Homework Problem Set I Submission Form

# Overview

| Your Name | Nick Videtti |
| --- | --- |
| Your SU Email | nvidetti@syr.edu |

# Instructions

Put your name and SU email at the top. Answer these questions all from the lab. When asked to include screenshots, please follow the screenshot guidelines from the first homework.

Remember as you complete the homework that it is not only about getting it right/correct. We will discuss the answers in class so it’s important to articulate anything you would like to contribute to the discussion in your answer:

* If you feel the question is vague, include any assumptions you've made.
* If you feel the answer requires interpretation or justification, provide it.
* If you do not know the answer to the question, articulate what you tried and how you are stuck.
* Highlight any doubts or questions you would like me to review.

This how you receive credit for answering questions that might not be correct. In addition, you must complete the reflection portion of the homework assignment for full credit. Since most answers will be similar this is an important part of your individual submission.

Complete Part II of this document first, then go back and complete the Reflection in Part I.

# Part I: Reflection

Use this section to reflect on your learning. To achieve the highest grade on the assignment, you must be as descriptive and personal as possible with your reflection.

1. As you completed this assignment, identify what you learned.

**This assignment definitely helped me better understand the concepts and potential use cases for a Graph Database, although it is still difficult to wrap my head around what a Graph Data Model may look like for truly Big Data.**

1. What barriers or challenges did you encounter while completing this assignment?

**Command to load data from the northwind csv files had the wrong protocol in it, I had to use https instead of http, which I only discovered through a shot in the dark after an hour of troubleshooting. I also spent many more hours troubleshooting issues with the org.neo4j.spark.DataSource formatting class and eventually had to contact the professor for assistance, but eventually found out I needed an additional configuration option in my PySpark session and that resolved the issue. I later learned that this was actually just something I missed at the top of the reference notebook and once I learned that it helped me a lot.**

1. How prepared were you to complete this assignment? What can you do to be better prepared?

**I was not very prepared for this assignment as I did not really grasp the concept of a Graph Database from the asynchronous work. I’m not sure there’s anything I could’ve done to be more prepared other than maybe go through the asynchronous work a second time over.**

1. Rate your comfort level with this week’s material. Use the rubric provided.

4 ==> I understand this material and can explain it to others.  
3 ==> I understand this material.  
**2 ==> I somewhat understand the material but sometimes need guidance from others.**  
1 ==> I understand very little of this material and need extra help.

# Part II: Questions

**For each question, include a copy of the code required to complete the question along with a screenshot of the code and a screenshot of the output.**

1. Using the **:play northwind-graph** command, build the Northwind Product Catalog in Neo4j.   
   Diagram, table

   Description automatically generated  
   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 three steps of the Northwind graph.   
     
   As proof you've completed this correctly, write a Cypher query to display all Products, Suppliers, and Categories using both relationships. Your screenshot should include your Cypher code plus the graph output. If you did it correctly, there should be eight categories, 77 products, 29 supplier notes (and lines connecting nodes).

