

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	2
CREATE NODES.....	2
SETUP CLUSTER	2
INSERT COLLECTION.....	2
SHARD COLLECTION	Error! Bookmark not defined.
PART 2	3
Installing APOC.....	3
Creating “defaulters” collection.....	4
Importing json file	4
Viewing mongodb database through APOC.....	4
Creating nodes and relationships	5
Viewing the nodes in the graph	6
Part 3.....	6
Are males or females on average bigger defaulters on their loans ?	6
What number of male loan defaulters who have 3 loans or more have defaulted on	7

PART 1

CREATE NODES

```
2020-04-09T06:37:01.934+0100 I COMMAND [conn1] command admin.$cmd command: getLastError { getLastError: 1, fsync: 1 } noreturn:1 keyUpdates:0 writeConflicts:0 numFields:0 reslen:140 locks:{ 216ms
2020-04-09T06:37:01.935+0100 I COMMAND [conn1] update config.lockpings query: ( { _id: "LAPTOP-F11C05QJ.26061:1586410038:41" } ) update: ( $set: { ping: new Date(1586410621881) } ) ns:config:1 ns:namespaces:1 nma
rched:1 nmodified:1 keyUpdates:1 writeConflicts:0 numFields:1 locks:{ Global: { acquireCount: { r: 3, w: 3 } }, MMAPV1Journal: { acquireCount: { w: 3 } }, acquireWaitCount: { w: 1 } }, timeAcquiringMicros: { w: 2874
} }, Database: { acquireCount: { w: 3 } }, Collection: { acquireCount: { w: 3 } } } 183ms
2020-04-09T06:37:02.032+0100 I SHARDING [LockPinger] cluster LAPTOP-F11C05QJ.local:26050,LAPTOP-F11C05QJ.local:26051,LAPTOP-F11C05QJ.local:26052,LAPTOP-F11C05QJ.local:26061:1586410038:41', sleeping for 30000ms
2020-04-09T06:37:02.066+0100 I SHARDING [LockPinger] cluster LAPTOP-F11C05QJ.local:26050,LAPTOP-F11C05QJ.local:26051,LAPTOP-F11C05QJ.local:26052,LAPTOP-F11C05QJ.local:26061:1586410038:41', sleeping for 30000ms
2020-04-09T06:37:02.136+0100 I COMMAND [conn1] command admin.$cmd command: getLastError { getLastError: 1, fsync: 1 } noreturn:1 keyUpdates:0 writeConflicts:0 numFields:0 reslen:122 locks:{ 103ms
2020-04-09T06:37:02.169+0100 I SHARDING [LockPinger] cluster LAPTOP-F11C05QJ.local:26050,LAPTOP-F11C05QJ.local:26051,LAPTOP-F11C05QJ.local:26052,LAPTOP-F11C05QJ.local:26061:1586410038:41', sleeping for 30000ms
2020-04-09T06:37:02.588+0100 I SHARDING [Balancer] distributed lock 'balancer/LAPTOP-F11C05QJ.local:26052/LAPTOP-F11C05QJ.local:26061:1586410038:41' acquired, ts : 5e8eb47e2c8a4d3053f62782
2020-04-09T06:37:02.589+0100 I STORAGE [conn32] CMD fsync: sync:1 lock:0
2020-04-09T06:37:02.589+0100 I STORAGE [conn32] CMD fsync: sync:1 lock:0
2020-04-09T06:37:02.692+0100 I COMMAND [conn32] command admin.$cmd command: fsync { fsync: true } noreturn:1 keyUpdates:0 writeConflicts:0 numFields:0 reslen:51 locks:{ Global: { acquireCount: { r: 3, w: 1 } }
, MMAPV1Journal: { acquireCount: { w: 2 } } } 102ms
2020-04-09T06:37:02.704+0100 I COMMAND [conn32] command admin.$cmd command: fsync { fsync: true } noreturn:1 keyUpdates:0 writeConflicts:0 numFields:0 reslen:51 locks:{ Global: { acquireCount: { r: 3, w: 1 } }
, MMAPV1Journal: { acquireCount: { w: 2 } } } 115ms
2020-04-09T06:37:02.704+0100 I COMMAND [conn32] command admin.$cmd command: fsync { fsync: true } noreturn:1 keyUpdates:0 writeConflicts:0 numFields:0 reslen:51 locks:{ Global: { acquireCount: { r: 3, w: 1 } }
