	Hope Foundation's **Finolex Academy of Management and Technology, Ratnagiri** **Information Technology Department**			
Subject name: Big Data Analytics				Subject Code: ITL801
Class	BE IT	Semester – VIII (CBGS)		Academic year: 2019-20
Name of Student	**Kazi Jawwad A Rahim**		**QUIZ Score :**	
Roll No	**28**	Assignment/Experiment No.		03

Title: Creating graph database on Neo4j and displaying the relationships of graph.

1. Course objectives applicable

COB3.Understand various concepts of finding similar items and mining data streams

COB4.Study Page Rank in Link Analysis and concepts of Handling larger datasets

COB5.Study various clustering algorithms and Recommendation systems.

2. Course outcomes applicable:

CO3 -Interpret business models and scientific computing paradigms, and apply software tools for big data analytics

CO4-Implement use of combiners to consolidate results and ability to handle larger datasets.

3. Learning Objectives:

1. Understand the Neo4j Database application domain
2. Install and run Neo4j on the machine
3. Create graph database and display the relations of graph
4. Analyze the queries for creating relationship between nodes

4. Practical applications of the assignment/experiment: Neo4J is a graph Database, used in relationship based analysis on social networks

5. Prerequisites:

1. Knowledge of SQL
2. Knowledge of Graph Databases

6. Hardware Requirements:

1. PC with 4GB RAM, 500GB HDD

7. Software Requirements:

1. Access to Neo4j Sandbox
2. Google chrome version 6.23 or higher

8. Quiz Questions (if any): (Online Exam will be taken separately batchwise, attach the certificate/ Marks obtained)

1. What is a Graph Database?
2. Which Query language does graph database use?
3. Can you install Neo4J on windows? If yes, explain the process.
4. What relations were plotted on the console ?

9. Experiment/Assignment Evaluation:

Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used -		6
2	Neatness/presentation		2
3	Punctuality		2

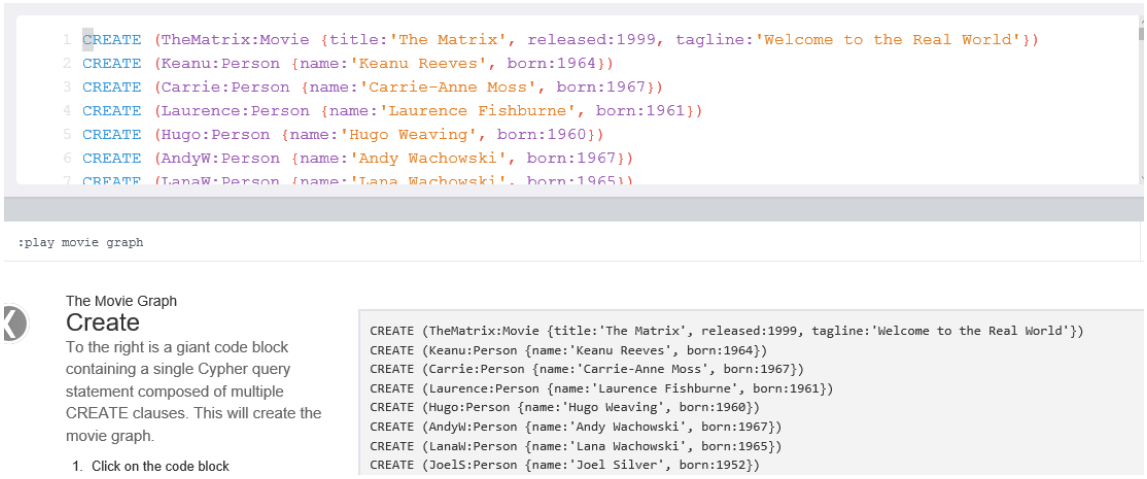
Date of performance (DOP)		**Total marks obtained**		**10**
Date of checking (DOC)		**Signature of teacher**		