**LOAD CSV WITH HEADERS FROM "https://data.neo4j.com/northwind/products.csv" AS row**

**CREATE (n:Product)**

**SET n = row,**

**n.unitPrice = toFloat(row.unitPrice),**

**n.unitsInStock = toInteger(row.unitsInStock), n.unitsOnOrder = toInteger(row.unitsOnOrder),**

**n.reorderLevel = toInteger(row.reorderLevel), n.discontinued = (row.discontinued <> "0")**

**LOAD CSV WITH HEADERS FROM "https://data.neo4j.com/northwind/categories.csv" AS row**

**CREATE (n:Category)**

**SET n = row**

**LOAD CSV WITH HEADERS FROM "https://data.neo4j.com/northwind/suppliers.csv" AS row**

**CREATE (n:Supplier)**

**SET n = row**

**CREATE INDEX ON :Product(productID)**

**CREATE INDEX ON :Product(productName)**

**CREATE INDEX ON :Category(categoryID)**

**CREATE INDEX ON :Supplier(supplierID)**

**MATCH (p:Product),(c:Category)**

**WHERE p.categoryID = c.categoryID**

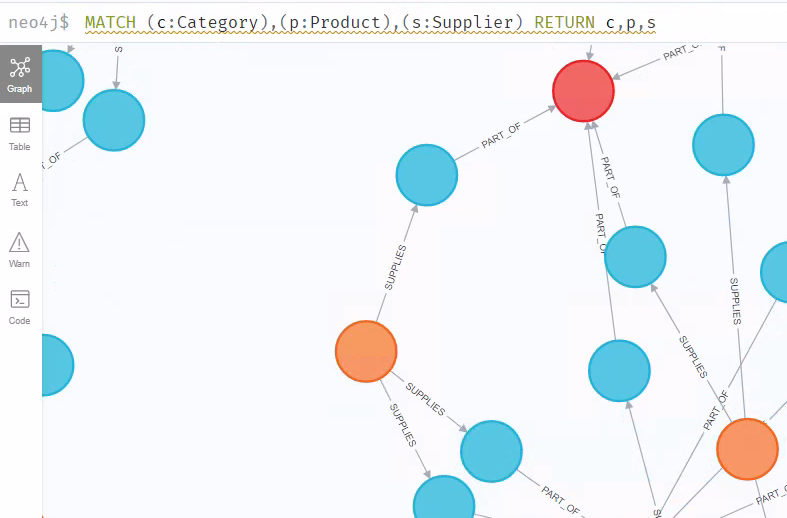
**CREATE (p)-[:PART\_OF]->(c)**

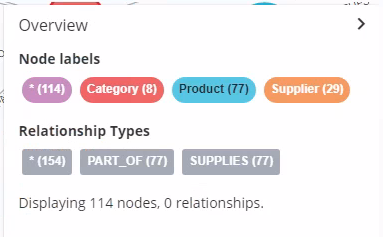
**MATCH (p:Product),(s:Supplier)**

**WHERE p.supplierID = s.supplierID**

**CREATE (s)-[:SUPPLIES]->(p)**

**MATCH (c:Category),(p:Product),(s:Supplier) RETURN c,p,s**

****



1. Select a supplier contact name of your choosing. Write a Cypher query to display the supplier’s name, the supplier’s company name, the names of the products it supplies, and the unit price of those products for only products that are not discontinued. Display this information in a table.

**match (s:Supplier {contactName: 'Shelley Burke'})-[srel:SUPPLIES]->(p:Product)**

**where p.discontinued = false**

**return s.contactName AS Contact, s.companyName AS Company, p.productName AS Product, p.unitPrice AS Unit\_Price**

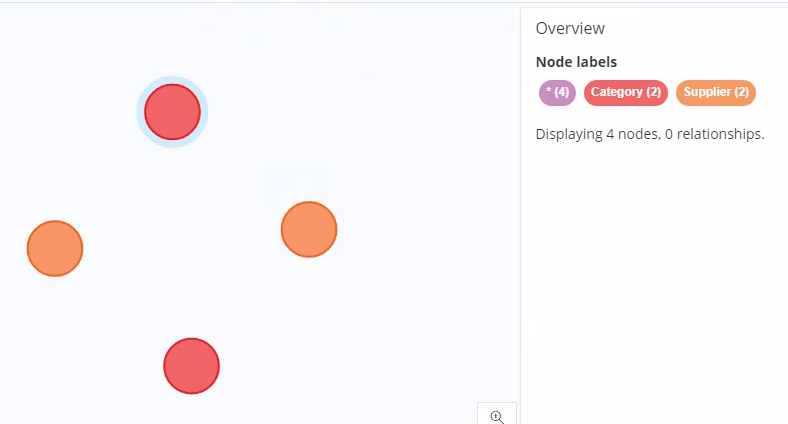


1. Select two products of your choosing. Write a Cypher query to display a graph of the supplier and category for those products. Based on the graph displayed as output, are the two products you selected in the same category?