, MMAPV1Journal: { acquireCount: { w: 2 } } } 115ms
2020-04-09T06:37:02.813+0100 I SHARDING [Balancer] distributed lock 'balancer/LAPTOP-F11C05QJ.local:26062:1586410038:41' unlocked.
2020-04-09T06:37:03.101+0100 I SHARDING [Balancer] distributed lock 'balancer/LAPTOP-F11C05QJ.local:26062:1586410038:41' acquired, ts : 5e8eb47e178ba43ddf846ff
2020-04-09T06:37:03.102+0100 I STORAGE [conn31] CMD fsync: sync:1 lock:0
2020-04-09T06:37:03.102+0100 I STORAGE [conn31] CMD fsync: sync:1 lock:0
2020-04-09T06:37:03.224+0100 I COMMAND [conn31] command admin.$cmd command: fsync { fsync: true } noreturn:1 keyUpdates:0 writeConflicts:0 numFields:0 reslen:51 locks:{ Global: { acquireCount: { r: 3, w: 1 } }
, MMAPV1Journal: { acquireCount: { w: 2 } } } 121ms
2020-04-09T06:37:03.236+0100 I COMMAND [conn31] command admin.$cmd command: fsync { fsync: true } noreturn:1 keyUpdates:0 writeConflicts:0 numFields:0 reslen:51 locks:{ Global: { acquireCount: { r: 3, w: 1 } }
, MMAPV1Journal: { acquireCount: { w: 2 } } } 133ms
2020-04-09T06:37:03.249+0100 I COMMAND [conn31] command admin.$cmd command: fsync { fsync: true } noreturn:1 keyUpdates:0 writeConflicts:0 numFields:0 reslen:51 locks:{ Global: { acquireCount: { r: 3, w: 1 } }
, MMAPV1Journal: { acquireCount: { w: 2 } } } 146ms
2020-04-09T06:37:03.299+0100 I SHARDING [Balancer] distributed lock 'balancer/LAPTOP-F11C05QJ.local:26062:1586410038:41' unlocked.
2020-04-09T06:37:03.994+0100 I NETWORK [conn16] end connection 192.168.103.50:63896 (5 connections now open)
2020-04-09T06:37:03.996+0100 I NETWORK [initandlisten] connection accepted from 192.168.103.50:63921 #18 (6 connections now open)
2020-04-09T06:37:04.184+0100 I NETWORK [conn21] end connection 192.168.103.50:63897 (9 connections now open)
2020-04-09T06:37:04.184+0100 I NETWORK [initandlisten] connection accepted from 192.168.103.50:63922 #23 (11 connections now open)
2020-04-09T06:37:04.197+0100 I NETWORK [conn17] end connection 192.168.103.50:63898 (5 connections now open)
2020-04-09T06:37:04.198+0100 I NETWORK [initandlisten] connection accepted from 192.168.103.50:63923 #19 (7 connections now open)
2020-04-09T06:37:04.221+0100 I NETWORK [conn22] end connection 192.168.103.50:63899 (9 connections now open)
2020-04-09T06:37:04.292+0100 I NETWORK [initandlisten] connection accepted from 192.168.103.50:63924 #24 (11 connections now open)
2020-04-09T06:37:04.298+0100 I NETWORK [conn16] end connection 192.168.103.50:63900 (5 connections now open)
2020-04-09T06:37:04.299+0100 I NETWORK [initandlisten] connection accepted from 192.168.103.50:63925 #18 (6 connections now open)
2020-04-09T06:37:04.303+0100 I SHARDING [Balancer] distributed lock 'balancer/LAPTOP-F11C05QJ.local:26063:1586410038:41' acquired, ts : 5e8eb480826d1bbcF96f63b1
2020-04-09T06:37:04.304+0100 I STORAGE [conn29] CMD fsync: sync:1 lock:0
2020-04-09T06:37:04.304+0100 I STORAGE [conn29] CMD fsync: sync:1 lock:0
2020-04-09T06:37:05.495+0100 I COMMAND [conn29] command admin.$cmd command: fsync { fsync: true } noreturn:1 keyUpdates:0 writeConflicts:0 numFields:0 reslen:51 locks:{ Global: { acquireCount: { r: 3, w: 1 } }
, MMAPV1Journal: { acquireCount: { w: 2 } } } 146ms
```

