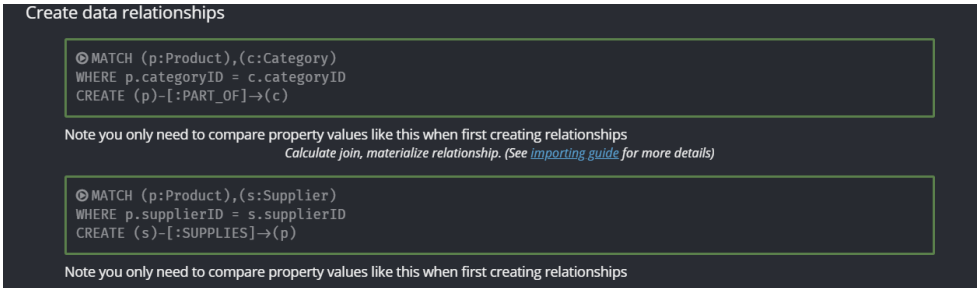
Peter Tadrous

CUS 725 - Advanced Database Systems

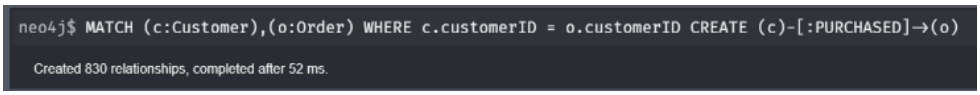
Week 11 - Homework 9

Neo4j Northwind Examples

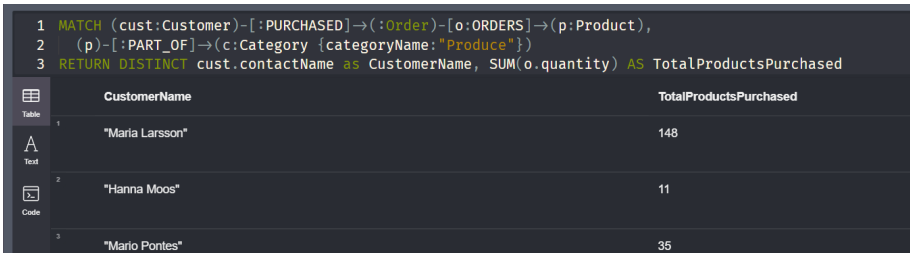
Q1. The node fields used to determine whether to insert a relationship or not for this example are the **categoryID** and the **supplierID**. These parts of the **WHERE** clause are used in conjunction with a **MATCH** statement to determine exactly which nodes are related to each other.



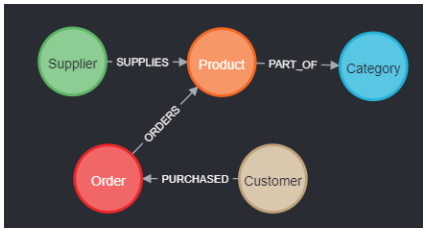
Q2. Comparing this to a relational database, we can pretend the labels are tables (and they rhyme!), and this statement would be similar to a table join since we're essentially selecting **FROM CUSTOMER c, ORDER o WHERE c.customerID = o.customerID**. However, in Neo4j, we are not selecting to return any results; we are instead querying to create a relationship. So, this is more like creating the relationship of a **FOREIGN KEY** in the **ORDER** "table" (**ORDER.customerID**) to the **PRIMARY KEY** of the **CUSTOMER** "table" (**CUSTOMER.customerID**).



Q3. This query returns a customer's name, along with how many total products they have purchased. The first part of the query creates a **cust** variable with the label **Customer**, which matches **Customer** nodes. Then there is the anonymous **PURCHASED** relationship to anonymous **Order** nodes, which relate to **o ORDERS** of **p Product**; this whole line now matches **cust** customers to **p** products they have ordered. The second part of the query matches the **p** products with another anonymous relationship **PART_OF** to **c Category** where the **categoryName** is "Produce". As a whole, the **MATCH** statement links **cust** customers to **o** orders where the **p** products in those orders are produce. The return statement returns distinct customer names and the sum of the order quantities (similar to a group by in sql) with field aliases for readability.