**match (c:Category)<-[:PART\_OF]-(p:Product)<-[:SUPPLIES]-(s:Supplier)**

**where p.productName IN ['Ipoh Coffee', 'Chocolade']**

**return c,s**

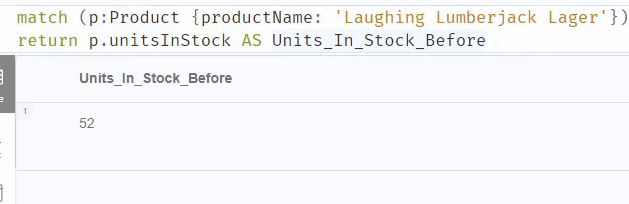
****

**It appears that the two products are not in the same category since two distinct categories are in the output graph.**

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

**match (p:Product {productName: 'Laughing Lumberjack Lager'})**

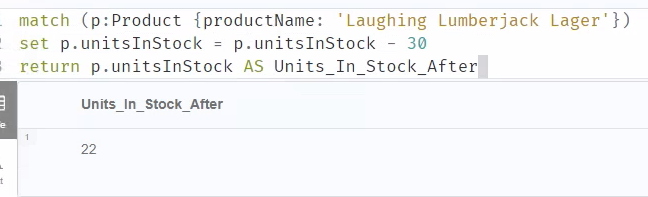
**return p.unitsInStock AS Units\_In\_Stock\_Before**

****

**match (p:Product {productName: 'Laughing Lumberjack Lager'})**

**set p.unitsInStock = p.unitsInStock - 30**

**return p.unitsInStock AS Units\_In\_Stock\_After**

****

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

**query = '''**

**match (s:Supplier)-[:SUPPLIES]->(p:Product)**

**where p.discontinued = false**

**return s.companyName AS Company,**

**s.contactName AS Contact, s.country AS Country, s.phone AS Phone,**

**p.productName AS Product, p.discontinued AS Discontinued,**

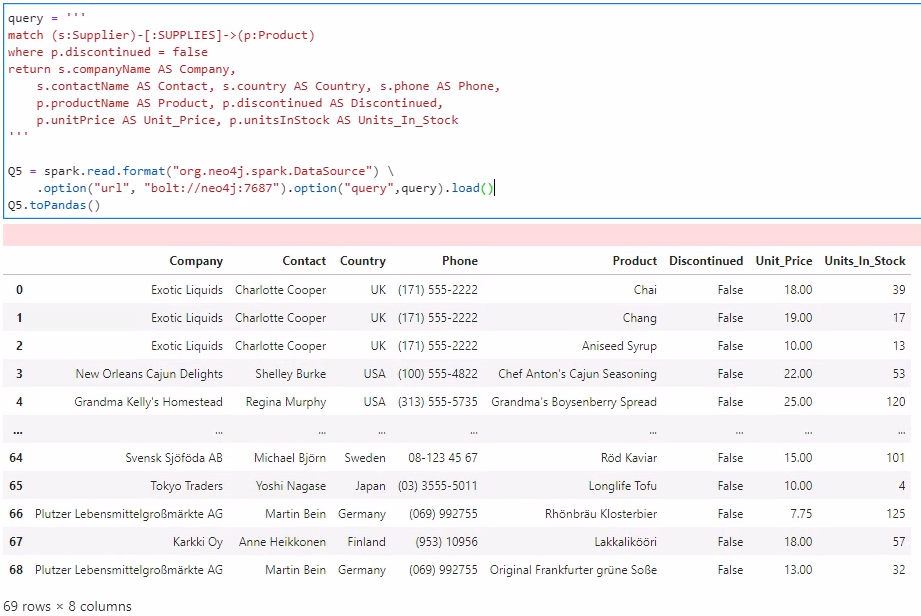
**p.unitPrice AS Unit\_Price, p.unitsInStock AS Units\_In\_Stock**

**'''**

**Q5 = spark.read.format("org.neo4j.spark.DataSource") \**

**.option("url", "bolt://neo4j:7687").option("query",query).load()**

**Q5.toPandas()**

****

1. Load the **/datasets/fudgemart/fudgemart-employees.json** into a Spark DataFrame. Make sure to create a column **employee\_name** that combines the first and last names together.

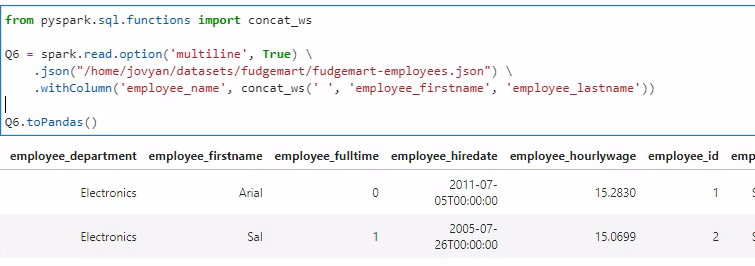
**from pyspark.sql.functions import concat\_ws**

**Q6 = spark.read.option('multiline', True) \**

**.json("/home/jovyan/datasets/fudgemart/fudgemart-employees.json") \**

**.withColumn('employee\_name', concat\_ws(' ', 'employee\_firstname', 'employee\_lastname'))**

**Q6.toPandas()**

****



1. 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.   
     
