

**M2 - MIAGE 2IS - Innovative Information Systems**

**Innovative Data Management**

**Term project – Project IDM**

**Professors**: THEVENIN Jean-Marc,

ALIGON Julien

**Students:** KIM Phuong Pham, JIN Xutong

Toulouse, 2019

Table of Contents

[1. Analysis of Data Stored in a Text Form 3](#_Toc1252342)

[1.1 PIG: Count the number of people with income <= 50K and> 50K 3](#_Toc1252343)

[1.1.1 By Age 3](#_Toc1252344)

[1.1.2 By Sex 3](#_Toc1252345)

[1.1.3 By Workclass 4](#_Toc1252346)

[1.2 MapReduce: Count the number of people with income <= 50K and> 50K 4](#_Toc1252347)

[1.1.1 By Age 4](#_Toc1252348)

[1.1.2 By Sex 4](#_Toc1252349)

[1.1.3 By Workclass 4](#_Toc1252350)

[1.3 Sqoop and Hive 5](#_Toc1252351)

[1.3.1 Import data from MySQL to HDFS 5](#_Toc1252352)

[1.3.2 By Age 6](#_Toc1252353)

[1.3.2 By Sex 6](#_Toc1252354)

[1.3.2 By Workclass 7](#_Toc1252355)

[2. Discovering Neo4j 7](#_Toc1252356)

[2.1 Load the Movie Database 7](#_Toc1252357)

[2.2 Count the Number of Actors and Movies in the Database 8](#_Toc1252358)

[2.3 Queries in Cypher 9](#_Toc1252359)

[2.3.1 View actors who played directly with Tom Cruise 9](#_Toc1252360)

[2.3.2 Number of actors who played directly with Tom Cruise 9](#_Toc1252361)

[2.3.3 View actors who have played with actors who have already played with Tom Cruise 10](#_Toc1252362)

[2.3.5 Show for each director the number of actors he directed in movies 10](#_Toc1252363)

[2.3.6 Show for each director the average number of actors he directed by movie 11](#_Toc1252364)

# 1. Analysis of Data Stored in a Text Form

First of all, we converted our dataset into csv file. To simplify our dataset, we modified “income” column: Replaced string “<=50k” by integer “1”, and string “>50k” by integer “2”.

## 1.1 PIG: Count the number of people with income <= 50K and> 50K

* Copy file from local

hadoop fs -copyFromLocal adult.data /user/Phuong/DB\_Adult/adult.data

* Load datafile into table “ad”, separate data by “,” and specify the schema by “(age: int,…)”

ad = load '/user/Phuong/DB\_Adult/adult.data' using PigStorage(',') as (age:int,workclass:chararray,fnlwgt:int,education:chararray,education\_num:int,marital\_status:chararray,occupation:chararray,relationship:chararray,race:chararray,sex:chararray,capital\_gain:int,capital\_loss:int,hours\_per\_week:int,native\_country:chararray,income:int);

### 1.1.1 By Age

* Sort data table by age and income, then store it as “ad\_by\_age”

ad\_by\_age = group ad by (age,income);

* Traverse “ad\_by\_age”, un-nest bags, group data by age, income (1 means <=50, 2 means >50), and count income number. Name this table as “ad\_by\_age\_count”

ad\_by\_age\_count = foreach ad\_by\_age generate FLATTEN(group) as (age,income),COUNT(ad.income) as income\_c;

* Data storage

store ad\_by\_age\_count into '/user/Phuong/DB\_Adult/income\_by\_age'



### 1.1.2 By Sex

* Sort data table by sex and income, then store it as “ad\_by\_sex”

ad\_by\_sex = group ad by (sex,income);

* Traverse “ad\_by\_sex”, un-nest bags, group data by sex, income and count income number. Name this table as “ad\_by\_sex\_count”

ad\_by\_sex\_count = foreach ad\_by\_sex generate FLATTEN(group) as (sex,income),COUNT(ad.income) as income\_c;

* Data storage

store ad\_by\_sex\_count into '/user/Phuong/DB\_Adult/income\_by\_sex'



### 1.1.3 By Workclass

* Sort data table by workclass and income, then store it as “ad\_by\_workclass”

ad\_by\_workclass = group ad by workclass,income);

* Traverse “ad\_by\_workclass”, un-nest bags, group data by workclass, income and count income number. Name this table as “ad\_by\_workclass\_count”

ad\_by\_workclass\_count = foreach ad\_by\_workclass generate FLATTEN(group) as (workclass,income),COUNT(ad.income) as income\_c;

* Data storage

store ad\_by\_workclass\_count into '/user/Phuong/DB\_Adult/income\_by\_workclass'



## 1.2 MapReduce: Count the number of people with income <= 50K and> 50K

### 1.1.1 By Age

Mapper: - Input: Text, Output: Map<Text, Int>

* Read a line and split it into words
* Identify the column (col 1: Age, col 14: Income)
* For each line:
  + If income <=50: Write a line to output <”Age,class 1”, 1)
  + If income >50: Write a line to output <”Age,class 2”, 1)

Reducer: Input (Text key, Iterable<Int> values), output (Map<Text, Int>)

* With each key: sum (count) the number of values and write to output

Code Result

### 1.1.2 By Sex

The same logic with “by age” function but in the mapper, split into Male, Class1; Male, Class 2 and Female, Class1; Female, Class2

Code Result

### 1.1.3 By Workclass

Mapper: - Input: Text, Output: Map<Text, Text>

* Read a line and split it into words
* Identify the column (col 2: Workclass, col 14: Income)
* For each line:
  + Write the couple of (key,value) = (workclass, income)

Reducer: Input (Text key, Iterable<Text> values), output (Map<Text, Text>)

* Define a Map<income\_class, count\_number of this class)
* For each income\_class in the array values: count the number of people in this class in the Map.
* With each key: Workclass, print out the map of Income\_class and its numbers of people.

