**Assignment 2: NoSQL Data Arch with MongoDB and Neo4j**

**Report by Aaron Cashman and Brian Coveney**

**November 2017**

Table of Contents

[Setup: 2](#_Toc499498380)

[MongoDB Queries: 6](#_Toc499498381)

[PyMongo Queries 10](#_Toc499498382)

[Neo4j Doc Manager 11](#_Toc499498383)

[Neo4j Queries 12](#_Toc499498384)

# Setup:

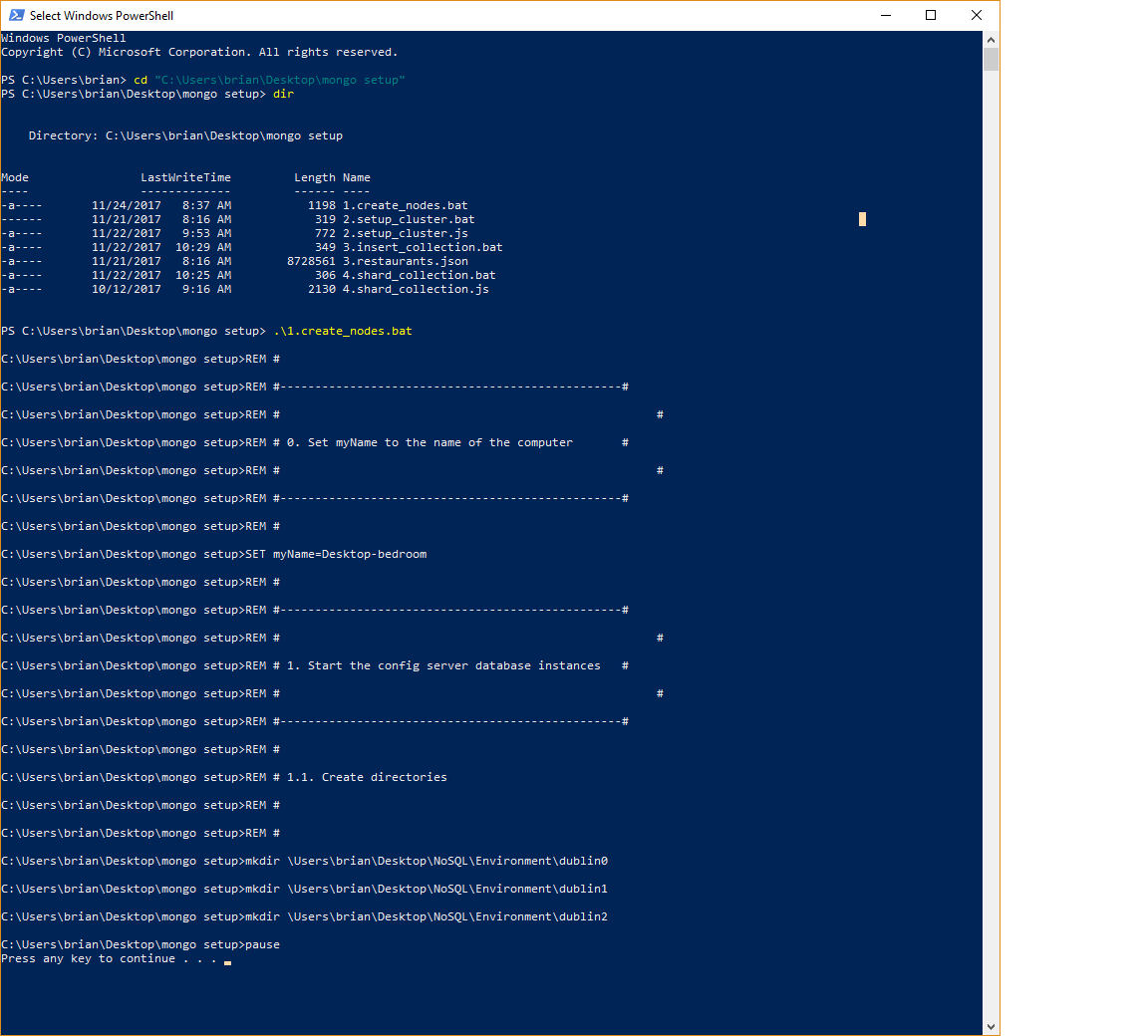


Fig1. Displays the initial set up

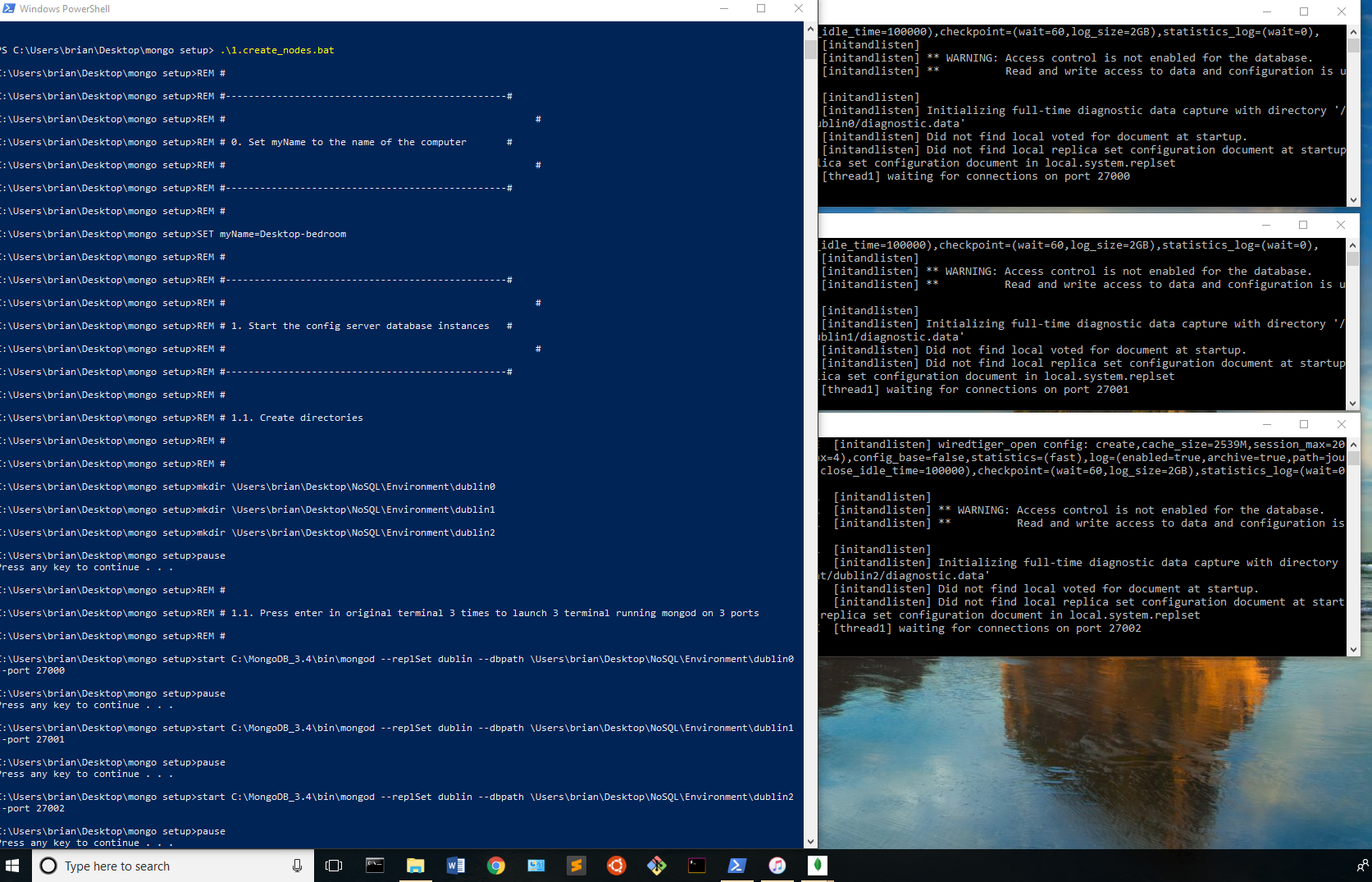