SETUP CLUSTER

```
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>mongo --shell 2.setup_cluster.js
MongoDB shell version: 3.0.15
connecting to: test
type "help" for help
connecting to: LAPTOP-F11C05QJ.local:27000/test
Dublin Replica Set Created!
Dublin Replica Set Up!
Dublin Shard Added!
connecting to: LAPTOP-F11C05QJ.local:27100/test
Cork Replica Set Created!
Cork Replica Set Up!
Adding Shard Failed. Trying it again
Cork Shard Added!
connecting to: LAPTOP-F11C05QJ.local:27200/test
Limerick Replica Set Created!
Limerick Replica Set Up!
Limerick Shard Added!
connecting to: LAPTOP-F11C05QJ.local:27300/test
Galway Replica Set Created!
Galway Replica Set Up!
Adding Shard Failed. Trying it again
Adding Shard Failed. Trying it again
Galway Shard Added!
C:\Users\rosha\OneDrive - mycit.ie\NoSQL Data Architectures\Assignment 2>pause
Press any key to continue . . .
```

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

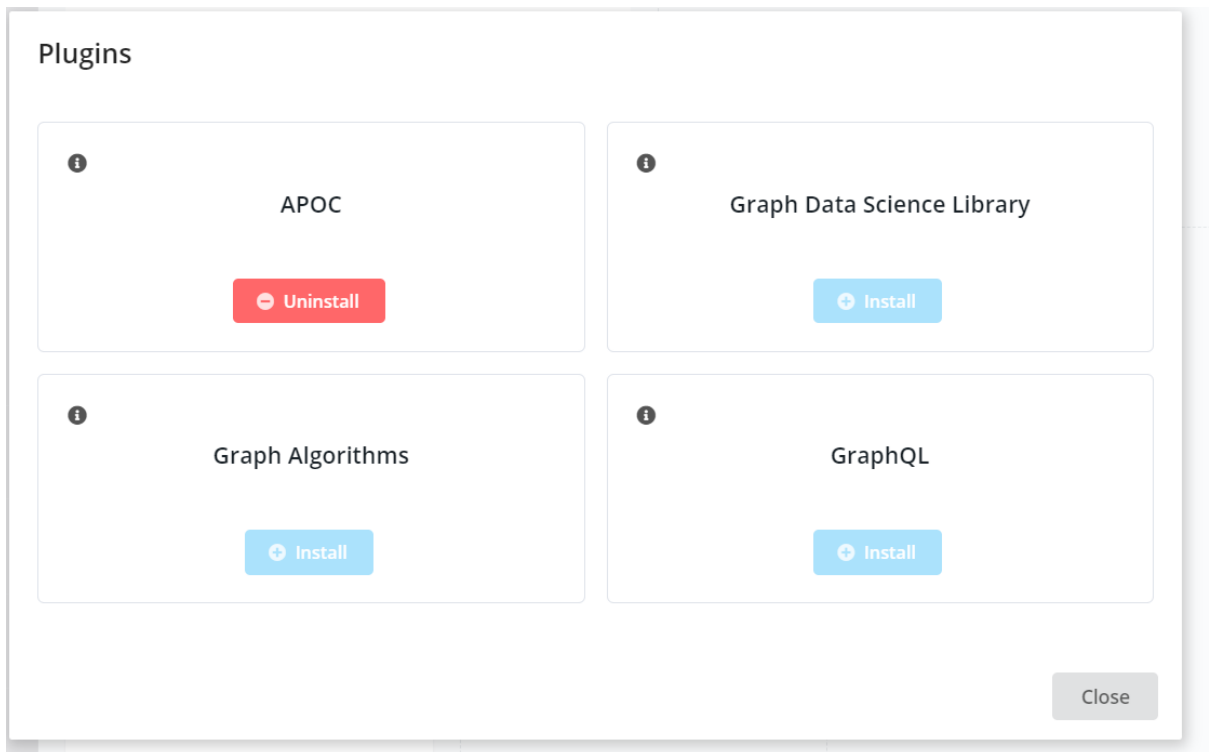
--- Sharding Status ---
  sharding version: {
    " _id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("5e8eb248826d1bbcf96f6370")
  }
  shards:
    { " _id" : "cork", "host" : "cork/LAPTOP-F1IC05QJ.local:27100,LAPTOP-F1IC05QJ.local:27101,LAPTOP-F1IC05QJ.local:27102" }
    { " _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)

