NoSQL Data Architectures Project 2- 2020

Roshan Sreekanth – R00170592

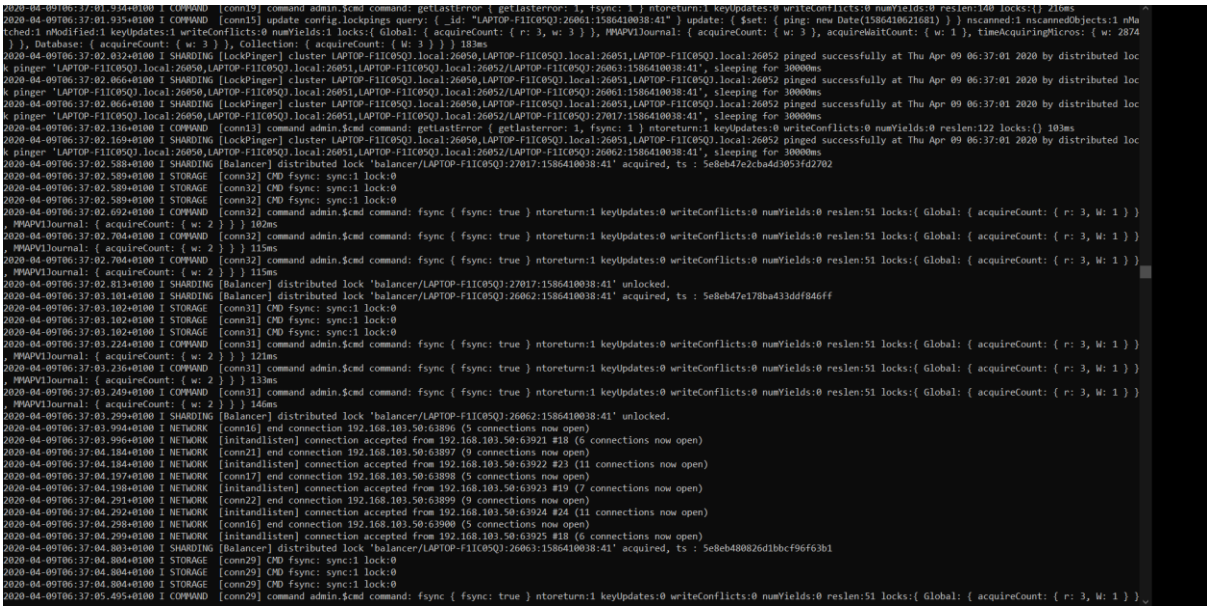
roshan.sreekanth@mycit.ie

Note: I have setup a cluster and have also commented the scripts.