Code Result

## 1.3 Sqoop and Hive

### 1.3.1 Import data from MySQL to HDFS

* Import csv file into etu-web2/phpMyAdmin database
* Testing in Hadoop until find the unbusy port

[cloudera@quickstart ~]$ nc -zv localhost 3307

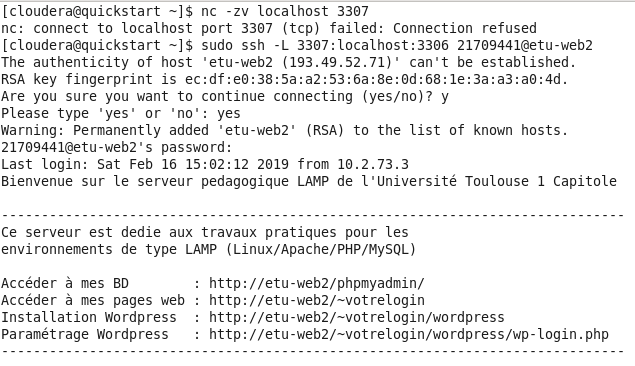


Figure 1 Free Port:3307

* Open a tunnel between the port 3307 of the PC and the MySQL server

[cloudera@quickstart ~]$ sudo ssh -L 3307:localhost:3306 21709441@etu-web2

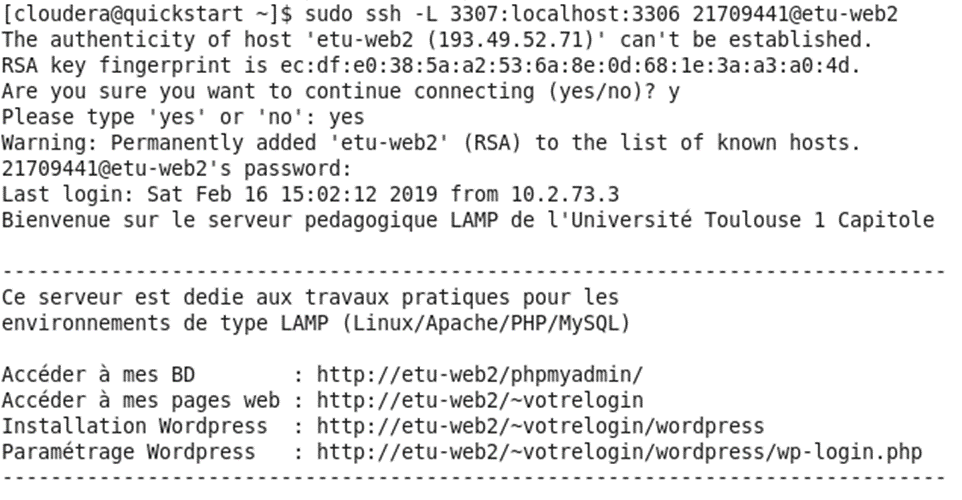


Figure 2 Open the Tunnel

* import data from MySQL into Hive

[cloudera@quickstart ~]$ sqoop import --connect jdbc:mysql://localhost:3307/db\_21709441 --username 21709441 --password S00AO4 --table adult --hive-import