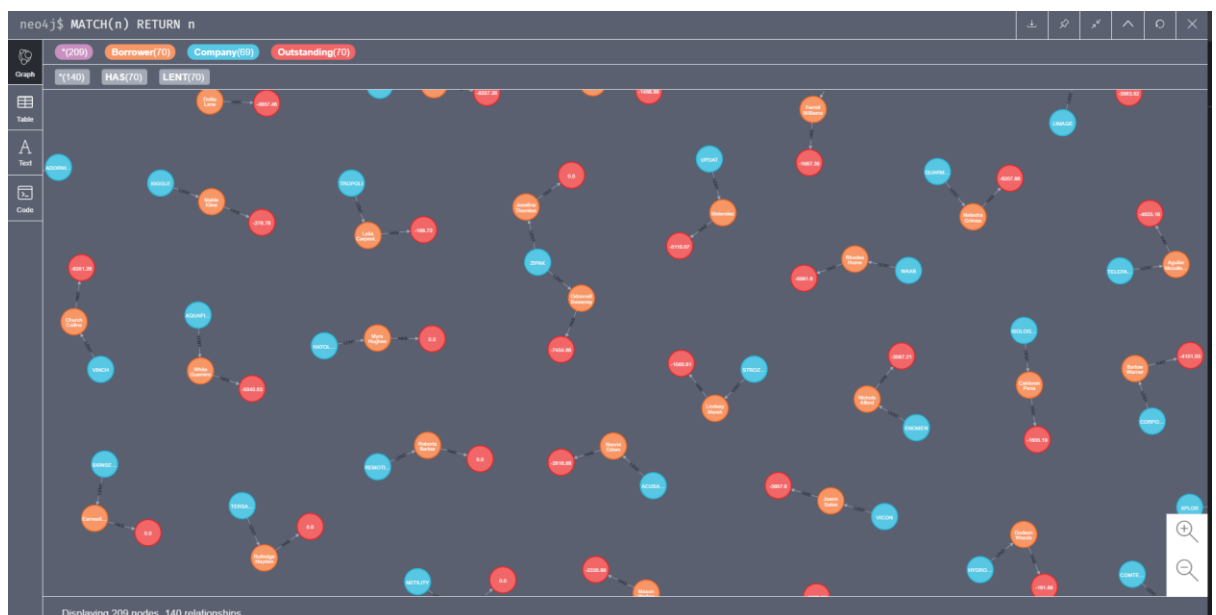
```
neo4j$ CALL apoc.mongodb.get('mongodb://localhost:27017', 'test', 'defaulters', {}) yield value
1 CALL apoc.mongodb.get('mongodb://localhost:27017', 'test', 'defaulters', {}) yield value
2 CREATE (b: Borrower
3 {
4   address : value.address,
5   gender : value.gender,
6   registered : value.registered,
7   isActive : value.isActive,
```

neo4j\$ CALL apoc.mongodb.get('mongodb://localhost:27017', 'test', 'defaulters', {}) yield value CREATE (b: Borrower { address...