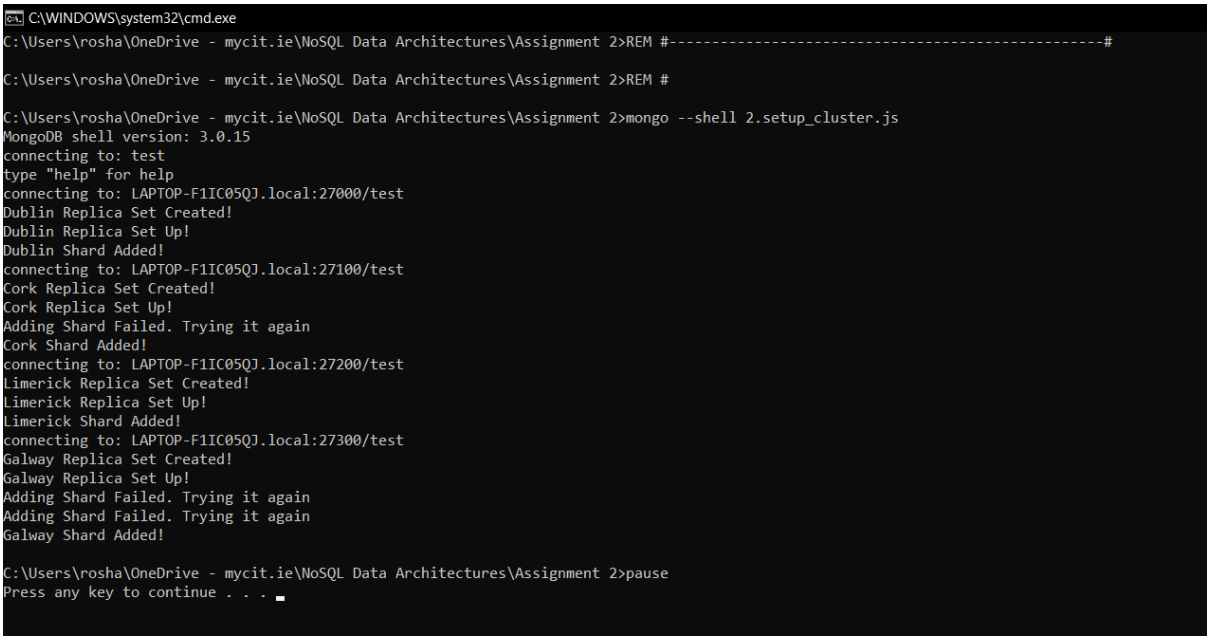
Contents

PART 1

CREATE NODES



SETUP CLUSTER



INSERT COLLECTION

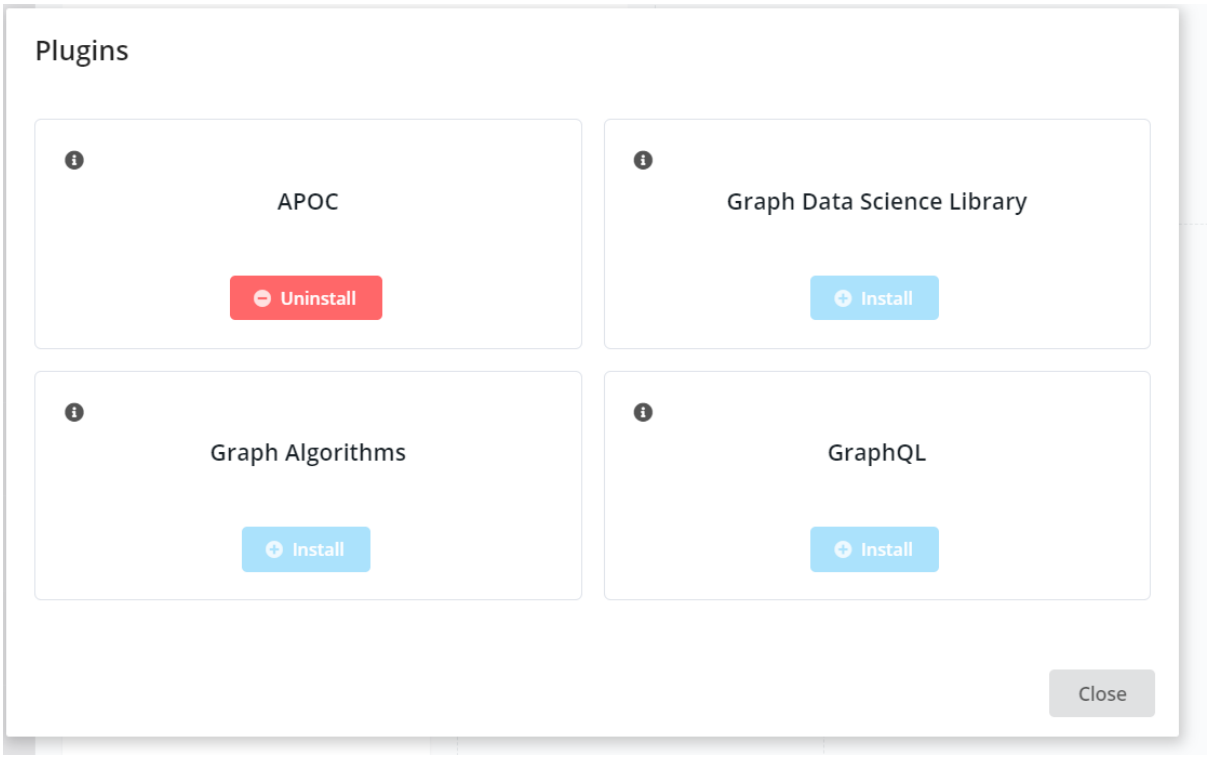
```
C:\WINDOWS\system32\cmd.exe

C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>REM #

C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>REM #------------------------------------------------#

C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>REM #                                          #

C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>REM # 5. Insert collection from File           #

C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>REM #                                          #

C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>REM #------------------------------------------------#

C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>REM #

C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>mongoimport --db test --collection restaurants --drop --file 3.restaurants_dataset.json
2020-04-09T06:39:15.275+0100    connected to: localhost
2020-04-09T06:39:15.277+0100    dropping: test.restaurants
2020-04-09T06:39:18.255+0100    [#########..............] test.restaurants     5.1 MB/11.2 MB (45.4%)
2020-04-09T06:39:20.074+0100    imported 25359 documents

C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>pause
Press any key to continue . . .
```

