Lab IST769 Unit I - Graph

## Agenda

1. Your Questions
2. Go over Problem Set

## 1. Your Questions

Ask any questions you have here!

* Revisit Question 7th, 8th and 9th

## 2. Go Over Problem Set I

1. Using the **:play northwind-graph** command, build the Northwind Product Catalog in Neo4j. Make sure to load the Product, Categories and Suppliers nodes along with the PART\_OF and SUPPLIES relationships by running the sample code provided. This should just be a matter of following the commands in the first 3 steps of the Northwind graph.

As proof you've completed this correctly, write a Cipher query to display all Products, Suppliers and Categories using both relationships. Your screenshot should include your Cipher code plus the graph output. If you did it correctly there should be 8 categoires, 77 products, 29 supplier notes (and lines connecting nodes).

2. Select a Supplier Contact Name of your choosing. Write a Cipher query to display the supplier name, the supplier’s company name, the names of the products they supply and the unit price of those products for only products that are not discontinued. Display this information in a table.

3. Select two products of your choosing. Write a Cypher query to display a graph of the supplier and category for those products. Based on the graph displayed as output, are the two products you selected in the same category?

4. You just sold 30 units of `laughing lumberjack lager` update the node to reflect the proper stock and display the output.

5. Load a Spark dataframe of USA country suppliers and their products for products that are not discontined. Include supplier company and contact names, country and phone. From products include product name, discontinued, unit price and units in stock.

6. Load the **/datasets/fudgemart/fudgemart-employees.json** into a spark dataframe. Make sure to create a column **employee\_name** which combines the first and last names together.

7. In Spark, load the employees into Neo4j under the label node Employee include employee\_name, employee\_department, employee\_id and employee\_jobtitle as node attributes. Make sure employee\_name is the first attribute as this will be the Node’s visible label.

# Cypher query

cql7 = '''

MERGE (e:Employee {name: event.employee\_name

, department: event.employee\_department

, id: event.employee\_id

, jobtitle: event.employee\_jobtitle

})

'''

# Select only relevant rows and unique it

data\_sdf = employee\_sdf.select('employee\_name', 'employee\_department'

, 'employee\_id', 'employee\_jobtitle').distinct()

# Execute Cypher query on data

data\_sdf.write.format("org.neo4j.spark.DataSource")\

.mode("Overwrite") \

.option("url", bolt\_url) \

.option("query",cql7).save()

Provide evidence the nodes were created in Neo4i UI with a Cypher query.

8. In Spark add a SUPERVISES relationship to the nodes. Basically you must match two nodes one where the id is the employee\_id and the other where the id is the employee\_supervisor\_id and then merge a relationship.

# Cypher query

cql8 = '''

MATCH (s:Employee {id: event.employee\_supervisor\_id}), (e:Employee {id: event.employee\_id})

MERGE (s)-[:SUPERVISES]->(e);

'''

# Select only relevant rows and unique it

data\_sdf = employee\_sdf.select('employee\_id', 'employee\_supervisor\_id').distinct()

# Execute Cypher query on data

data\_sdf.write.format("org.neo4j.spark.DataSource")\

.mode("Overwrite") \

.option("url", bolt\_url) \

.option("query",cql8) \

.save()

Provide evidence the relationships were created in Neo4i UI with a Cypher query to show the employees of a supervisor of your choosing.

In Neo4J build an organizational chart by starting at the supervisor who is the “CEO” and the employees recursively 4 levels deep. To learn how to query recursively, check out:

<https://stackoverflow.com/questions/31079881/simple-recursive-cypher-query>

Display the graph.