Homework Problem Set I Submission Form

# Overview

|  |  |
| --- | --- |
| Your Name | Sana Khan |
|  | Skhan53@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.
2. What barriers or challenges did you encounter while completing this assignment?
3. How prepared were you to complete this assignment? What can you do to be better prepared?
4. 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

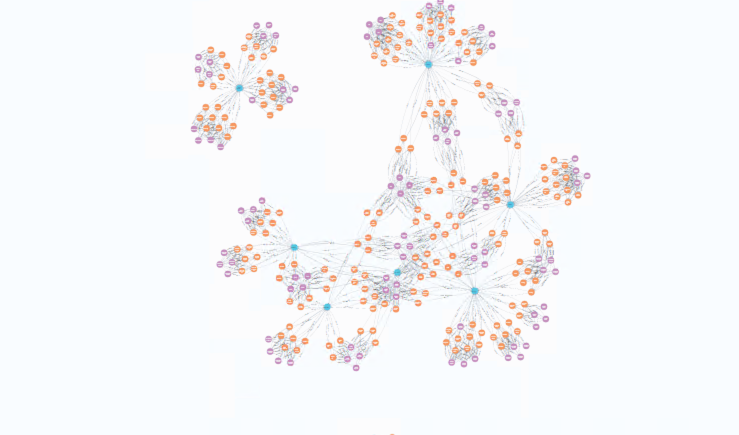
**For each question, include a copy of the code required to complete the question along with a screenshot of the code and a screenshot of the output.**

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

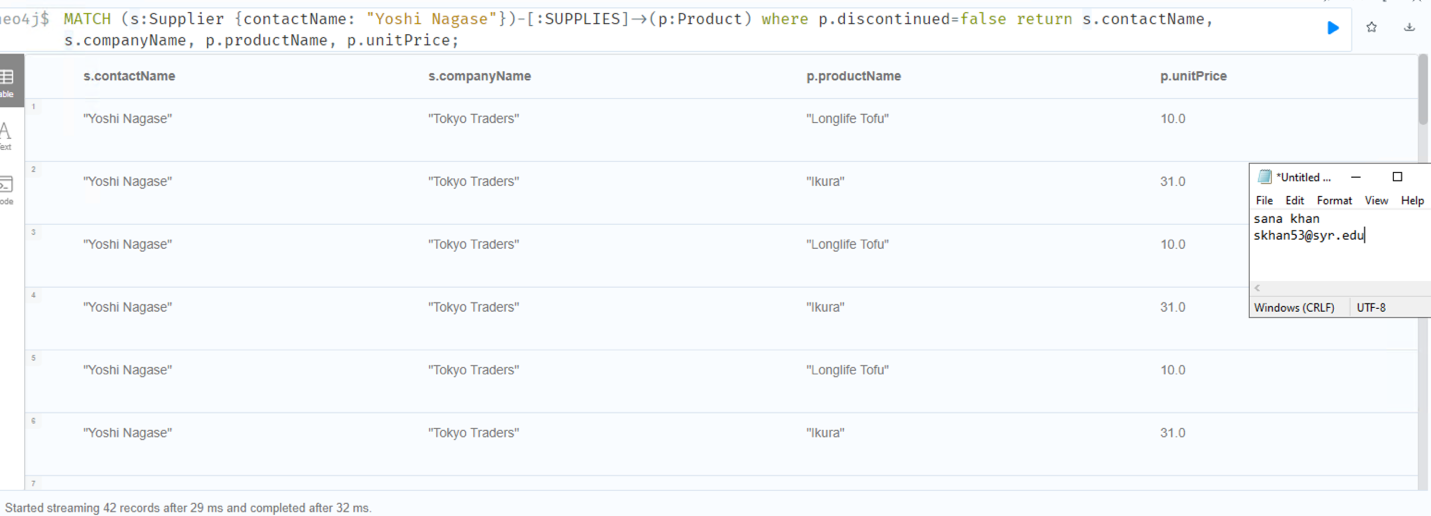
   Description automatically generated  
   Make sure to load the Product, Categories, and Suppliers nodes along with the PART\_OF and SUPPLIES relationships by running the sample code provided. This should just be a matter of following the commands in the first three steps of the Northwind graph.