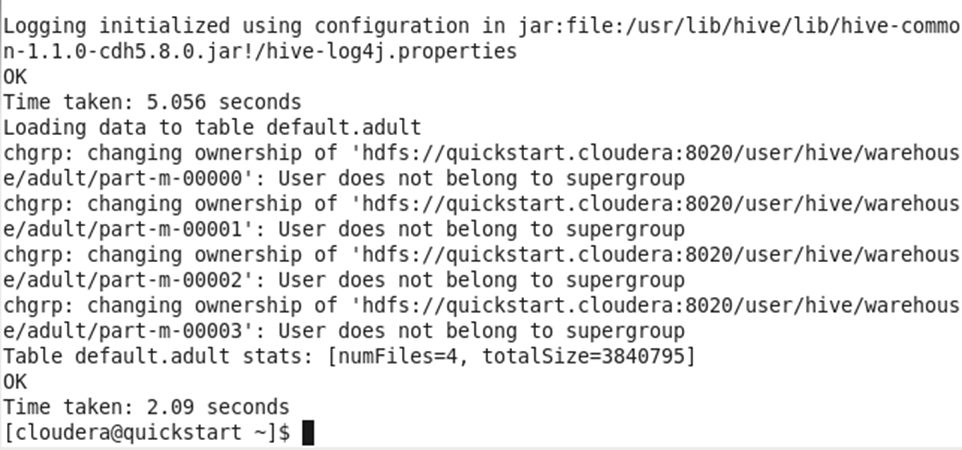


Figure 3 Data Importing

### 1.3.2 By Age

select count(\*) as numAge, age, income as income\_class

from adult

group by age,income

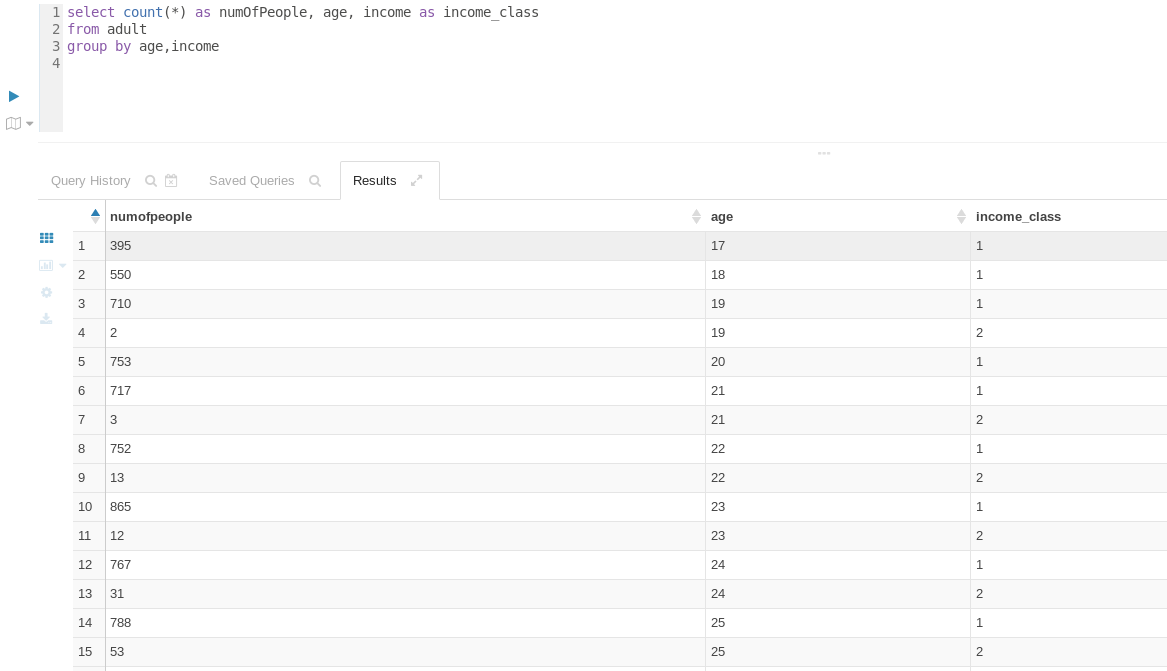


Figure 4 Hive: Income by Age

### 1.3.2 By Sex

select count(\*) as numSex, sex, income as income\_class

from adult

group by sex,income

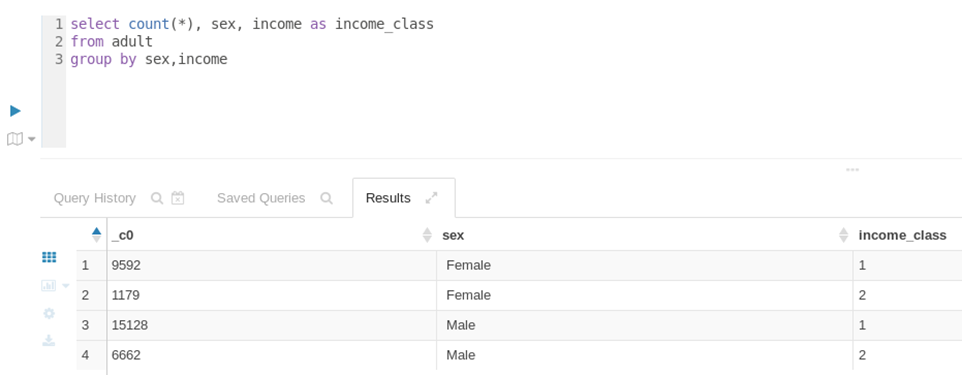


Figure 5 Hive: Income by Sex