Q4. Here is the visualization of the graph:



Neo4j Creating Queries

1. `MATCH (cust:Customer {contactName:"Rene Phillips"})-[:PURCHASED]-(order:Order) RETURN cust, order`

```
j$ MATCH (cust:Customer {contactName:"Rene Phillips"})-[:PURCHASED]-(order:Order) RETURN cust, order
```

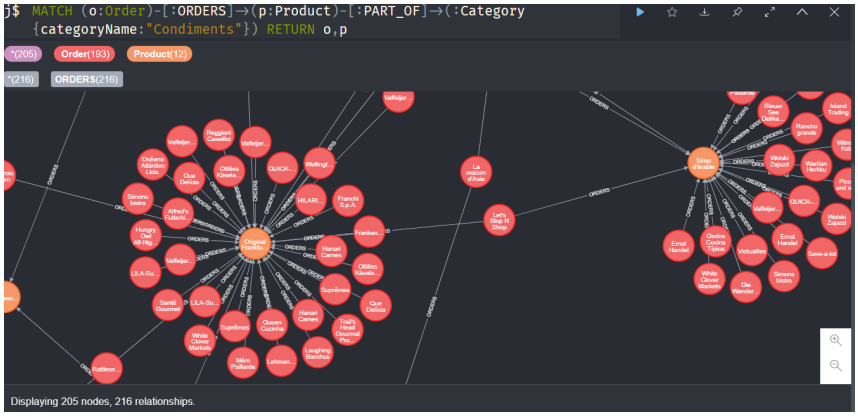
"cust"	"order"
{"country":"USA","contactTitle":"Sales Representative","address":"2743 Bering St.","phone":"(907) 555-7584","city":"Anchorage","contactName":"Rene Phillips","companyName":"Old World Delicatessen","postalCode":"99508","customerID":"OLDWO","fax":"(907) 555-2880","region":"AK"}	{"shipCity":"Anchorage","orderID":"10441","freight":"73.02","requiredDate":"1997-03-24 00:00:00.000","employeeID":"3","shipPostalCode":"99508","shipName":"Old World Delicatessen","shipCountry":"USA","shipAddress":"2743 Bering St.","shipVia":"2","customerID":"OLDWO","shippedDate":"1997-03-14 00:00:00.000","orderDate":"1997-02-10 00:00:00.000","shipRegion":"AK"}
{"country":"USA","contactTitle":"Sales Representative","address":"2743 Bering St.","phone":"(907) 555-7584","city":"Anchorage","contactName":"Rene Phillips","companyName":"Old World Delicatessen","postalCode":"99508","customerID":"OLDWO","fax":"(907) 555-2880","region":"AK"}	{"shipCity":"Anchorage","orderID":"10855","freight":"170.97","requiredDate":"1998-02-24 00:00:00.000","employeeID":"3","shipPostalCode":"99508","shipName":"Old World Delicatessen","shipCountry":"USA","shipAddress":"2743 Bering St.","shipVia":"1","customerID":"OLDWO","shippedDate":"1998-02-04 00:00:00.000","orderDate":"1998-01-27 00:00:00.000","ship

2. `MATCH (cust:Customer {city:"Paris"})-[:PURCHASED]-(order:Order) RETURN cust.companyName, order.orderID, order.orderDate`

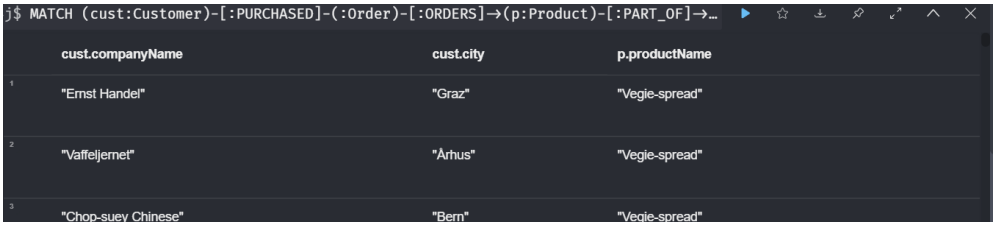
```
$ MATCH (cust:Customer {city:"Paris"})-[:PURCHASED]-(order:Order) RETURN
  cust.companyName, order.orderID, order.orderDate
```

cust.companyName	order.orderID	order.orderDate
"Spécialités du monde"	"10964"	"1998-03-20 00:00:00.000"
"Spécialités du monde"	"10738"	"1997-11-12 00:00:00.000"

3. `MATCH (o:Order)-[:ORDERS]->(p:Product)-[:PART_OF]->(:Category {categoryName:"Condiments"}) RETURN o,p`
    a. There are **12** product nodes returned.