Added 209 labels, created 209 nodes, set 979 properties, created 140 relationships, completed after 737 ms.

Added 209 labels, created 209 nodes, set 979 properties, created 140 relationships, completed after 737 ms.

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 }

  })

```

```

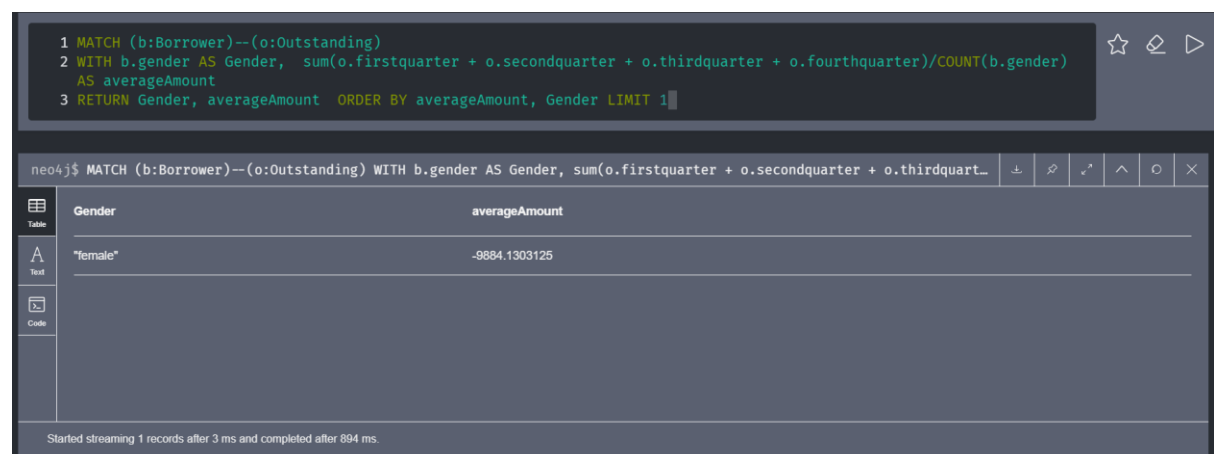
> db.defaulters.aggregate([
...   { $unwind : "$outstandingloans" },
...   { $group : { _id : { _id: "$_id", gender: "$gender", name: "$name", age: "$age"}, totalDebt : { $sum : "$outstandingloans.balance" } } },
...   { $group : { _id : "$_id.gender", averageDebt : { $avg : "$totalDebt" } } },
...   { $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



The screenshot shows the Neo4j Cypher query editor and results viewer. The query is:

```

1 MATCH (b:Borrower)--(o:Outstanding)
2 WITH b.gender AS Gender, sum(o.firstquarter + o.secondquarter + o.thirdquarter + o.fourthquarter)/COUNT(b.gender)
  AS averageAmount
3 RETURN Gender, averageAmount ORDER BY averageAmount, Gender LIMIT 1

```

The results are displayed in a table with two columns: Gender and averageAmount.

Gender	averageAmount
"female"	-9884.1303125

At the bottom, it states: "Started streaming 1 records after 3 ms and completed after 894 ms."

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`

The screenshot shows a Neo4j Cypher query editor with the following query:

```
1 MATCH (b:Borrower)--(o:Outstanding)  
2 WHERE b.gender = "male" AND o.firstquarter < 0 AND o.secondquarter < 0 AND o.thirdquarter < 0  
3 WITH b.name AS name, sum(o.firstquarter + o.secondquarter + o.thirdquarter + o.fourthquarter) AS totalAmount  
4 WHERE totalAmount < -15000  
5 RETURN COUNT(name) AS `Number of male loan defaulters who have 3 loans or more have defaulted on`  
6 loans in total valued over 15,000`
```

Below the query editor, the results viewer shows a table with the following data:

Count
6

The status bar at the bottom indicates: "Started streaming 1 records after 1 ms and completed after 4 ms."