare the data to use it on Neo4j. The script used to modify and prepare these datasets will be written in python.

- Borough

Our four datasets will all be connected to one dataset of five rows containing the name of the different Borough in New-York : we create that dataset from another one for example from the one about the firehouses. This dataset has 2 columns

Id	Borough
0	Manhattan
1	Bronx
2	Brooklyn
3	Queens
4	Staten Island

- Firehouse

In this dataset we have to modify the row 62 which has a comma as part of a value and as the comma is reserved for the separator we change it. Then we create an index, to be able to identify each of the firehouse. This dataset counts 3 columns :

Id	FacilityName	FacilityAddress
0	Engine 4/Ladder 15	42 South Street

- Population

In this dataset the only thing to do is to create the index. We obtain 6 columns :

Id	Year	FIPS.County.Code	NTA.Code	NTA.Name	Population
0	2000	5	BX01	Claremont-Bathgate	28149

- MathsTest

In this dataset we only keep 5 columns and create an index which leads us to 6 columns :

Id	Grade	Year	Demographic	Number Tested	Mean Scale Score
0	3	2006	Female	7984	664

- Restaurant

In this dataset we keep 8 columns and create an index as well :

Id	DBA	SCORE	BUILDING	BORO	CUISINE DESCRIPTION
0	THE RIVER CAFE	10.0	1	BROOKLYN	American

GRADE	INSPECTION DATE	STREET
A	06/25/2014	WATER STREET

Notice the difference

Let notice that the 3 tables **Firehouse**, **Population** and **MathsTest** don't have a column *Borough* but **Restaurant** does.
For the first three tables this column will be added thanks to the relationship **-IsIn** in Neo4j which will link each of this three tables to the table **Borough**. We could have done the same thing for the table **Restaurant** but as you want this code to be reproducible and as it takes a huge time (more than 2 hours) to create that relationship I decided not to do so. You will still find the code (commented) in the script to show you how to do it.

All the python script needed to create/modify these tables can be found on the *script.py* file.
In order to run that script please make sure that the *script.py* file is on the same directory as the datasets downloaded directly from the website and then open a terminal, using cd move to the directory containing the files and execute `python script.py`.

3. Loading data into the database

What we have to load in the database ? Well we have 5 tables having the informations about respectively the list of the boroughs of New-York, the firehouses in New-York, the population in New-York, the results of maths test among children in New-York and the restaurants in New-York.
Then we have 3 other tables each linking one of this last four tables (expect *restaurants*) : to the *Borough* one thanks to their Id.

When loading the data we actually have to make sure that the columns representing a number are loaded as integer so we can do some operations on them (maximum, average,. . .).

The code used to load the data can be found on the *loadData.txt* file. Each block of code need to be entered on the interface of Neo4j (Cypher).

Before loading the data, make sure to have moved the 8 files :

- borough.csv
- firehouse.csv
- mathsResult.csv
- population.csv
- restaurant.csv
- firehouse_is_in.csv
- mathsTest_is_in.csv
- population_is_in.csv

in the folder /usr/share/neo4j/import.

4. Process the data

In the same file *loadData.txt* we should find 5 lines for the creation of INDEX : this should make access to the tables in Neo4j faster. No other work will be made on the data once you executed all the codes in this file (in the command line of Neo4j). What we can do now is query the data to find the best borough to live in for Sarah.

5. Query the data

We'll use **Cypher** syntax to querry the data. The main querys are gonna be about : * findind the borough with the more firehouses, * where the mean Score of restaurants is the higher, * where the level of maths of children is the best, * and where the total population is the lowest.

You'll find all the querys in the file *querys.txt*.

6. Display the result

In this section we'll consider the opinion of Sarah. Indeed thinking that you'll find one borough that respects the four characteristics cited above is utopian. So we'll rank each neighbourhood according to each criterion and Sarah will decide which one is the most important and hence decide of the neighbourhood she wants to leave in. We executed some other querys to help her make her decision. Above you'll find the principals ranking and some other results followed by the choice of Sarah.

- borough with the more firehouses :

Brooklyn	Queens	Manhattan	Bronx	Staten Island

- borough ranked according to the population, increasing order :

Staten Island	Bronx	Manhattan	Queens	Brooklyn

- ranked according to the mean Score restaurants, decreasing order :

Staten Island	Brooklyn	Manhattan	Queens	Bronx

- ranked according to the level of maths, decreasing order :

Queens	Staten Island	Manhattan	Brooklyn	Bronx

We presented these results to Sarah and after a few days of reflections, she decided that as she won't have time for cooking the proximity of firehouses was not that important so she decided to find an appartment in **STATEN ISLAND**.

Afterwards we showed her what were the top 5 type restaurants in Staten Island and she was even more excited to move in : she has a thing for pizza.
Top 5 type restaurants in Staten Island : American, Italian, Chinese, Pizza/Italian, Pizza, Mexican ;)

IV. Quick steps to reproduce the code

- download the files (links in links.txt)
- run `script.py`
- move files into usr/share/neo4j/import
- on neo4j in line command run each block of code of *loadData.txt*
- on neo4j in line comand run each block of code of *querys.txt*
- read part III)6) for conclusion

V. Benefits of this project

In this part I am stating what I learnt from this project.

- First, I got familiar with python and the library pandas. I never used pandas before this project so I can confidently say that thanks to this project I know how to deal with data frames in python.
- Even if we already used Neo4j in class and had to work with the Cypher language I am now way more confortable with it and thanks to some research for this project I can do almost everything.
- Having to buit the graph database of Neo4j made me understand better graph database and how and why to use it.
- This report has been made using Rmarkdown of Rstudio : my skills in markdown are strengthened.