### 1.3.2 By Workclass

select count(\*) as numWorkclass, workclass, income as income\_class

from adult

group by workclass,income

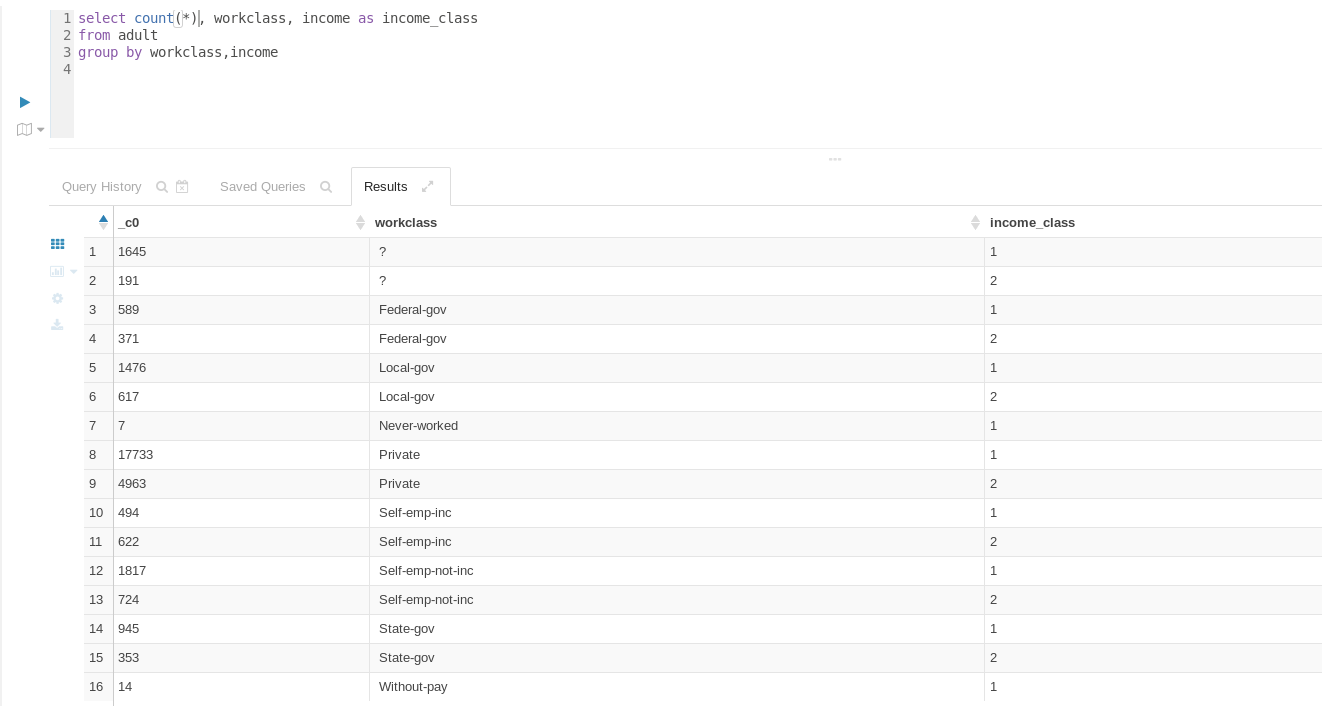


Figure 6 Income by Workclass

# 2. Discovering Neo4j

## 2.1 Load the Movie Database

:play movie graph

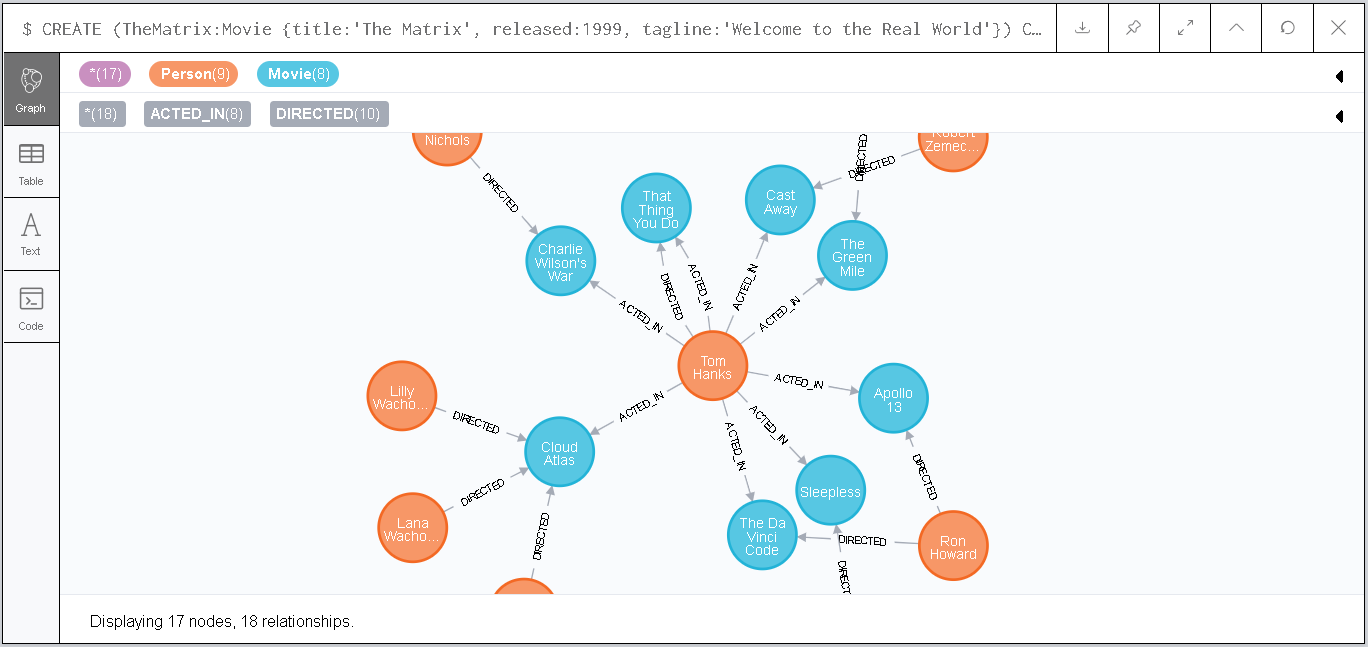


Figure 7 The Movie Database

## 2.2 Count the Number of Actors and Movies in the Database

MATCH (movie:Movie) RETURN COUNT(DISTINCT(movie))