   Provide evidence the nodes were created in Neo4i UI with a Cypher query.

**query = '''**

**merge (e:Employee {Name: event.employee\_name,**

**Department: event.employee\_department,**

**EID: event.employee\_id,**

**Title: event.employee\_jobtitle})**

**'''**

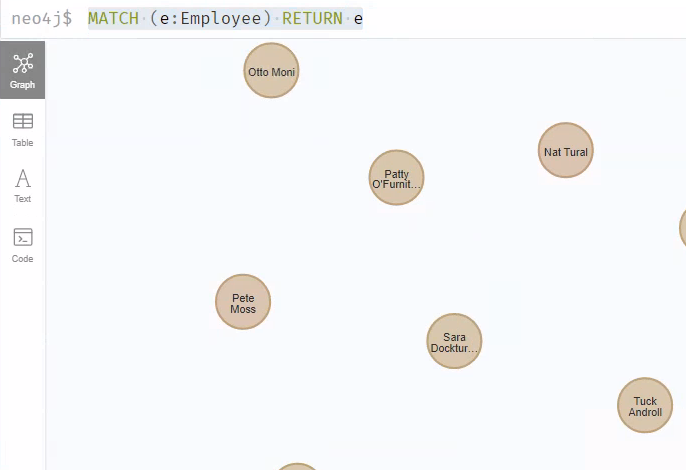
**Q6.write.format("org.neo4j.spark.DataSource").mode("Overwrite") \**

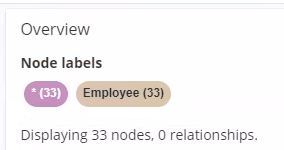
**.option("url", "bolt://neo4j:7687") \**

**.option("query",query) \**

**.save()**

**MATCH (e:Employee) RETURN e**

****







1. 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.  
     
   Provide evidence the relationships were created in Neo4i UI with a Cypher query to show the employees of a supervisor of your choosing.

**query = '''**

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

**MERGE (e)<-[:SUPERVISES]-(s)**

**'''**

**Q6.write.format("org.neo4j.spark.DataSource").mode("Overwrite") \**

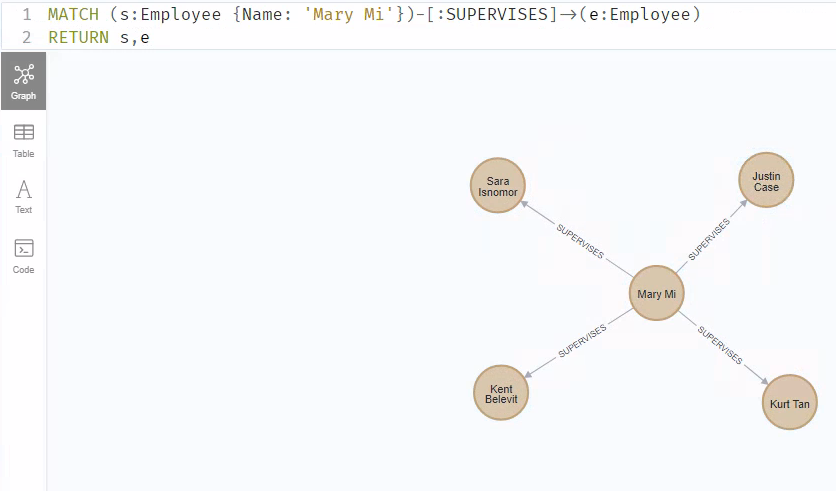
**.option("url", "bolt://neo4j:7687") \**

**.option("query",query) \**

**.save()**

**MATCH (s:Employee {Name: 'Mary Mi'})-[:SUPERVISES]->(e:Employee)**

**RETURN s,e**

****

1. In Neo4J build an organizational chart by starting at the supervisor who is the “CEO” and the employees recursively four levels deep. To learn how to query recursively, check out:  
   <https://stackoverflow.com/questions/31079881/simple-recursive-cypher-query>   
   Display the graph.

**MATCH (s:Employee)-[:SUPERVISES \*1..]->(e:Employee)**

**RETURN s,e**