SHARD COLLECTION

```
C:\WINDOWS\system32\cmd.exe
                5 : Failed with error 'chunk too big to move', from dublin to cork
                6 : Failed with error 'chunk too big to move', from dublin to limerick
  databases:
        {  "_id" : "admin",  "partitioned" : false,  "primary" : "config" }
        {  "_id" : "test",  "partitioned" : true,  "primary" : "dublin" }
                test.restaurants
                        shard key: { "cuisine" : 1, "borough" : 1 }
                        chunks:
                                cork    6
                                dublin  12
                                galway  6
                                limerick        6
                        too many chunks to print, use verbose if you want to force print

--- Sharding Status ---
  sharding version: {
        "_id" : 1,
        "minCompatibleVersion" : 5,
        "currentVersion" : 6,
        "clusterId" : ObjectId("5e8eb248826d1bbcf96f6370")
}
  shards:
        {  "_id" : "cork",  "host" : "cork/LAPTOP-F1IC05QJ.local:27100,LAPTOP-F1IC05QJ.local:27101,LAPTOP-F1IC05QJ.local:27102" }
        {  "_id" : "dublin",  "host" : "dublin/LAPTOP-F1IC05QJ.local:27000,LAPTOP-F1IC05QJ.local:27001,LAPTOP-F1IC05QJ.local:27002" }
        {  "_id" : "galway",  "host" : "galway/LAPTOP-F1IC05QJ.local:27300,LAPTOP-F1IC05QJ.local:27301,LAPTOP-F1IC05QJ.local:27302" }
        {  "_id" : "limerick",  "host" : "limerick/LAPTOP-F1IC05QJ.local:27200,LAPTOP-F1IC05QJ.local:27201,LAPTOP-F1IC05QJ.local:27202" }
  balancer:
        Currently enabled:  yes
        Currently running:  no
        Failed balancer rounds in last 5 attempts:  0
        Migration Results for the last 24 hours:
                19 : Success
                6 : Failed with error 'chunk too big to move', from dublin to galway
                6 : Failed with error 'chunk too big to move', from dublin to cork
                6 : Failed with error 'chunk too big to move', from dublin to limerick
  databases:
        {  "_id" : "admin",  "partitioned" : false,  "primary" : "config" }
        {  "_id" : "test",  "partitioned" : true,  "primary" : "dublin" }
                test.restaurants
                        shard key: { "cuisine" : 1, "borough" : 1 }
                        chunks:
                                cork    7
                                dublin  11
                                galway  6
                                limerick        6
                        too many chunks to print, use verbose if you want to force print


C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>pause
Press any key to continue . . .
```