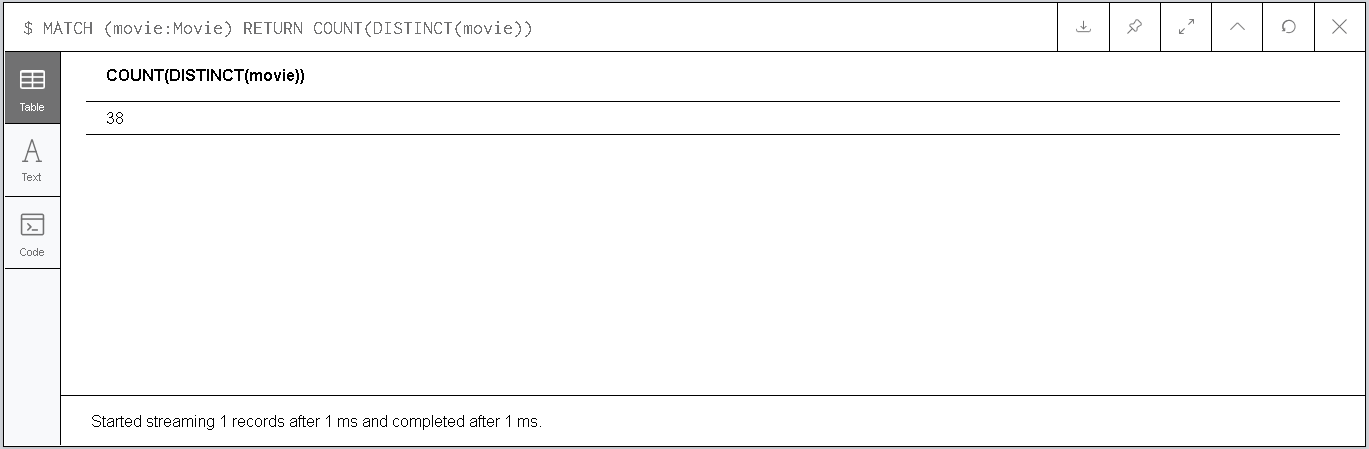


Figure 8 Number of Movies

MATCH (actor:Person)-[r:ACTED\_IN]->() RETURN COUNT(DISTINCT(actor.name))

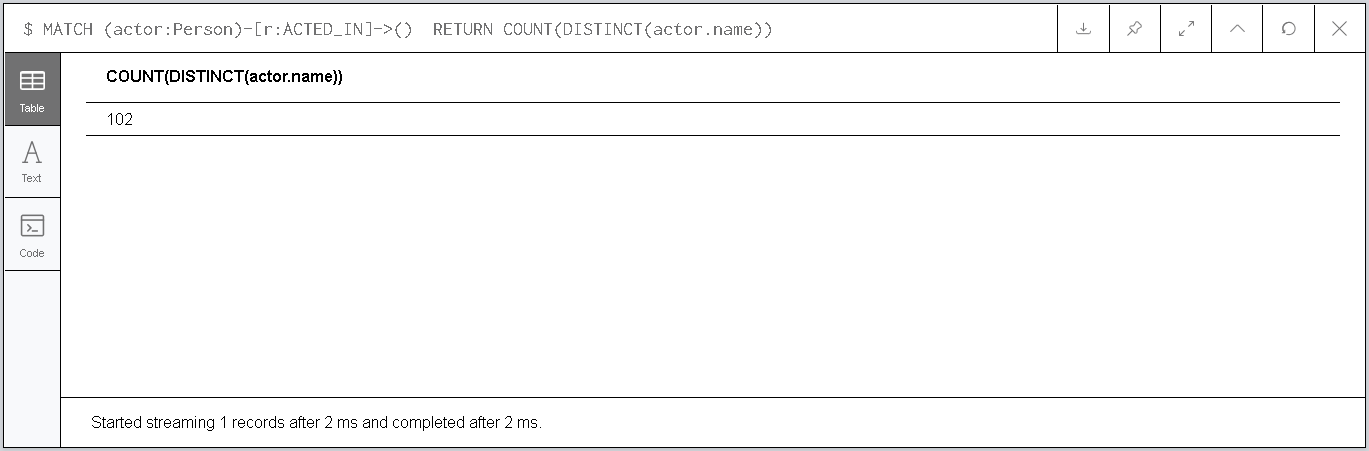


Figure 9 Number of Actors

## 2.3 Queries in Cypher

### 2.3.1 View actors who played directly with Tom Cruise

MATCH (TomC:Person {name:"Tom Cruise"})-[:ACTED\_IN]->(m)<-[:ACTED\_IN]-(coActors)

WHERE TomC <> coActors

RETURN DISTINCT(coActors)