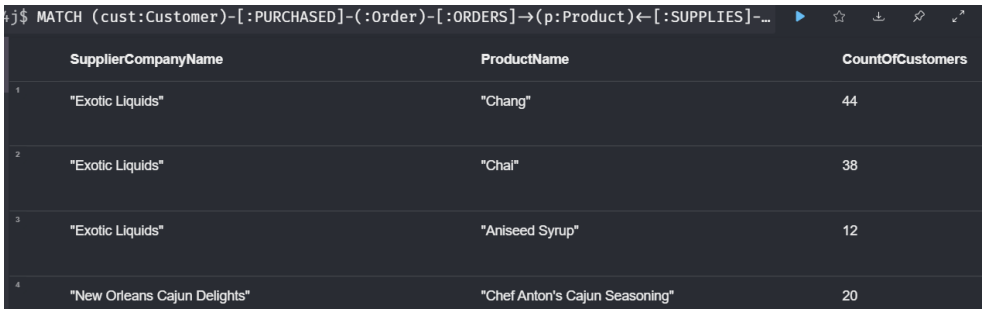
4. `MATCH (cust:Customer)-[:PURCHASED]-(:Order)-[:ORDERS]->(p:Product)-[:PART_OF]->(:Category {categoryName:"Condiments"}) RETURN cust.companyName, cust.city, p.productName`



```
j$ MATCH (cust:Customer)-[:PURCHASED]-(:Order)-[:ORDERS]→(p:Product)-[:PART_OF]→…
```

	cust.companyName	cust.city	p.productName
1	"Ernst Handel"	"Graz"	"Vegie-spread"
2	"Vaffeljernet"	"Århus"	"Vegie-spread"
3	"Chop-suey Chinese"	"Bern"	"Vegie-spread"

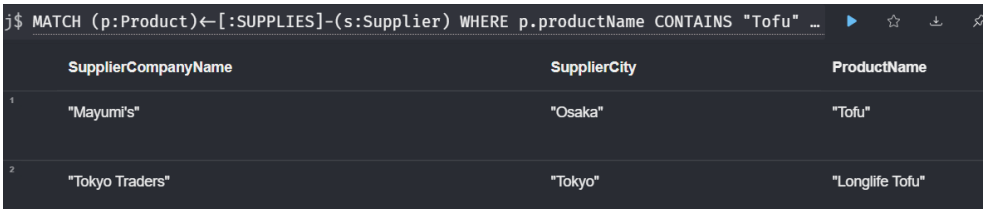
5. `MATCH (cust:Customer)-[:PURCHASED]-(:Order)-[:ORDERS]->(p:Product)<-[:SUPPLIES]-(s:Supplier) RETURN DISTINCT s.companyName AS SupplierCompanyName, p.productName AS ProductName, COUNT(cust.companyName) AS CountOfCustomers`



```
4j$ MATCH (cust:Customer)-[:PURCHASED]-(:Order)-[:ORDERS]→(p:Product)←[:SUPPLIES]-…
```

	SupplierCompanyName	ProductName	CountOfCustomers
1	"Exotic Liquids"	"Chang"	44
2	"Exotic Liquids"	"Chai"	38
3	"Exotic Liquids"	"Aniseed Syrup"	12
4	"New Orleans Cajun Delights"	"Chef Anton's Cajun Seasoning"	20

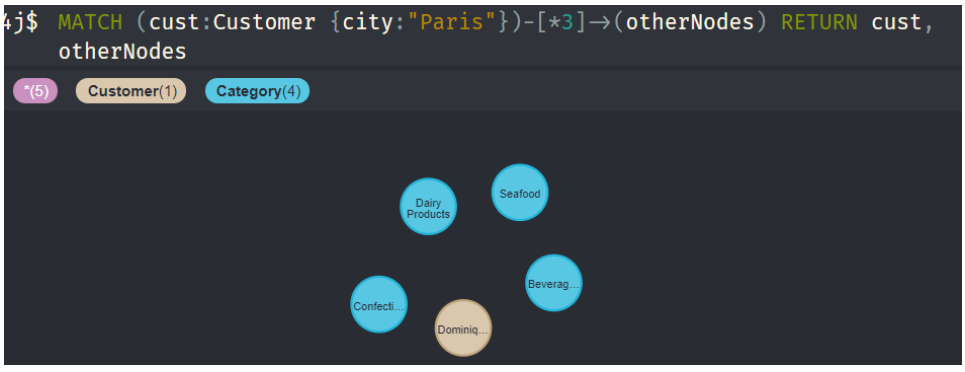
6. `MATCH (p:Product)<-[:SUPPLIES]-(s:Supplier) WHERE p.productName CONTAINS "Tofu" RETURN s.companyName AS SupplierCompanyName,s.city AS SupplierCity, p.productName AS ProductName`



```
j$ MATCH (p:Product)←[:SUPPLIES]-(s:Supplier) WHERE p.productName CONTAINS "Tofu" …
```

	SupplierCompanyName	SupplierCity	ProductName
1	"Mayumi's"	"Osaka"	"Tofu"
2	"Tokyo Traders"	"Tokyo"	"Longlife Tofu"

7. `MATCH (cust:Customer {city:"Paris"})-[*3]->(otherNodes) RETURN cust, otherNodes`



8. `MATCH p=shortestPath( (paris:Customer {city:"Paris"})-[*]-(anchorage:Customer {city:"Anchorage"}) ) RETURN p`