PART 2

Installing APOC



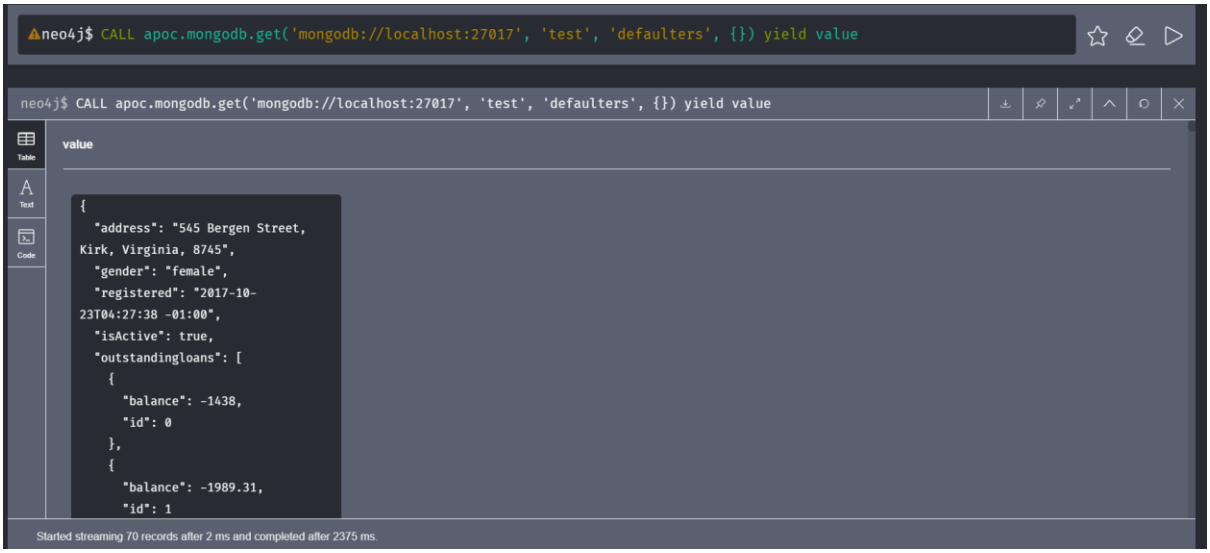
Creating “defaulters” collection

```
PS C:\Users\rosha> mongo
MongoDB shell version: 3.0.15
connecting to: test
> show collections
> db.createCollection("defaulters")
{ "ok" : 1 }
> show dbs
local  0.078GB
test   0.078GB
> show collections
defaulters
system.indexes
> _
```

Importing json file

```
PS C:\Users\rosha> mongoimport --db test --collection defaulters --file "C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2\defaulters.json" --jsonArray
2020-04-22T11:57:55.413+0100    connected to: localhost
2020-04-22T11:57:55.417+0100    imported 70 documents
PS C:\Users\rosha>
```