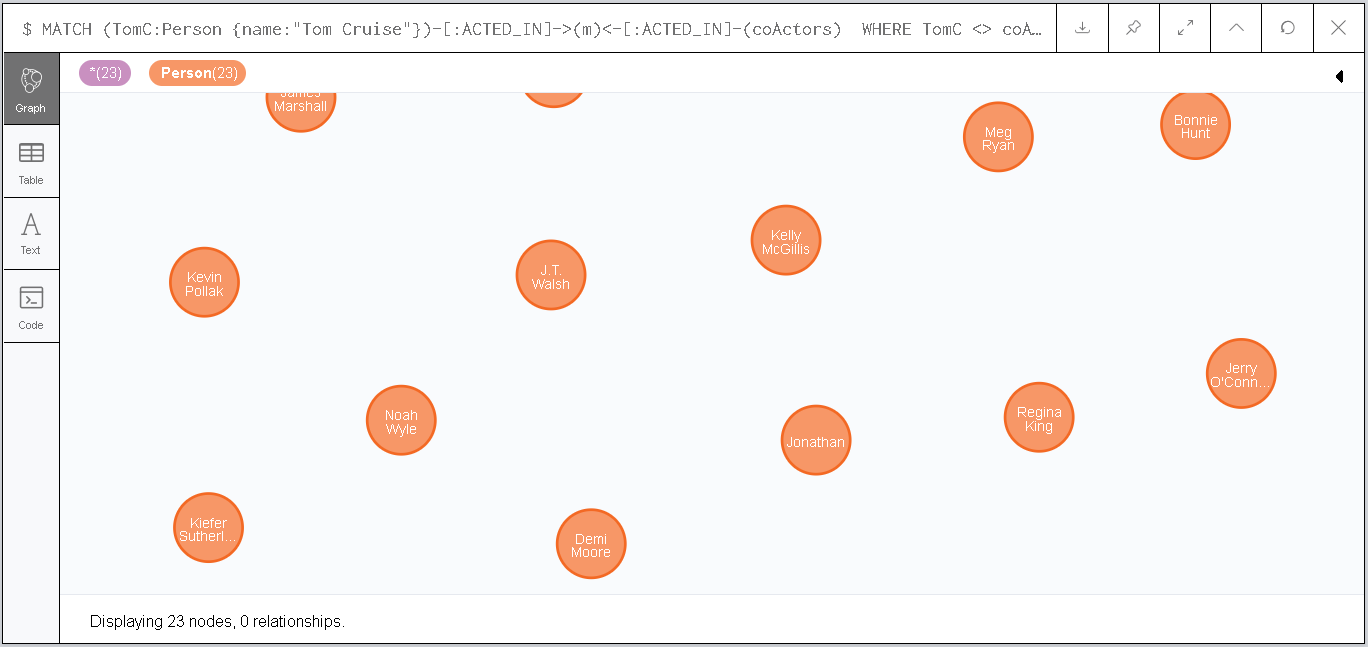


Figure 10 Co-actors of Tom Cruise

### 2.3.2 Number of actors who played directly with Tom Cruise

MATCH (TomC:Person {name:"Tom Cruise"})-[:ACTED\_IN]->(m)<-[:ACTED\_IN]-(coActors)

WHERE TomC <> coActors

RETURN COUNT(DISTINCT(coActors))

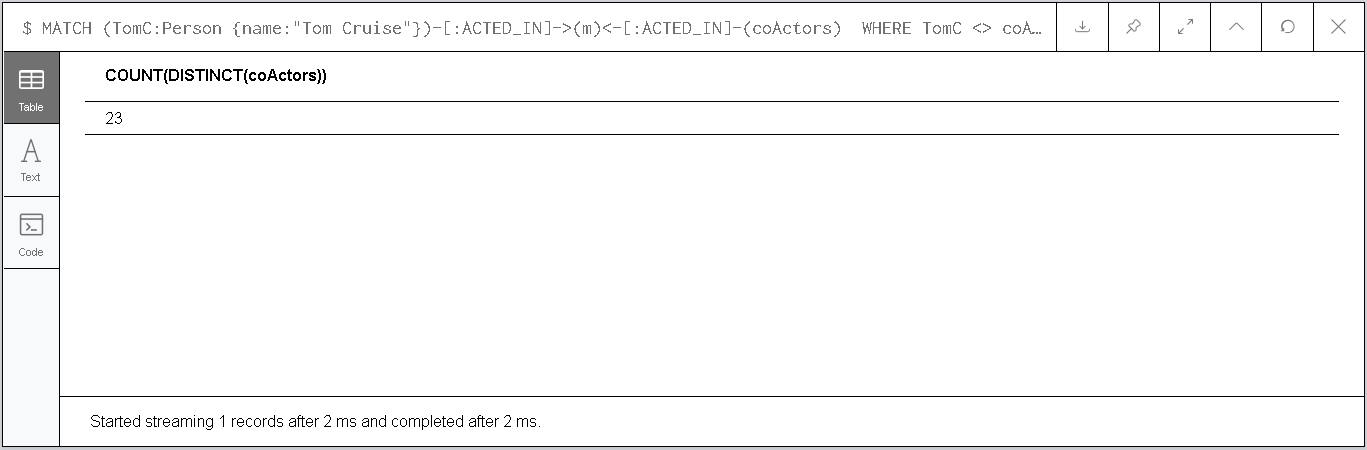


Figure 11 Number of Co-actors of Tome Cruise

### 2.3.3 View actors who have played with actors who have already played with Tom Cruise

MATCH (TomC:Person {name:"Tom Cruise"})-[:ACTED\_IN]->(m)<-[:ACTED\_IN]-(coActors), (coActors)-[:ACTE D\_IN]->(m2)<-[:ACTED\_IN]-(cocoActors)