Fig1b. Displays the 3 mongo processes started on ports 27000, 27001 and 27002

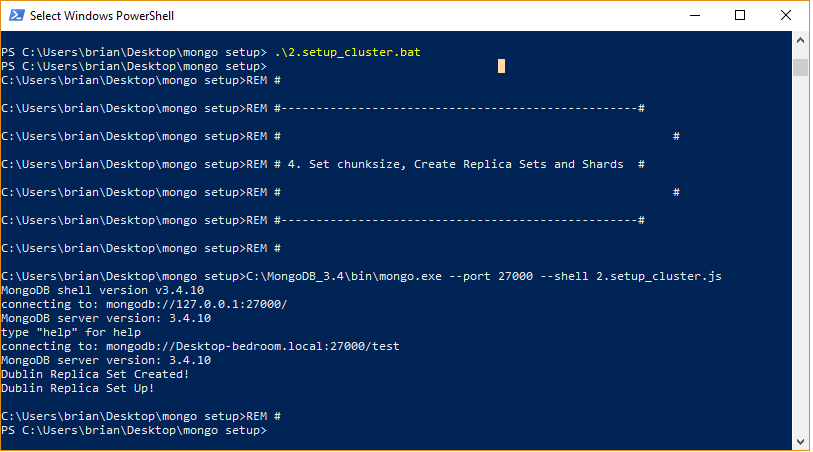


Fig2. Displays the setup of our cluster

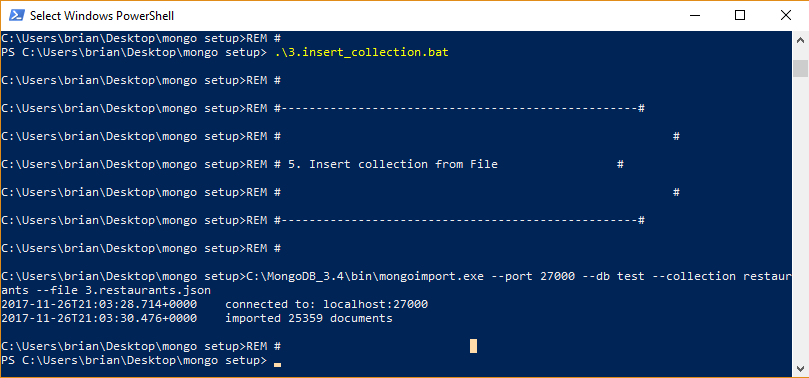


Fig3. Displays insertion of the restaurant collection