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).

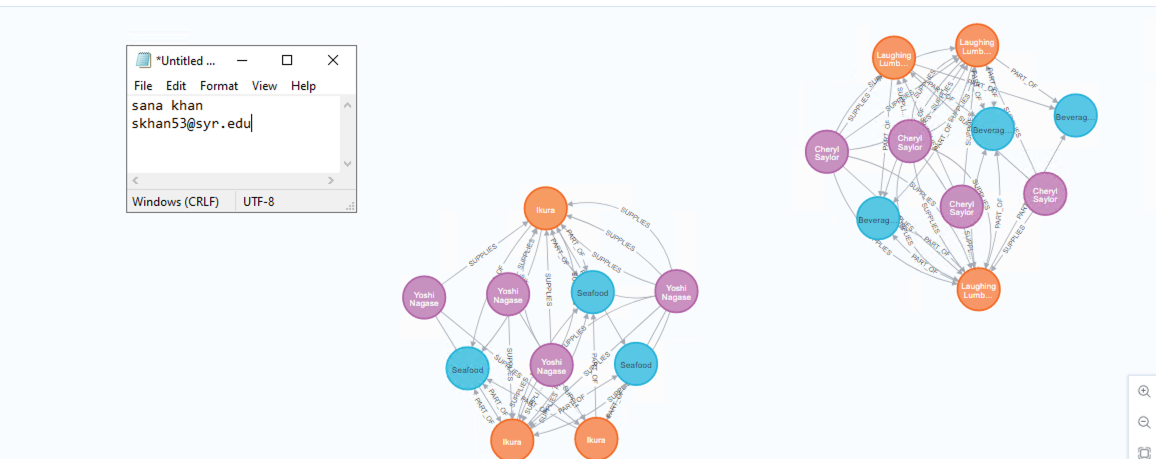
* 1. MATCH (s:Supplier) RETURN s;
  2. MATCH (c: Categories) RETURN c;
  3. MATCH (p: Product) RETURN p;
  4. MATCH(s:Supplier)-->(p:Product)-->(c:Category)
  5. RETURN s,p,c



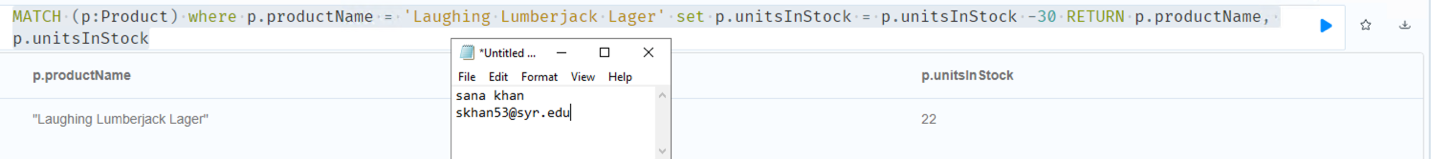
1. Select a supplier contact name of your choosing. Write a Cypher query to display the supplier’s name, the supplier’s company name, the names of the products it supplies, and the unit price of those products for only products that are not discontinued. Display this information in a table.
   1. **MATCH (s:Supplier {contactName: "Yoshi Nagase"})-[:SUPPLIES]->(p:Product) where p.discontinued=false return s.contactName, s.companyName, p.productName, p.unitPrice;**

****

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?
   1. MATCH (s:Supplier)-[:SUPPLIES]->(p:Product)-[:PART\_OF]->(c:Category) where p.productName IN ['Laughing Lumberjack Lager','Ikura'] RETURN s,p,c



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) where p.productName = 'Laughing Lumberjack Lager' set p.unitsInStock = p.unitsInStock -30 RETURN p.productName, p.unitsInStock**

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.

