

# Problem Set I Submission Form

## Overview

Your Name	Hendi Kushta
Your SU Email	hkushta@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 screen shot guidelines from the first homework.

Remember as you complete the homework 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 which 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.

Query a graph model using Neo4j's Cypher query language. Import and export data from Neo4j into spark dataframes. Build a Graph from a table of data.

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.

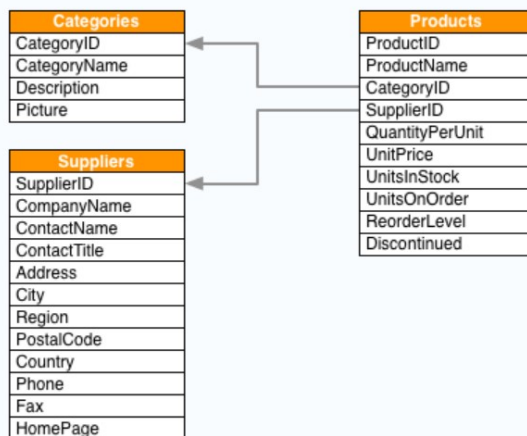
1 ==> I understand very little of this material and need extra help.

## 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.**

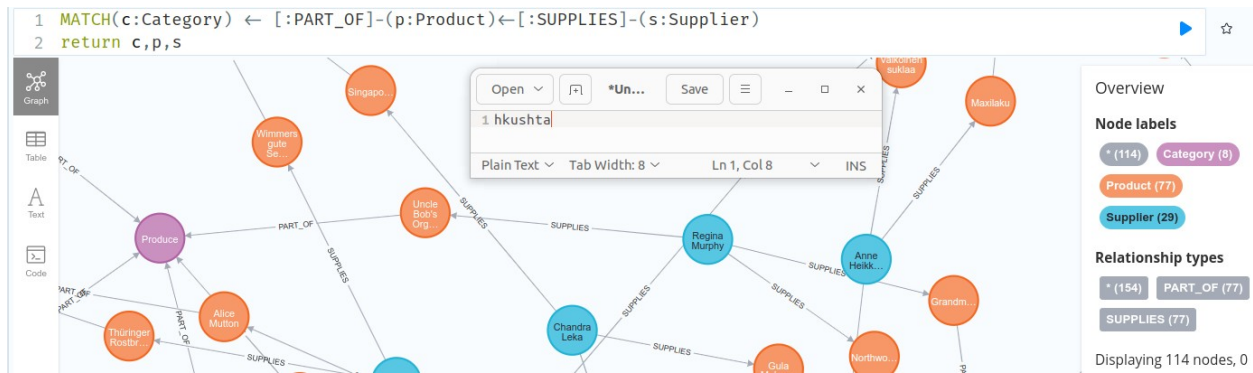
1. Using the **:play northwind-graph** command, build the Northwind Product Catalog in Neo4j.

The load statements to the right require public internet access. **LOAD CSV** will retrieve a CSV file from a valid URL, applying a Cypher statement to each row using a named map (here we're using the name ``row``).