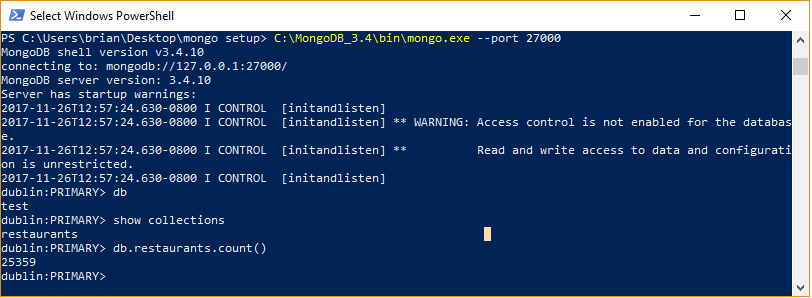


Fig4. Initial connection to our database

# MongoDB Queries:

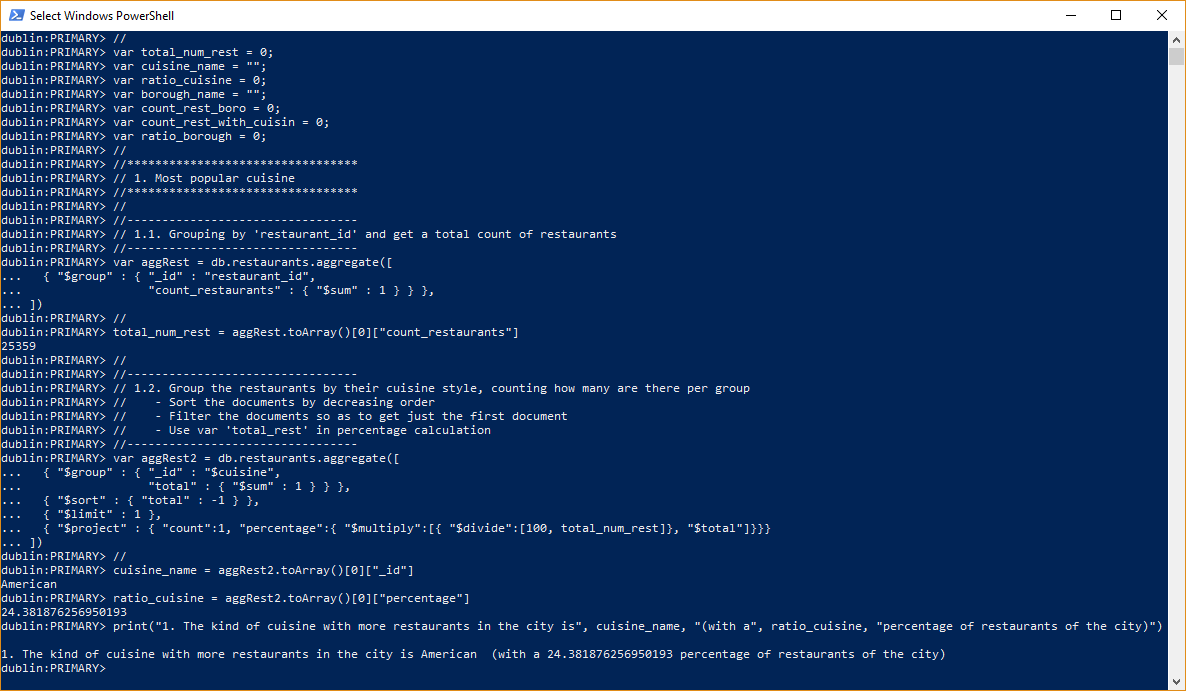


Fig5. Displays the MongoDB query No.1, with the following result:

*1. The kind of cuisine with more restaurants in the city is* ***American*** *(with* ***a 24.381876256950193*** *percentage of restaurants of the city)*

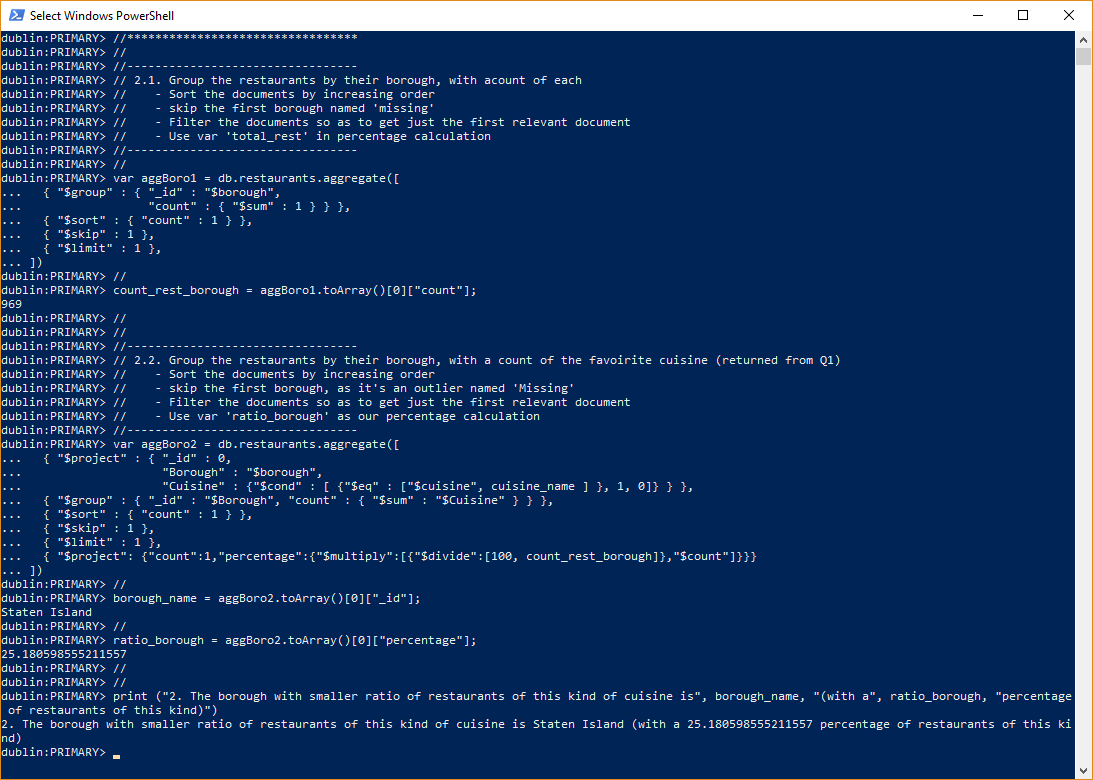
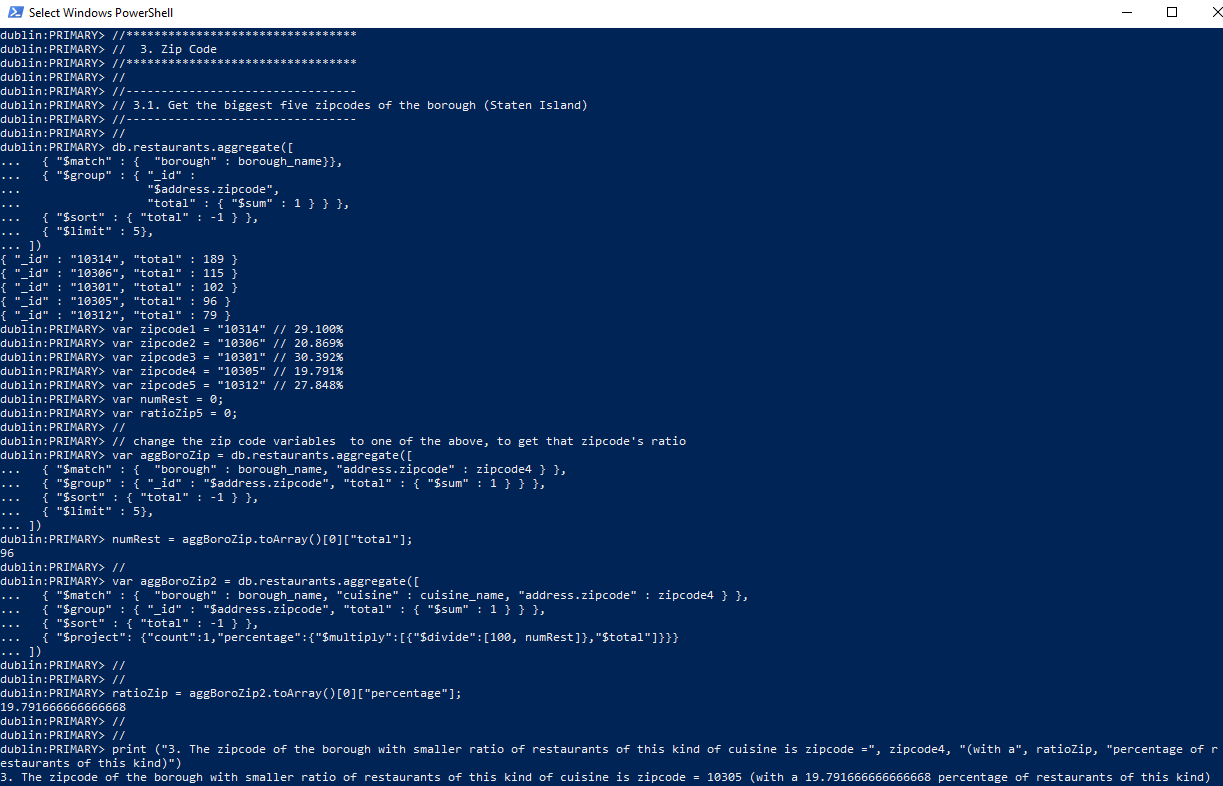