WHERE TomC <> cocoActors

RETURN DISTINCT(cocoActors)

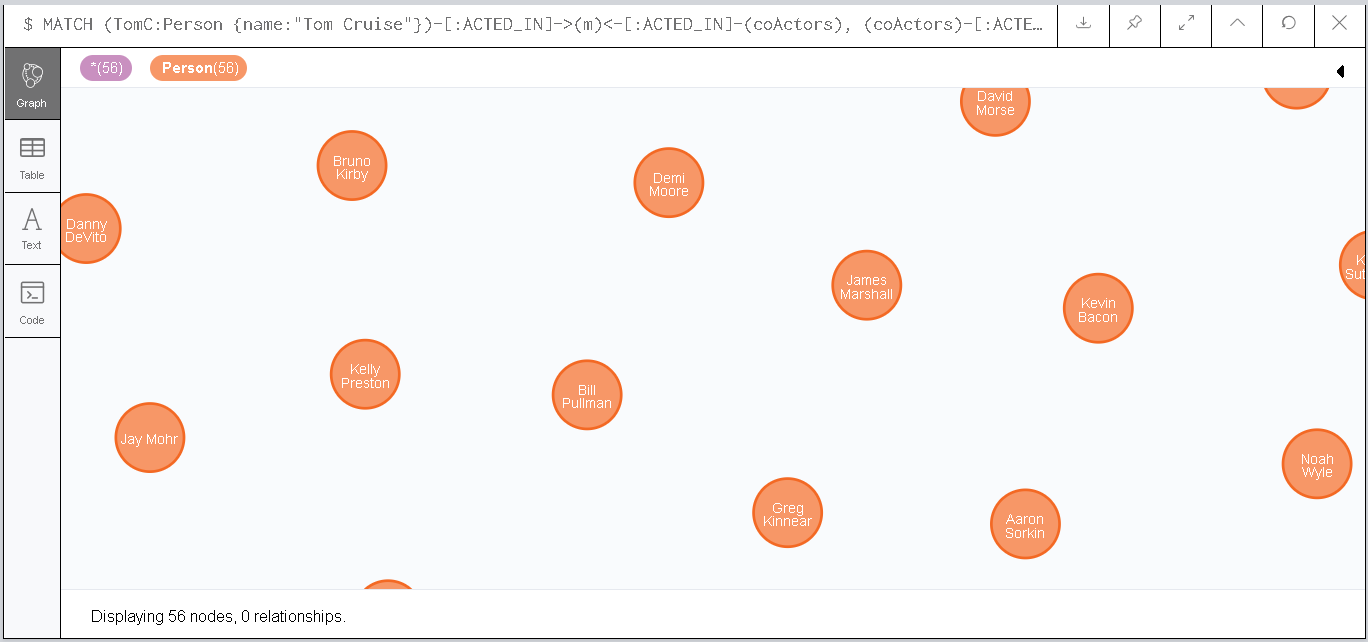


Figure 12 Co-coactors of Tom Cruise

### 2.3.5 Show for each director the number of actors he directed in movies

MATCH (director:Person)-[:DIRECTED]->() <-[:ACTED\_IN]-(Actors) RETURN director.name,COUNT(DISTINCT(Actors))