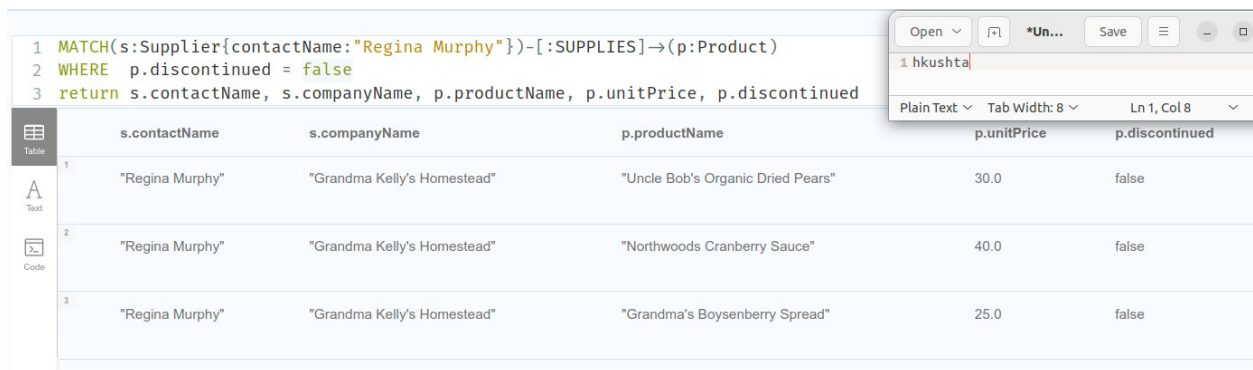


Make sure to load the Product, Categories and Suppliers nodes along with the PART\_OF and SUPPLIES relationships by running the sample code provided. This should just be a matter of following the commands in the first 3 steps of the Northwind graph.

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

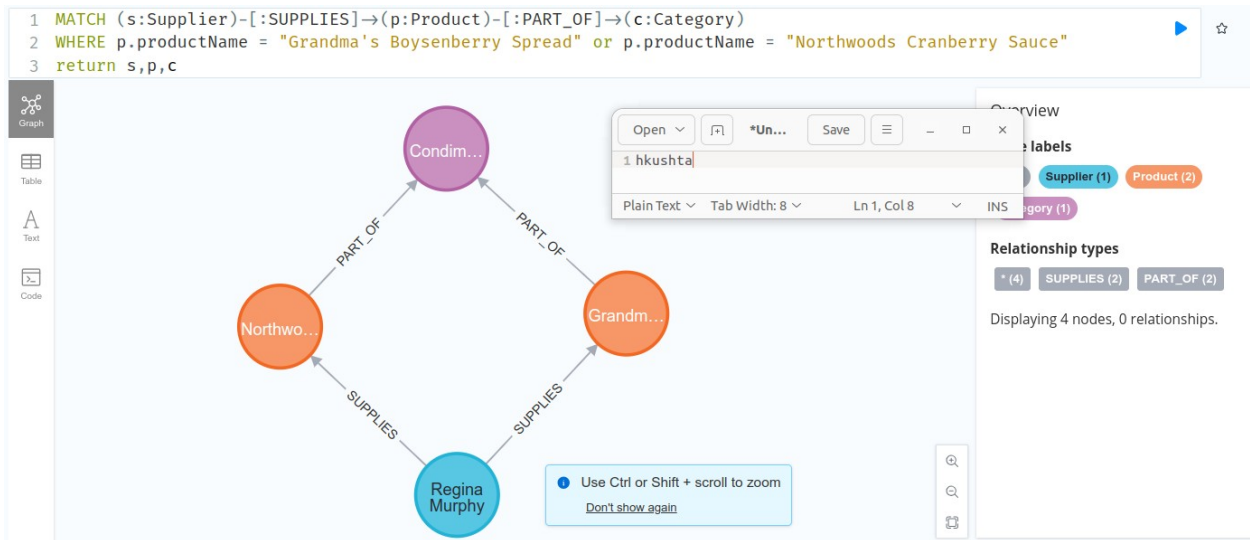


2. Select a Supplier Contact Name of your choosing. Write a Ciper query to display the supplier name, the supplier's company name, the names of the products they supply and the unit price of those products for only products that are not discontinued. Display this information in a table.



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?

Both the products that I chose are supplied by Regina Murphy and both of them are part of the same category (Condiments) which include sweet and savory sauces, relishes, spreads and seasonings.



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

To help make Neo4j Browser better we collect information on product usage. Review your [settings](#) at any time.

```

1 MATCH(p:Product)
2 WHERE p.productName = "Laughing Lumberjack Lager"
3 SET p.unitsInStock = p.unitsInStock - 30
4 return p.productName, p.unitsInStock

```

	p.productName	p.unitsInStock
1	"Laughing Lumberjack Lager"	22

- 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.

```

cipher_query = '''
MATCH (s:Supplier)-[:SUPPLIES]->(p:Product)
WHERE s.country = "USA" and p.discontinued = false
return s.companyName, s.contactName, s.country, p.productName, p.discontinued, p.unitPrice, p.unitsInStock
'''

df = spark.read.format("org.neo4j.spark.DataSource") \
    .option("url", bolt_url) \
    .option("query", cipher_query) \
    .load()