Fig6. Displays the MongoDB query No.2, with the following result:

*2. The borough with smaller ratio of restaurants of this kind of cuisine is* ***Staten Island*** *(with a* ***25.180598555211557*** *percentage of restaurants of this kind)*



*Fig7. Displays the MongoDB query No.3, with the following result:*

*3. The zipcode of the borough with smaller ratio of restaurants of this kind of cuisine is zipcode =* ***10305*** *(with a* ***19.791666666666668*** *percentage of restaurants of this kind)*

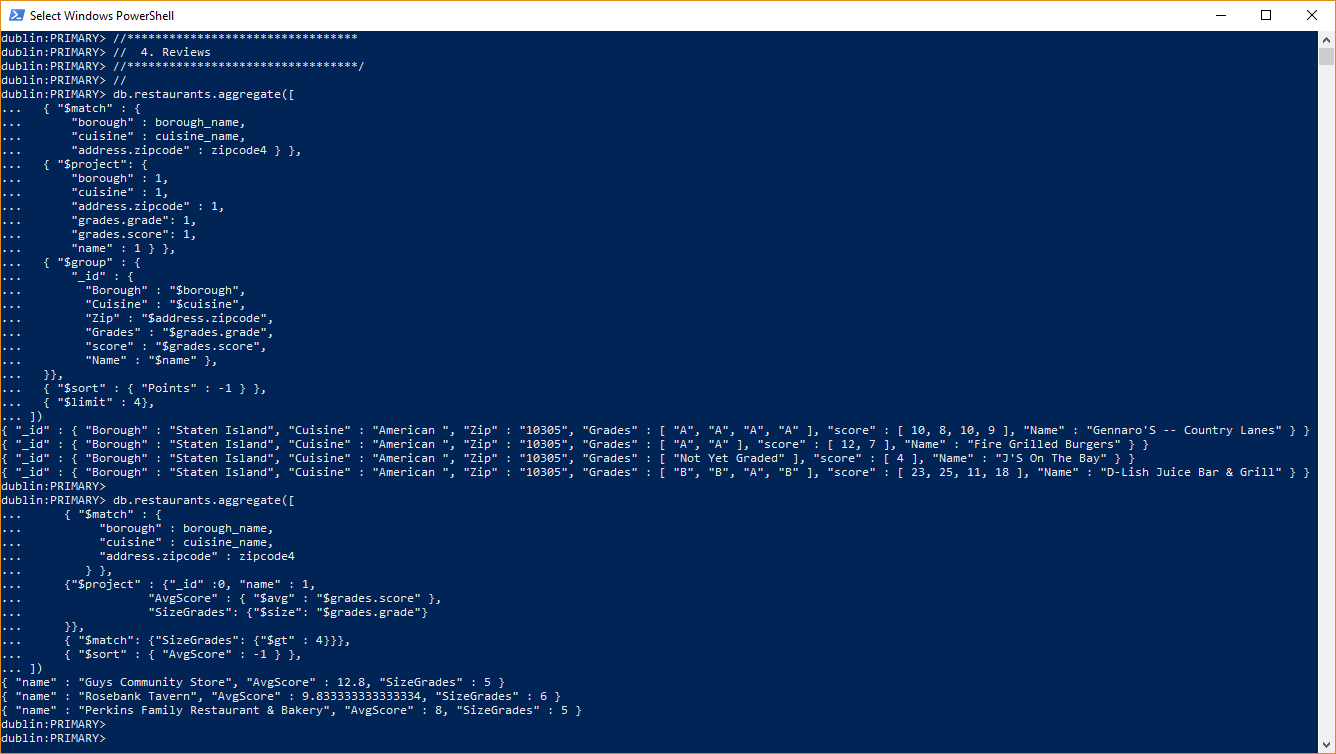


Fig7. Displays the MongoDB query No.4, with the following result:

*{ "name" : "Guys Community Store", "AvgScore" : 12.8, "SizeGrades" : 5 }*

*{ "name" : "Rosebank Tavern", "AvgScore" : 9.833333333333334, "SizeGrades" : 6 }*

*{ "name" : "Perkins Family Restaurant & Bakery", "AvgScore" : 8, "SizeGrades" : 5 }*

# PyMongo Queries

Next, we display the results for the PyMongo queries, but for brevity we will only show Function No. 4 of the PyMongo query code.