2. Precautions :

1. Register on Neo4J first,
2. Request for crating sandbox, it will take time

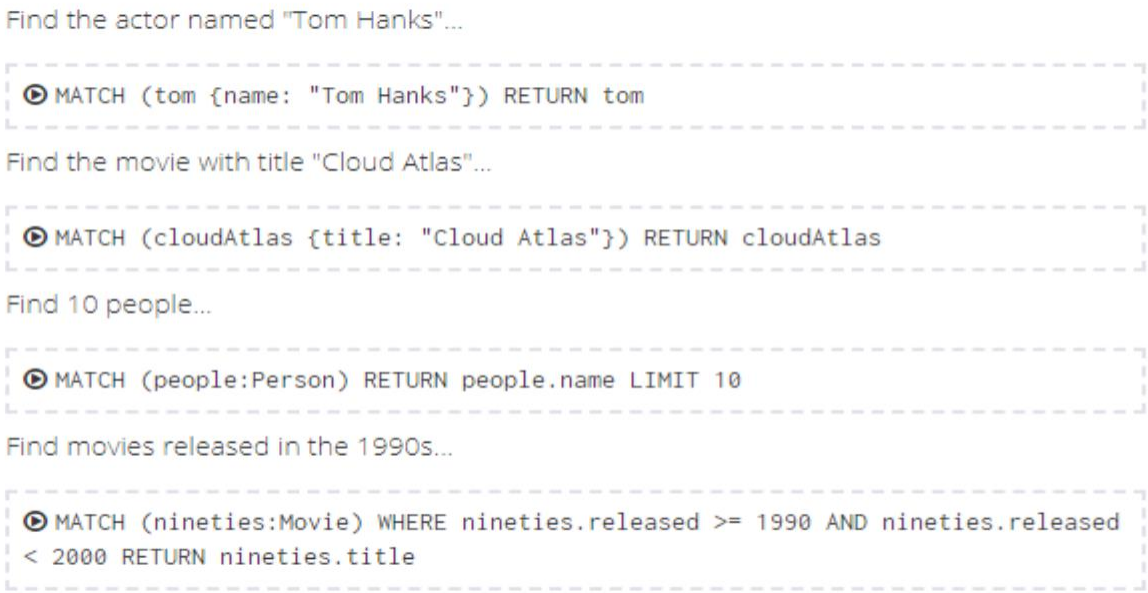
3. Installation Steps / Performance Steps -

1) Sign in to Neo4j Sandbox installer.
2) Run the cypher queries
3) View the database relationships:
4) NoSQL Graph database Neo4j (The Movie graph)



4.Observations

1)



2)

List all Tom Hanks movies...

```
MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]->(tomHanksMovies) RETUR
N tom,tomHanksMovies
```

Who directed "Cloud Atlas"?

```
MATCH (cloudAtlas {title: "Cloud Atlas"})<-[:DIRECTED]-(directors) RETURN
directors.name
```

Tom Hanks' co-actors...

```
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coAct
ors) RETURN coActors.name
```

How people are related to "Cloud Atlas"...

```
MATCH (people:Person)-[relatedTo]-(:Movie {title: "Cloud Atlas"}) RETURN p
eople.name, Type(relatedTo), relatedTo
```

3)

Movies and actors up to 4 "hops" away from Kevin Bacon

```
MATCH (bacon:Person {name:"Kevin Bacon"})-[*1..4]-(hollywood)
RETURN DISTINCT hollywood
```

Bacon path, the shortest path of any relationships to Meg Ryan

```
MATCH p=shortestPath(
  (bacon:Person {name:"Kevin Bacon"})-[*]-(meg:Person {name:"Meg Ryan"})
)
RETURN p
```

Note you only need to compare property values like this when first creating relationships

4)

Extend Tom Hanks co-actors, to find co-co-actors who haven't worked with Tom Hanks...

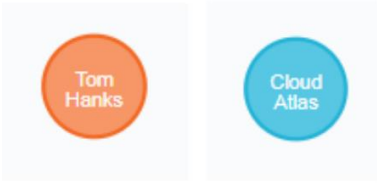
```
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coAct
ors),
      (coActors)-[:ACTED_IN]->(m2)<-[:ACTED_IN]-(cocoActors)
WHERE NOT (tom)-[:ACTED_IN]->()<-[:ACTED_IN]-(cocoActors) AND tom <> cocoActo
rs
RETURN cocoActors.name AS Recommended, count(*) AS Strength ORDER BY Strength
DESC
```

Find someone to introduce Tom Hanks to Tom Cruise

```
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coAct
ors),
      (coActors)-[:ACTED_IN]->(m2)<-[:ACTED_IN]-(cruise:Person {name:"Tom Cru
ise"})
RETURN tom, m, coActors, m2, cruise
```

5. Results:

Output of SET 1:



people.name
"Keanu Reeves"
"Keanu Reeves"
"Carrie-Anne Moss"
"Laurence Fishburne"
"Hugo Weaving"
"Lilly Wachowski"
"Lana Wachowski"
"Joel Silver"
"Emil Eifrem"
"Charlize Theron"

nineties.title
"The Matrix"
"The Matrix"
"The Devil's Advocate"
"A Few Good Men"
"As Good as It Gets"
"What Dreams May Come"
"Snow Falling on Cedars"
"You've Got Mail"
"Sleepless in Seattle"