**cypher\_query = '''**

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

**WHERE s.country = 'USA' and p.discontinued=false**

**RETURN s.contactName, s.companyName, p.unitsInStock, p.unitPrice;**

**'''**

**df1 = spark.read.format("org.neo4j.spark.DataSource") \**

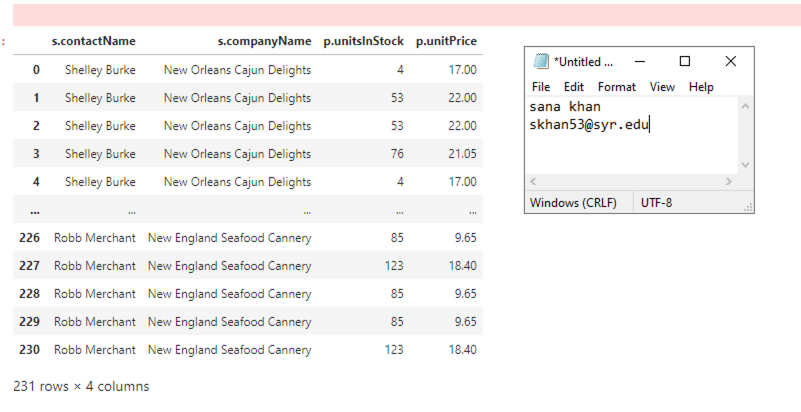
**.option("url", bolt\_url) \**

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

**.load()**

**print(bolt\_url)**

**df.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 col,concat,lit**

**employees = spark.read.option("multiline",True).json(file\_name)\**

**.withColumn("employee\_name", concat(col("employee\_firstname"),lit(" "),col("employee\_lastname")))**

**query = employees.select("employee\_id", "employee\_name", "employee\_department", "employee\_jobtitle")**

**query.toPandas()**

**A screenshot of a computer

Description automatically generated**

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.

**cypher\_merge = '''**

**MERGE (e:Employee {**

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

**name: event.employee\_name,**

**jobtitle: event.employee\_jobtitle,**

**id: event.employee\_id**

**})**

**'''**

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

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

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

**.save()**

**print(cypher\_merge)**

**A screen shot of a computer

Description automatically generated with low confidence**

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.

**cipher\_sql = '''**

**MATCH (e:Employee)**

**MATCH (s:Employee)**

**where e.id=event.employee\_id and s.id=event.employee\_supervisor\_id**

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

**'''**

**sup\_employee = employees.select("employee\_supervisor\_id", "employee\_id").orderBy("employee\_supervisor\_id")**

**sup\_employee.toPandas()**

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

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

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

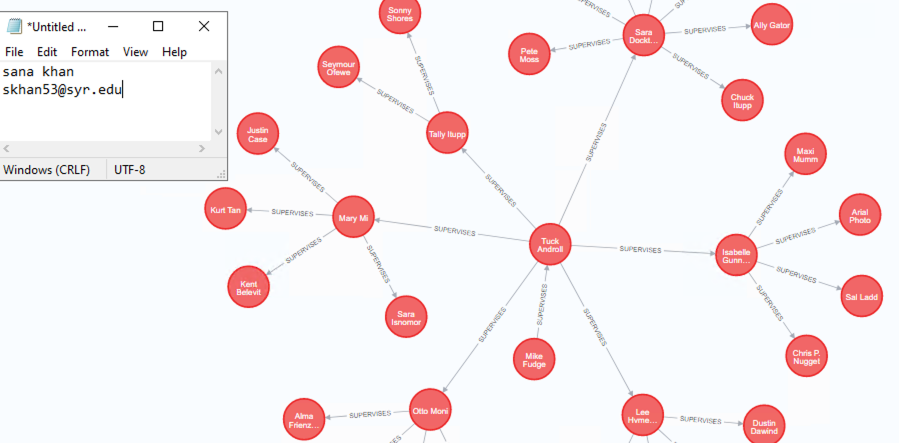
**.save()**

**print(cipher\_sql)**

**MATCH p=()-[r:SUPERVISES}->() RETURN p**

1. A screenshot of a computer

   Description automatically generated with medium confidence
2. 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 {jobtitle: "CEO"})-[:SUPERVISES \*1..3]-> (e:Employee) RETURN s,e;**