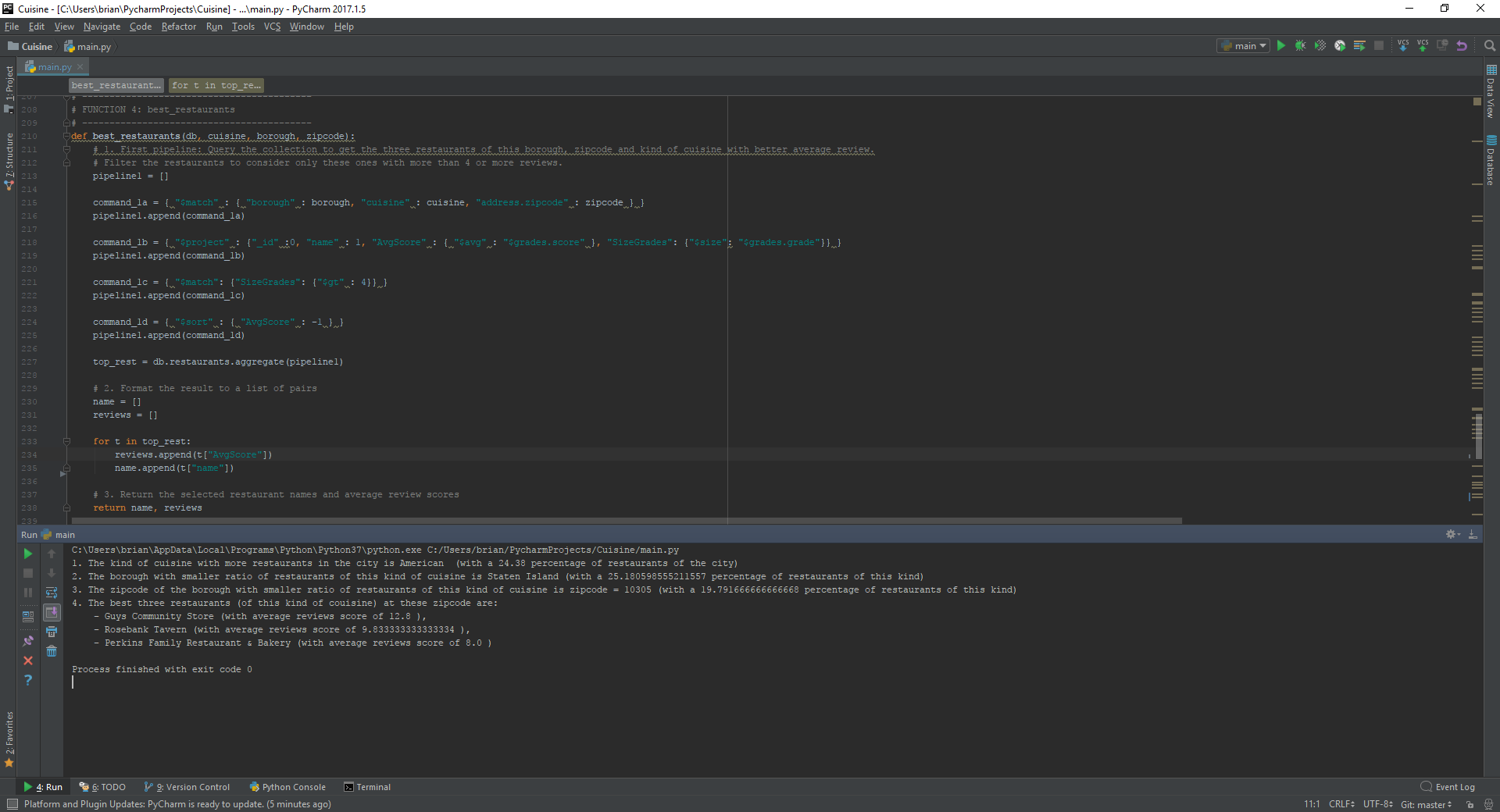


Fig8. Displays the PyMongo query No.4. All queries are displayed with the following result:

1. The kind of cuisine with more restaurants in the city is American (with a 24.38 percentage of restaurants of the city)

2. The borough with smaller ratio of restaurants of this kind of cuisine is Staten Island (with a 25.180598555211557 percentage of restaurants of this kind)

3. The zipcode of the borough with smaller ratio of restaurants of this kind of cuisine is zipcode = 10305 (with a 19.791666666666668 percentage of restaurants of this kind)

4. The best three restaurants (of this kind of couisine) at these zipcode are:

- Guys Community Store (with average reviews score of 12.8 ),

- Rosebank Tavern (with average reviews score of 9.833333333333334 ),

- Perkins Family Restaurant & Bakery (with average reviews score of 8.0 )