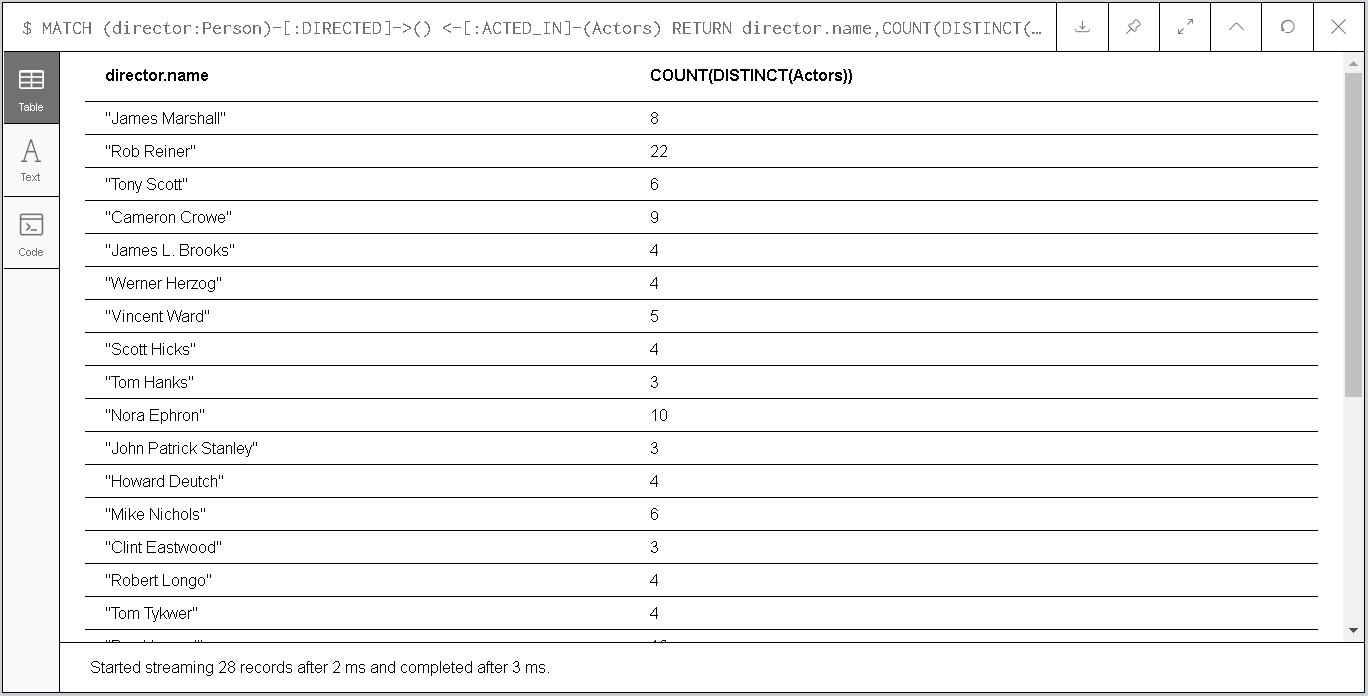


Figure 13 Number of Actors Directed by Each Director

### 2.3.6 Show for each director the average number of actors he directed by movie

MATCH (director:Person)-[:DIRECTED]->(movie:Movie)

WITH director, movie

MATCH (director:Person)-[:DIRECTED]->(movie)<-[:ACTED\_IN]-(actors:Person)

WITH director,movie,COUNT(actors) AS numAct

RETURN director.name, numAct/COUNT(movie) AS avgActMov

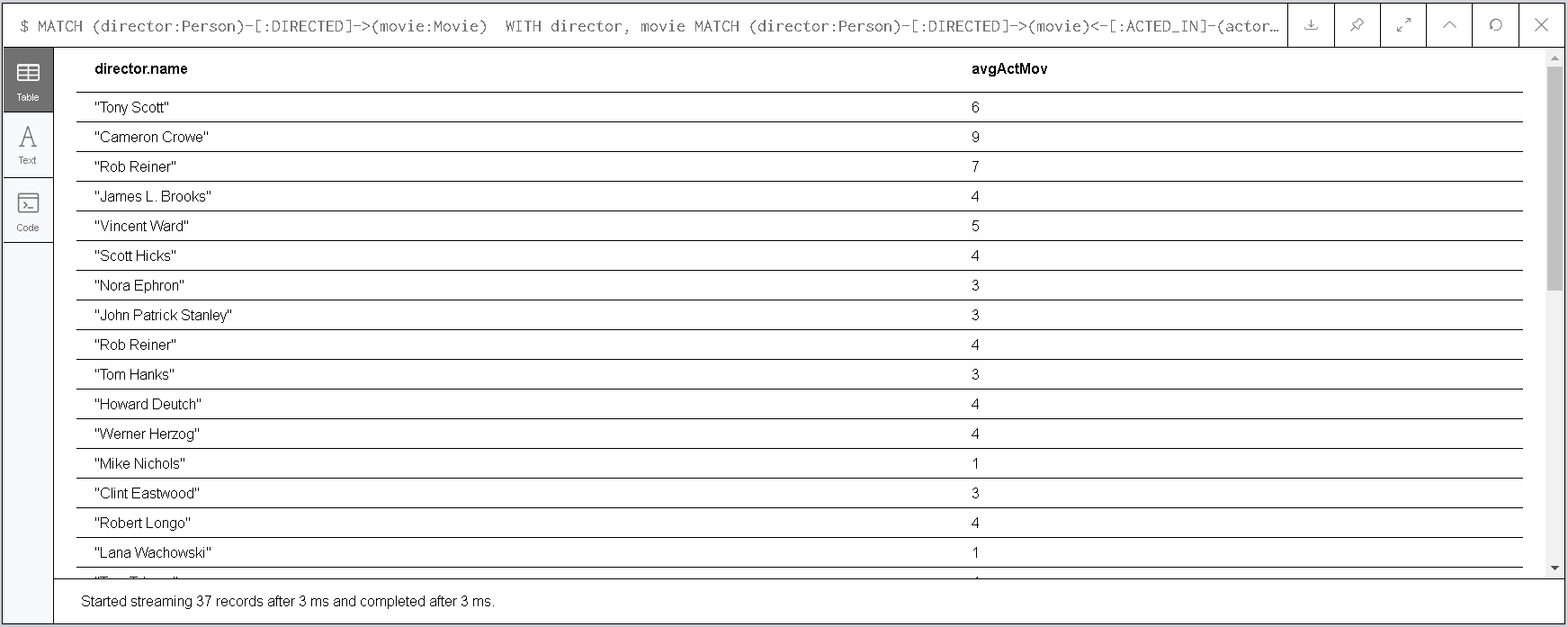


Figure 14 For Each Director, Average Number of Actors Directed by Movie