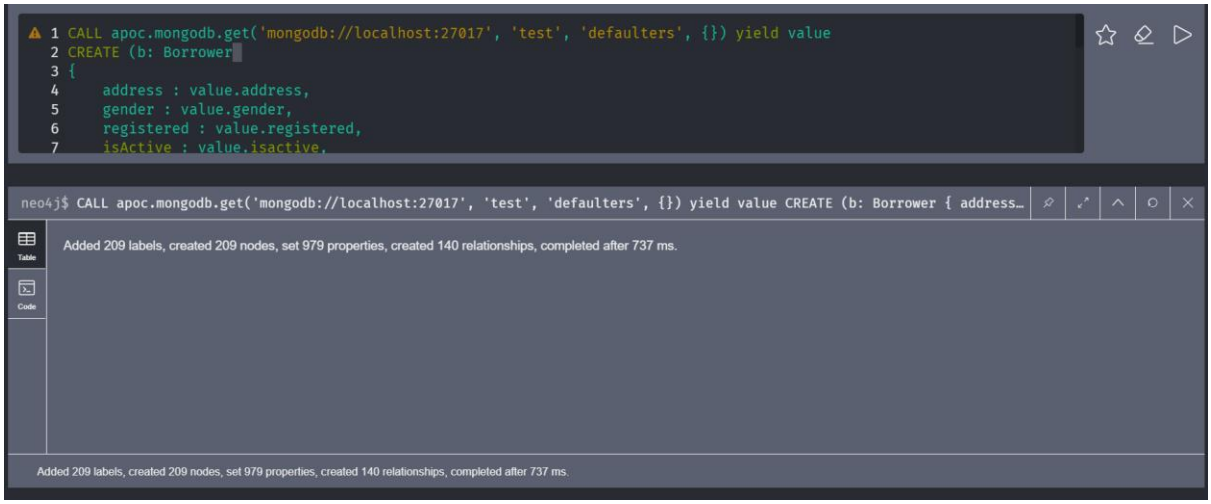
Viewing mongodb database through APOC



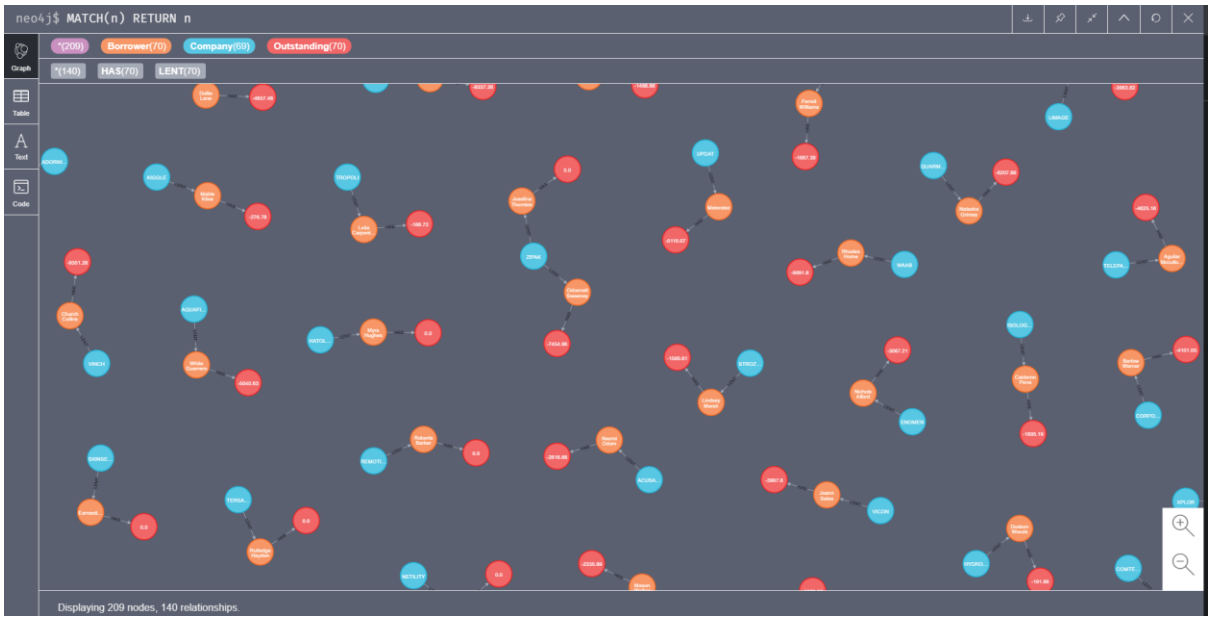
Creating nodes and relationships

```
CALL apoc.mongodb.get('mongodb://localhost:27017', 'test', 'defaulters', {}) yield value
CREATE (b: Borrower
{
   address : value.address,
   gender : value.gender,
   registered : value.registered,
   isActive : value.isactive,
   balance : toFloat(value.balance),
   eyeColor : value.eyeColor,
   phone : value.phone,
   name : value.name,
   age : toInteger(value.age),
   email : value.email
})

MERGE(c: Company
{
   company: value.company
})

MERGE(o : Outstanding
{
   firstquarter : toFloat(coalesce(value.outstandingloans[0].balance, 0)),
   secondquarter : toFloat(coalesce(value.outstandingloans[1].balance,0)),
   thirdquarter : toFloat(coalesce(value.outstandingloans[2].balance,0)),
   fourthquarter : toFloat(coalesce(value.outstandingloans[3].balance,0))
})
CREATE (c)-[:LENT]->(b)
```