```

```
df.printSchema()
```

```

root
|-- s.companyName: string (nullable = true)
|-- s.contactName: string (nullable = true)
|-- s.country: string (nullable = true)
|-- p.productName: string (nullable = true)
|-- p.discontinued: boolean (nullable = true)
|-- p.unitPrice: double (nullable = true)
|-- p.unitsInStock: long (nullable = true)

```

Open ▾ [?] \*Un... Save [≡] - □ ×

1 hkushta

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 8 ▾ INS

```
df.toPandas()
```

	s.companyName	s.contactName	s.country	p.productName	p.discontinued	p.unitPrice	p.unitsInStock
0	New Orleans Cajun Delights	Shelley Burke	USA	Chef Anton's Cajun Seasoning	False	22.00	53
1	New Orleans Cajun Delights	Shelley Burke	USA	Louisiana Fiery Hot Pepper Sauce	False	21.05	76
2	New Orleans Cajun Delights	Shelley Burke	USA	Louisiana Hot Spiced Okra	False	17.00	4
3	Grandma Kelly's Homestead	Regina Murphy	USA	Uncle Bob's Organic Dried Pears	False	30.00	15
4	Grandma Kelly's Homestead	Regina Murphy	USA	Northwoods Cranberry Sauce	False	40.00	6
5	Grandma Kelly's Homestead	Regina Murphy	USA	Grandma's Boysenberry Spread	False	25.00	120
6	Bigfoot Breweries	Cheryl Saylor	USA	Sasquatch Ale	False	14.00	111
7	Bigfoot Breweries	Cheryl Saylor	USA	Steeleye Stout	False	18.00	20
8	Bigfoot Breweries	Cheryl Saylor	USA	Laughing Lumberjack Lager	False	14.00	22
9	New England Seafood Cannery	Robb Merchant	USA	Jack's New England Clam Chowder	False	9.65	85
10	New England Seafood Cannery	Robb Merchant	USA	Boston Crab Meat	False	18.40	123

- 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.

```

from pyspark.sql.functions import col, concat, lit
q6_emp_df = spark.read.option("multiline", True) \
    .json("file:///home/jovyan/datasets/fudgemart/fudgemart-employees.json") \
    .withColumn("employee_name", concat(col("employee_firstname"), lit(" "), col("employee_lastname")))
q6_emp_df.toPandas()

```

thdate	employee_department	employee_firstname	employee_fulltime	employee_hiredate	employee_hourlywage	employee_id	employee_jobtitle	employee_lastname	employee_ssn	employee_supervisor_id	employee_termdate	employee_name
182-01-10:00:00	Electronics	Arial	0	2011-07-05T00:00:00	15.2830	1	Sales Associate	Photo	111220001	5.0	None	Arial Photo
182-11-10:00:00	Electronics	Sal	1	2005-07-26T00:00:00				Ladd	111220002	5.0	None	Sal Ladd
172-09-10:00:00	Hardware	Dustin	0	2004-07-02T00:00:00				Dawind	111220003	6.0	2010-11-06T00:00:00	Dustin Dawind
190-05-10:00:00	Hardware	Sandi	1	2011-01-02T00:00:00				Shores	111220004	6.0	None	Sandi Shores
174-02-	Electronics	Isabella	1	2005-08-	20.1387	5	Department	Gunnarson	111220005	10.0	None	Isabella