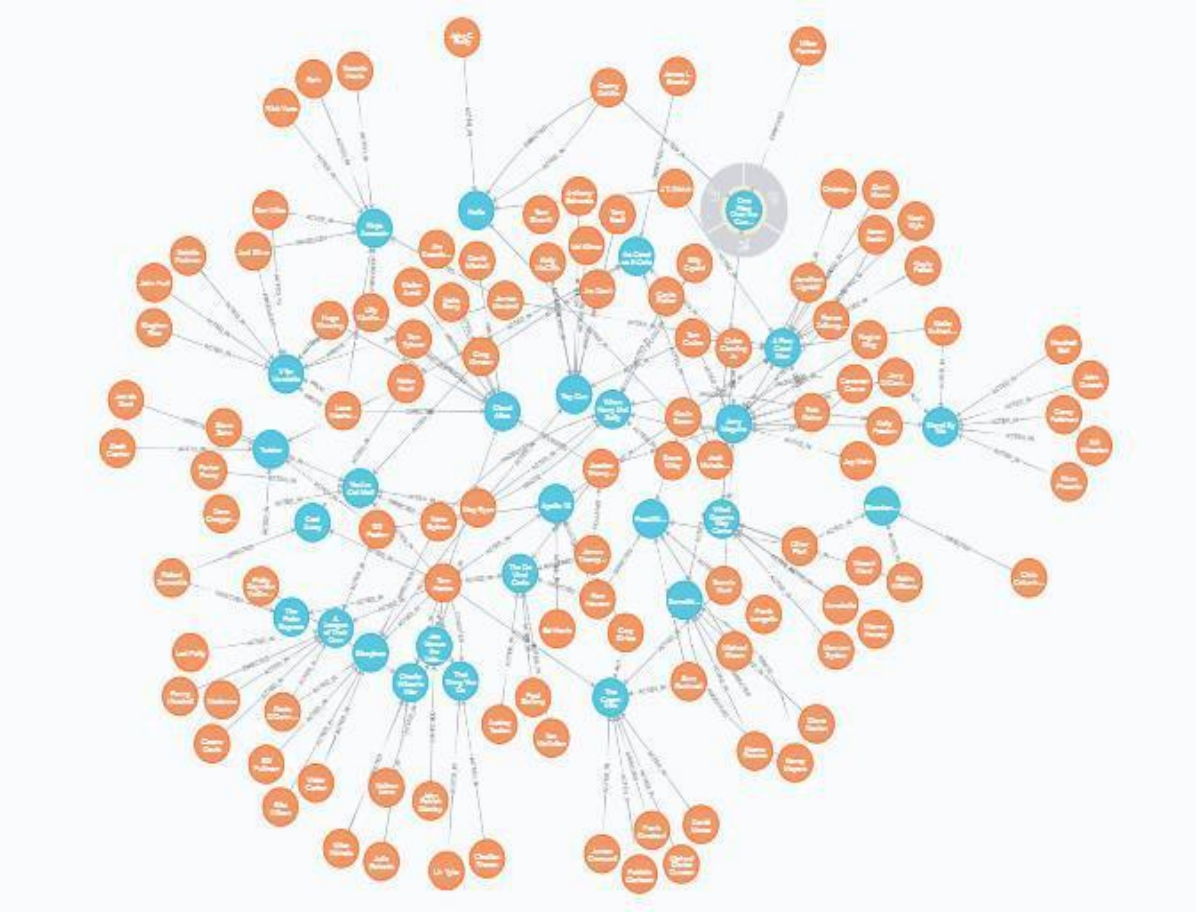
Output of SET 2:

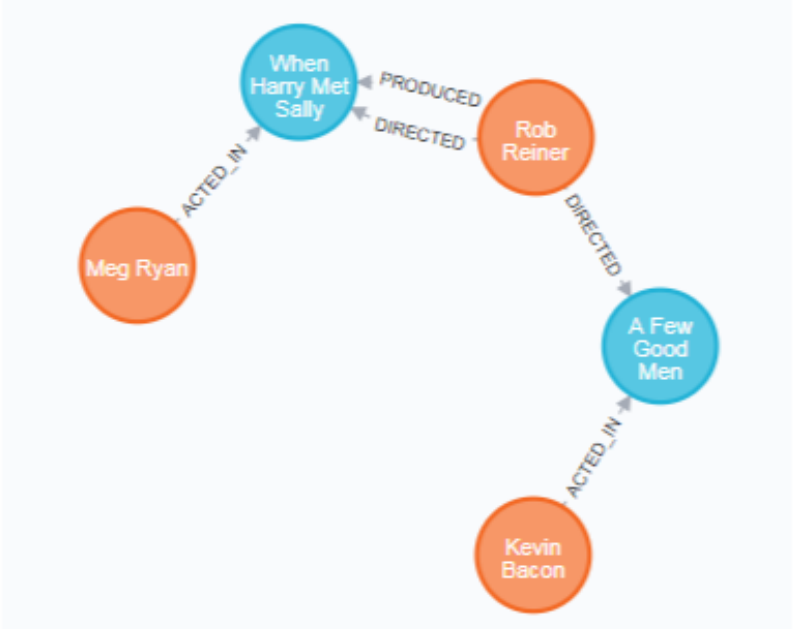
directors.name
"Lilly Wachowski"
"Tom Tykwer"
"Lana Wachowski"