# Neo4j Doc Manager

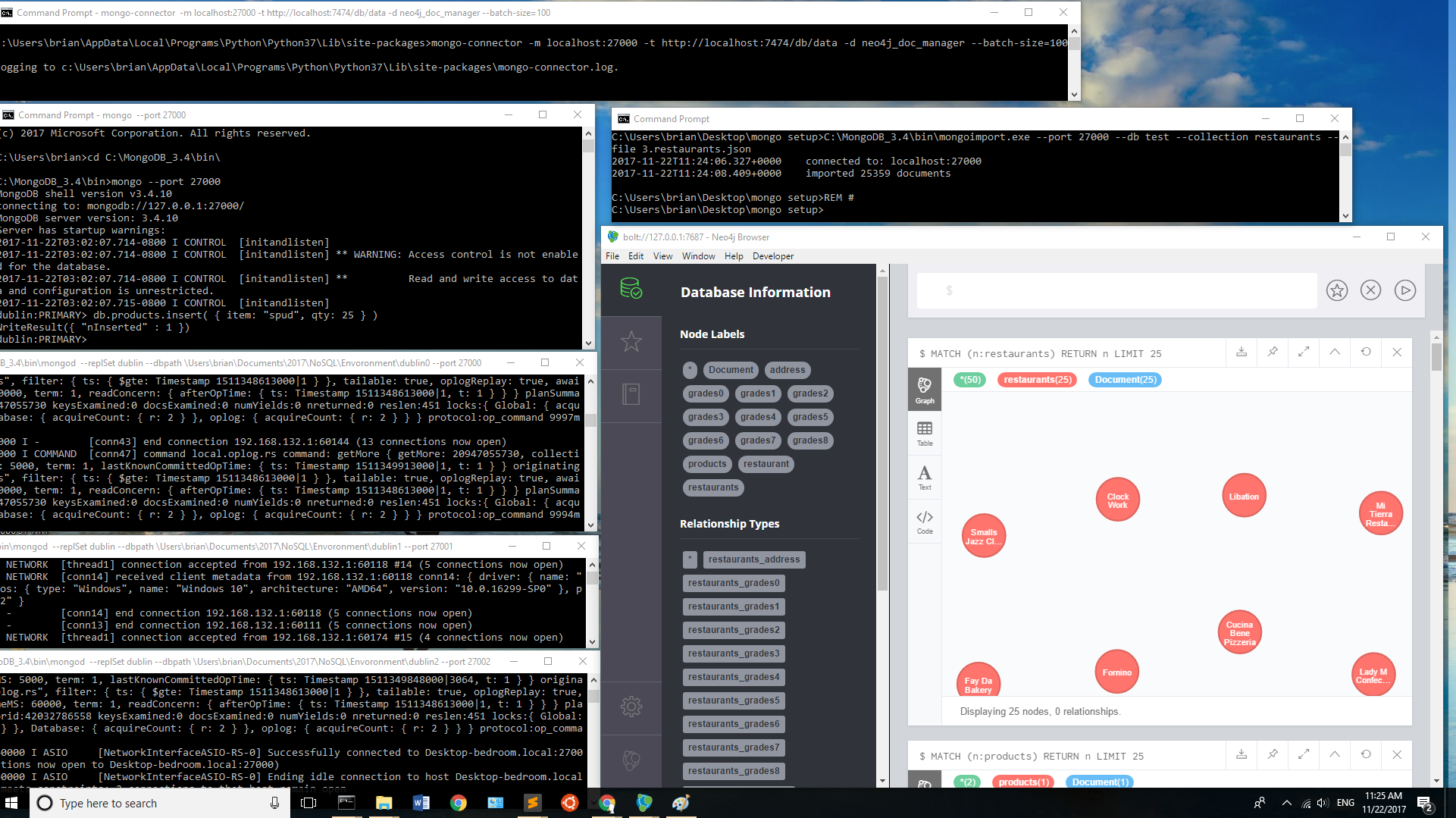


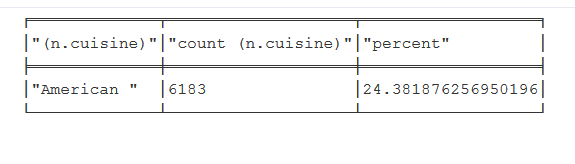
Fig8. Displays neo4j-doc-manager, mongo instances on ports 27000 – 27002, and Neo4j Browser

# Neo4j Queries

**Find the most\_popular\_cuisine:**

**Returns the name of the kind of cuisine with higher number of restaurants in New York and its ratio (percentage).**

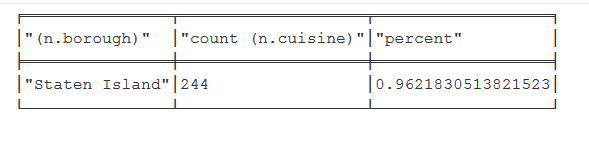
*match (n:restaurants) with count(n.cuisine) as total match (n:restaurants) return (n.cuisine),count (n.cuisine),((100.0\*((count(n.cuisine))))/total) as percent order by count(n.cuisine) DESC Limit 1*



**Find the ratio\_per\_borough\_and\_cuisine :**

**Returns the name of the borough with smaller percentage of restaurants of the kind of cuisine from (i). It also returns the proper percentage.**

*match (n:restaurants) with count(n.cuisine) as total match (n:restaurants) where (n.cuisine) = "American " return (n.borough),count (n.cuisine), ((100.0\*((count(n.cuisine))))/total) as percent order by n.borough DESC Limit 1*



**Find the ratio\_per\_zipcode:**

**Returns the name of the zipcode with smaller percentage of restaurants of a particular kind of cuisine from (i) and (ii). It also returns the proper percentage**

*match (n:restaurants) with count(n.cuisine) as total match (n:restaurants)-[:restaurants\_address]->(a:address) where n.cuisine = "American " and n.borough = "Staten Island" return a.zipcode,Count(n.cuisine),((100.0\*((count(n.cuisine))))/total) as percent order by a.zipcode LIMIT 1*

