## HU Extension Assignment 10 E63 Big Data Analytics

### Handed out: 04/09/2016 Due by 11:30PM EST on Friday, 04/15/2016

Please, describe every step of your work and present all intermediate and final results in a Word document. Please, copy past text version of all essential command and snippets of results into the Word document with explanations of the purpose of those commands. We cannot retype text that is in JPG images. Please, always submit a separate copy of the original, working scripts and/or class files you used. Sometimes we need to run your code and retyping is too costly. Please include in your MS Word document only relevant portions of the console output or output files. Sometime either console output or the result file is too long and including it into the MS Word document makes that document too hard to read. PLEASE DO NOT EMBED files into your MS Word document. For issues and comments visit the class Discussion Board. You are not obliged to use Java or Eclipse. You are welcome to use any language and any IDE of your choice.

**Problem 1.** The following is the content ofMovies database.Bring that database into Neo4J using curl.

CREATE (matrix1:Movie { title : 'The Matrix', year : '1999-03-31' })

CREATE (matrix2:Movie { title : 'The Matrix Reloaded', year : '2003-05-07' })

CREATE (matrix3:Movie { title : 'The Matrix Revolutions', year : '2003-10-27' })

CREATE (keanu:Actor { name:'Keanu Reeves' })

CREATE (laurence:Actor { name:'Laurence Fishburne' })

CREATE (carrieanne:Actor { name:'Carrie-Anne Moss' })

CREATE (keanu)-[:ACTS\_IN { role : 'Neo' }]->(matrix1)

CREATE (keanu)-[:ACTS\_IN { role : 'Neo' }]->(matrix2)

CREATE (keanu)-[:ACTS\_IN { role : 'Neo' }]->(matrix3)

CREATE (laurence)-[:ACTS\_IN { role : 'Morpheus' }]->(matrix1)

CREATE (laurence)-[:ACTS\_IN { role : 'Morpheus' }]->(matrix2)

CREATE (laurence)-[:ACTS\_IN { role : 'Morpheus' }]->(matrix3)

CREATE (carrieanne)-[:ACTS\_IN { role : 'Trinity' }]->(matrix1)

CREATE (carrieanne)-[:ACTS\_IN { role : 'Trinity' }]->(matrix2)

CREATE (carrieanne)-[:ACTS\_IN { role : 'Trinity' }]->(matrix3)

**Problem 2**. Keanu Reeves acted in the movie “John Wick” which is not in the database. That movie was directed by Chad Stahelski and David Leitch. Cast of the movie included William Dafoe and Michael Nyquist. Add all of those people and the roles they played in this movie to the database using JAVA REST API or one of other RESTful APIs for Neo4J in a language of your choice. Demonstrate that you have successfully brought data about John Wick movie into the database. You can use Cypher Browser or any other means.

To get details of John Wick movie:

MATCH (actor:Actor)-[r:ACTS\_IN]->(movie:Movie {title:"John Wick"}) return actor.name as person\_name, type(r) as contributed\_as, movie.title as movie\_title

union

MATCH (director:Director)-[r:DIRECTED]->(movie:Movie {title:"John Wick"}) return director.name as person\_name, type(r) as contributed\_as, movie.title as movie\_title

To get whole graph:

START n=node(\*) MATCH (n)-[r]->(m) RETURN n,r,m;

To get all nodes:

START n=node(\*) RETURN n;

**Problem 3**. Find a list of actors playing in movies in which Keanu Reeves played. Find directors of movies in which K. Reeves played.

MATCH (:Actor {name:"Keanu Reeves"})-[:ACTS\_IN]->(:Movie)<-[:ACTS\_IN]-(actor:Actor) return distinct actor.name;

curl -i -H accept:application/json -H content-type:application/json -XPOST --user neo4j:Elcapitan1011 http://localhost:7474/db/data/transaction/commit -d '{"statements":[{"statement":"MATCH (:Actor {name:{actor\_name}})-[:ACTS\_IN]->(:Movie)<-[:ACTS\_IN]-(actor:Actor) return distinct actor.name;", "parameters": {"actor\_name": "Keanu Reeves"}}]}'

**Problem 4**. Find a way to export data from Neo4j into a set of CSV files. Delete your database and demonstrate that you can recreate it by loading those CSV files.

MATCH (actor:Actor) return actor.name as name;

MATCH (movie:Movie) return movie.title as title,movie.year as year;

MATCH (director:Director) return director.name as name;

MATCH (actor:Actor)-[r]->(movie:Movie)

return actor.name as actor\_name, r.role as role,

movie.title as movie\_title;

MATCH (director:Director)-[c]->(movie:Movie)

return director.name as director\_name, c.credits as credits,

movie.title as movie\_title;

LOAD CSV WITH HEADERS FROM "file:///Users/rpulekar/work/big-data-analytics-harvard/lectures\_and\_assignments/week-10-apr-8-2016/assignment/solution/database\_exports/export\_movie.csv" AS line MERGE (movie:Movie { title:line.title}) ON CREATE SET movie.year=line.year;

LOAD CSV WITH HEADERS FROM "file:///Users/rpulekar/work/big-data-analytics-harvard/lectures\_and\_assignments/week-10-apr-8-2016/assignment/solution/database\_exports/export\_actor.csv" AS line MERGE (a:Actor { name:line.name});

LOAD CSV WITH HEADERS FROM "file:///Users/rpulekar/work/big-data-analytics-harvard/lectures\_and\_assignments/week-10-apr-8-2016/assignment/solution/database\_exports/export\_director.csv" AS line MERGE (a:Director { name:line.name});

LOAD CSV WITH HEADERS FROM "file:///Users/rpulekar/work/big-data-analytics-harvard/lectures\_and\_assignments/week-10-apr-8-2016/assignment/solution/database\_exports/export\_acts\_in\_relationship.csv" AS line

FIELDTERMINATOR ","

MERGE (m:Movie { title:line.movie\_title })

MERGE (a:Actor { name:line.actor\_name })

MERGE (a)-[:ACTS\_IN { role:line.role}]->(m);

LOAD CSV WITH HEADERS FROM "file:///Users/rpulekar/work/big-data-analytics-harvard/lectures\_and\_assignments/week-10-apr-8-2016/assignment/solution/database\_exports/export\_directed\_relationship.csv" AS line

FIELDTERMINATOR ","

MERGE (m:Movie { title:line.movie\_title })

MERGE (d:Director { name:line.director\_name })

MERGE (d)-[:DIRECTED { credits:line.credits}]->(m);

START n=node(\*) MATCH (n)-[r]->(m) RETURN n,r,m;