CREATE (b)-[:HAS]->(o)



Viewing the nodes in the graph



Part 3

Are males or females on average bigger defaulters on their loans ?

The size of a person's loan default is calculated by summing the balances in their outstanding loans

```
db.defaulters.aggregate([
  { $unwind : "$outstandingloans" },
```

{ $group : { _id : {_id: "$_id", gender: "$gender", name: "$name", age: "$age"}, totalDebt : {$sum : "$outstandingloans.balance" } } },

{ $group : { _id : "$_id.gender", averageDebt : {$avg : "$totalDebt" } } },

{ $sort : { averageDebt: 1 }},

{ $limit : 1 }

])

```
> db.defaulters.aggregate([
...     { $unwind : "$outstandingloans" },
...     { $group : { _id :  {_id: "$_id", gender: "$gender", name: "$name", age: "$age"}, totalDebt : {$sum : "$outstandingloans.balance" } } },
...     { $group : { _id : "$_id.gender", averageDebt : {$avg : "$totalDebt" } } },
...     { $sort : { averageDebt: 1 }},
...     { $limit : 1 }
... ])
{ "_id" : "female", "averageDebt" : -9884.130312500001 }
```

MATCH (b:Borrower)--(o:Outstanding)

WITH b.gender AS Gender, sum(o.firstquarter + o.secondquarter + o.thirdquarter + o.fourthquarter)/COUNT(b.gender) AS averageAmount

RETURN Gender, averageAmount ORDER BY averageAmount, Gender LIMIT 1

What number of male loan defaulters who have 3 loans or more have defaulted on loans in total valued over 15,000 ?

db.defaulters.aggregate([

{ $unwind : "$outstandingloans"},

{ $match: { gender: "male" } },

{ $group : { _id : "$name", numberOfLoans: {$sum:1}, totalAmount: {$sum:"$outstandingloans.balance"} }},

{ $match: { totalAmount: { $lt: -15000 }, numberOfLoans: {$gte: 3}}},

{$group : {_id : "maleDefaultersOver15000", amount: {$sum : 1}}}

]);

```
> db.defaulters.aggregate([
... { $unwind : "$outstandingloans"},
... { $match: { gender: "male" } },
... { $group : { _id : "$name", numberOfLoans: {$sum:1}, totalAmount: {$sum:"$outstandingloans.balance"} }},
... { $match: { totalAmount: { $lt: -15000 }, numberOfLoans: {$gte: 3}}},
... {$group : {_id : "maleDefaultersOver15000", amount: {$sum : 1}}}
... ]);
{ "_id" : "maleDefaultersOver15000", "amount" : 6 }
> 
```

MATCH (b:Borrower)--(o:Outstanding)

WHERE b.gender = "male" AND o.firstquarter < 0 AND o.secondquarter < 0 AND o.thirdquarter < 0

WITH b.name AS name, sum(o.firstquarter + o.secondquarter + o.thirdquarter + o.fourthquarter) AS totalAmount

WHERE totalAmount < -15000

RETURN COUNT(name) AS `Number of male loan defaulters who have 3 loans or more have defaulted on

loans in total valued over 15,000`