coActors.name
"Sam Rockwell"
"David Morse"
"Patricia Clarkson"
"James Cromwell"
"Michael Clarke Duncan"
"Gary Sinise"
"Bonnie Hunt"
"Dave Chappelle"
"Meg Ryan"
"Greg Kinnear"
"Steve Zahn"
"Parker Posey"
"Paul Bettany"

people.name	Type(relatedTo)	relatedTo
"Lilly Wachowski"	"DIRECTED"	{ }
"Hugo Weaving"	"ACTED_IN"	{ "roles": ["Bill Smoke", "Haskell Moore", "Tadeusz Kesselring", "Nurse Noakes", "Boardman Mephi", "Old Georgie"] }
"Tom Hanks"	"ACTED_IN"	{ "roles": [

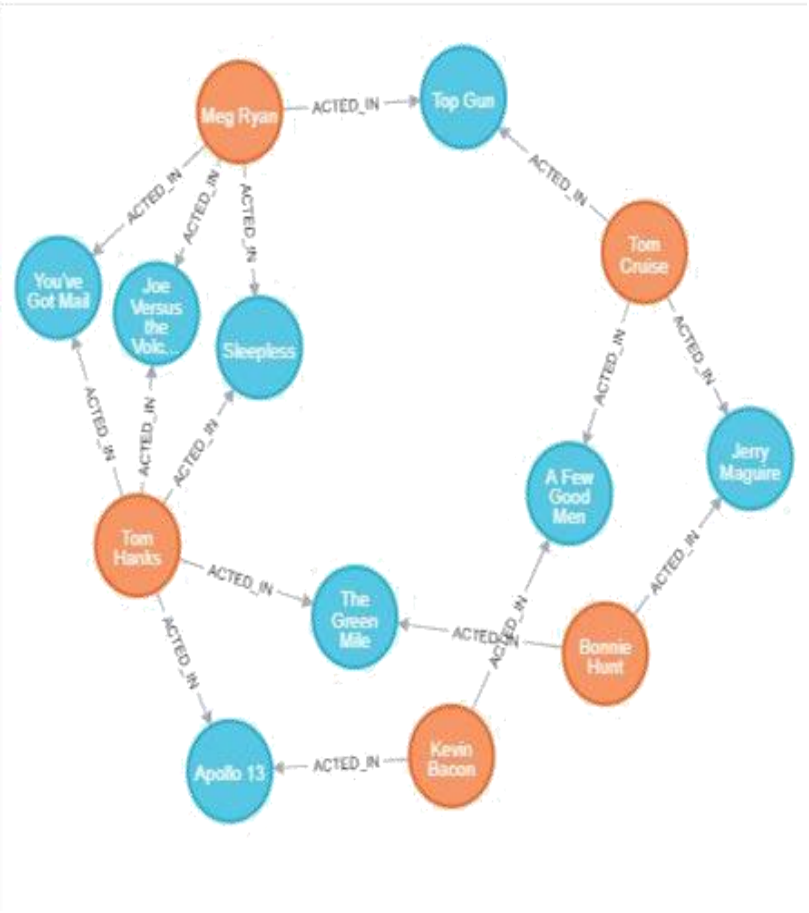
Output of SET 3:





Output of SET 4:

Recommended	Strength
"Tom Cruise"	5
"Zach Grenier"	5
"Cuba Gooding Jr."	4
"Keanu Reeves"	4
"Carrie Fisher"	3
"Billy Crystal"	3
"Bruno Kirby"	3
"Val Kilmer"	3
"Kelly McGillis"	3
"Tom Skerritt"	3
"Anthony Edwards"	3
"Jack Nicholson"	3
"Laurence Fishburne"	3
"Carrie-Anne Moss"	3
"Oliver Platt"	2
"Michael Sheen"	2
"Frank Langella"	2
"Max von Sydow"	1
"Rick Yune"	1



8. References :

[1] https://neo4j.com/developer/graph-database/
[2] http://whatis.techtarget.com/definition/graph-database