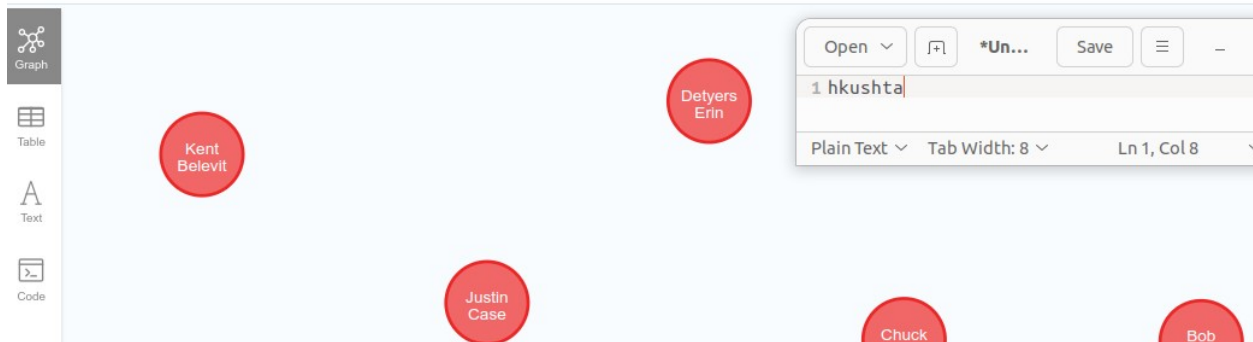
7. In Spark, load the employees into Neo4j under the label node Employee include employee\_name, employee\_department, employee\_id and employee\_jobtitle as node attributes. Make sure employee\_name is the first attribute as this will be the Node's visible label.

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

```
q7 = '''
MERGE (e:Employee { name: coalesce(event.employee_name, 'Unknown'), dept: event.employee_department,
                      jobtitle: event.employee_jobtitle, id: event.employee_id})
'''

employees = q6_emp_df.select("employee_id", "employee_name", "employee_jobtitle", "employee_department")
employees.write.format("org.neo4j.spark.DataSource").mode("Overwrite") \
    .option("url", bolt_url) \
    .option("query", q7) \
    .save()
```

```
neo4j$ MATCH (e:Employee) return e;
```



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.

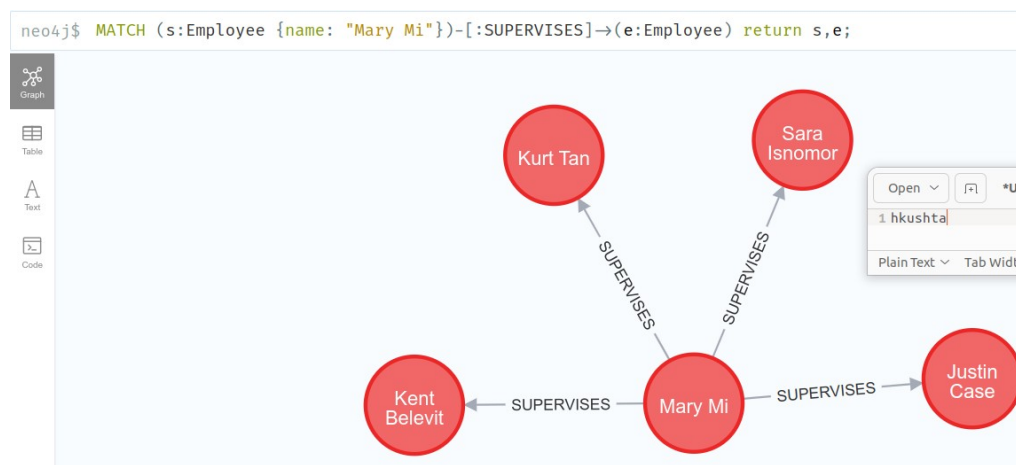
Provide evidence the relationships were created in Neo4i UI with a Cypher query to show the employees of a supervisor of your choosing.

```

q8_cipher_emp = '''
MATCH (e:Employee)
MATCH (s:Employee)
WHERE e.id = event.employee_id AND s.id = event.employee_supervisor_id
MERGE (s)-[:SUPERVISES]->(e)
'''

sup_empl = q6_emp_df.select("employee_supervisor_id", "employee_id").orderBy("employee_supervisor_id")
sup_empl.write.format("org.neo4j.spark.DataSource").mode("Overwrite") \
    .option("url", bolt_url) \
    .option("query", q8_cipher_emp) \
    .save()

```



9. 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.

