

Assignment 6: Graph Database Application

Homework questions:

1. Use the provided Cypher script to create the graph database:
 - a. You could use any names for your project and the graph database.

Created Project “Assign 8”

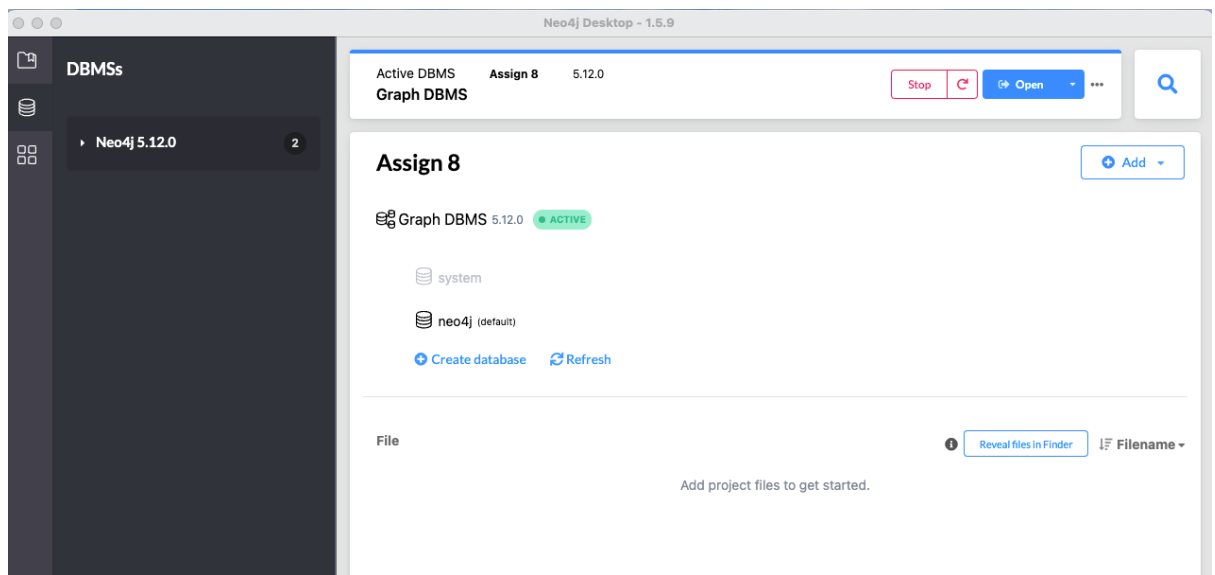


Fig 1: Project “Assign 8” Created and connected to Local DBMS

- b. Copy the ENTIRE Cypher code in the script and paste it in neo4j\$ prompt and then click the blue play button on the right.

Created Database using the Cypher code provided with assignment and ran it in neo4j browser console. Attaching below screenshot for reference:

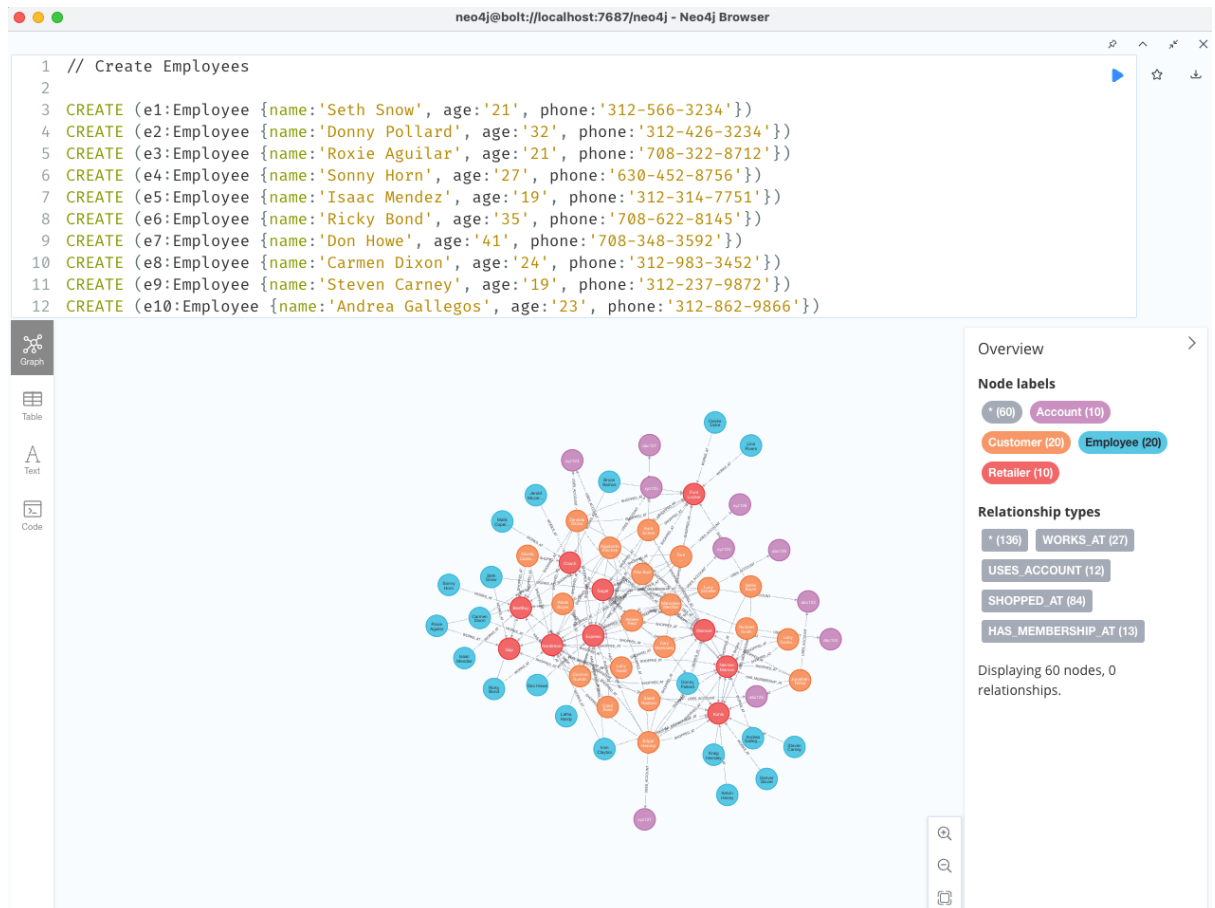


Fig 2: Graph DB Created using Cypher code

- c. **NOTE in step 15 above that your version may only allow one command at a time.**

I am using Neo4j Desktop 1.5.9 on Mac OS platform, we can expand the console to execute more than 1 line at a time. Attaching screenshot with highlighted icon to do the same.

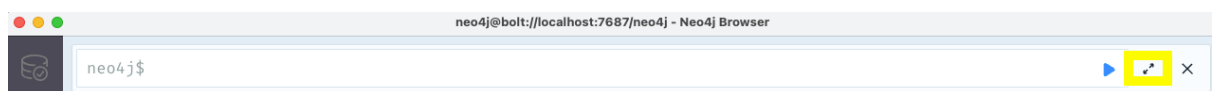


Fig 3: Neo4j command console expand icon

- d. **Run the command below. Find the Customer Ashlee Reid and pull the node to the far left of the screen. Include a screen capture of this view to show you were able to load the database. (5 points)**

```
MATCH (n) RETURN (n);
```

Ran the command successfully and pulled the customer **Ashlee Reid** node to the far left of the screen. Attaching the below screenshot for the same.

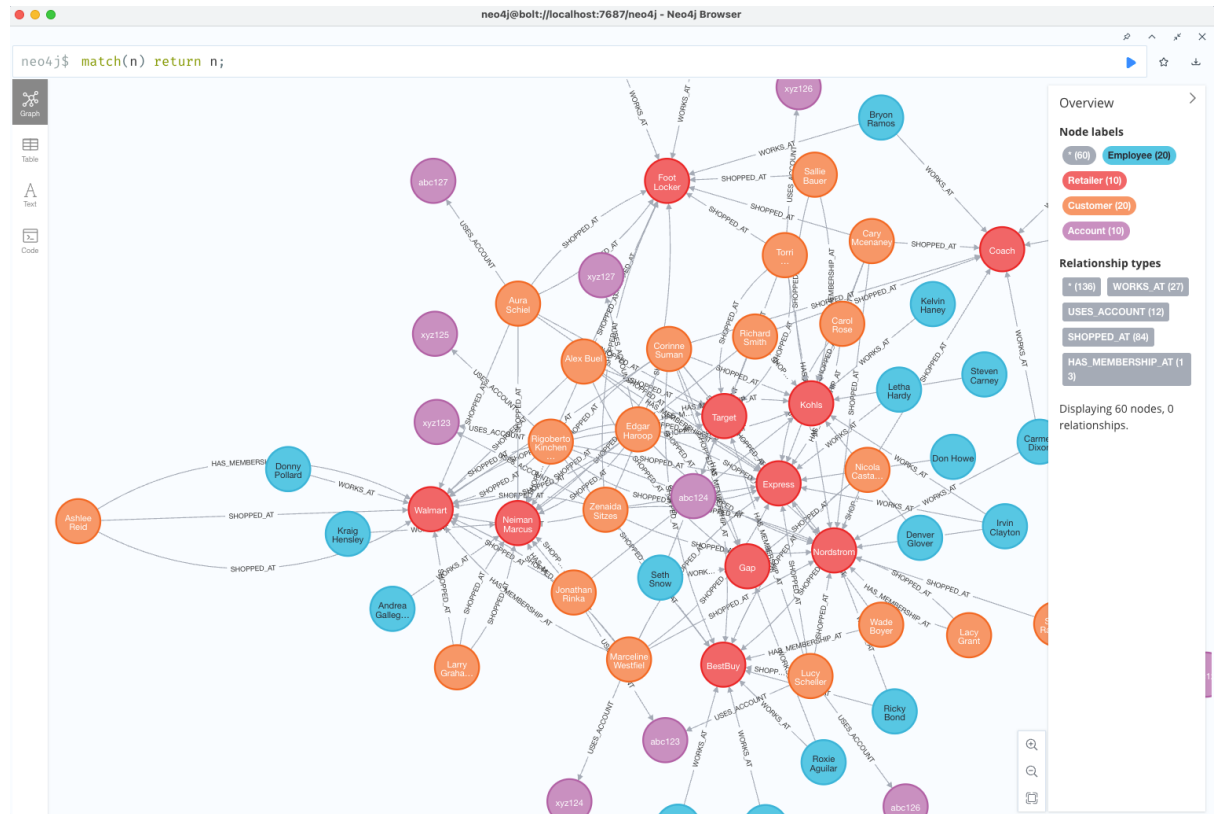


Fig 4: Screenshot of customer Ashlee Reid node on the far left

2. Execute the following Cypher code to get the list of retailers. (0 point).

```
MATCH (r:Retailer) RETURN (r);
```

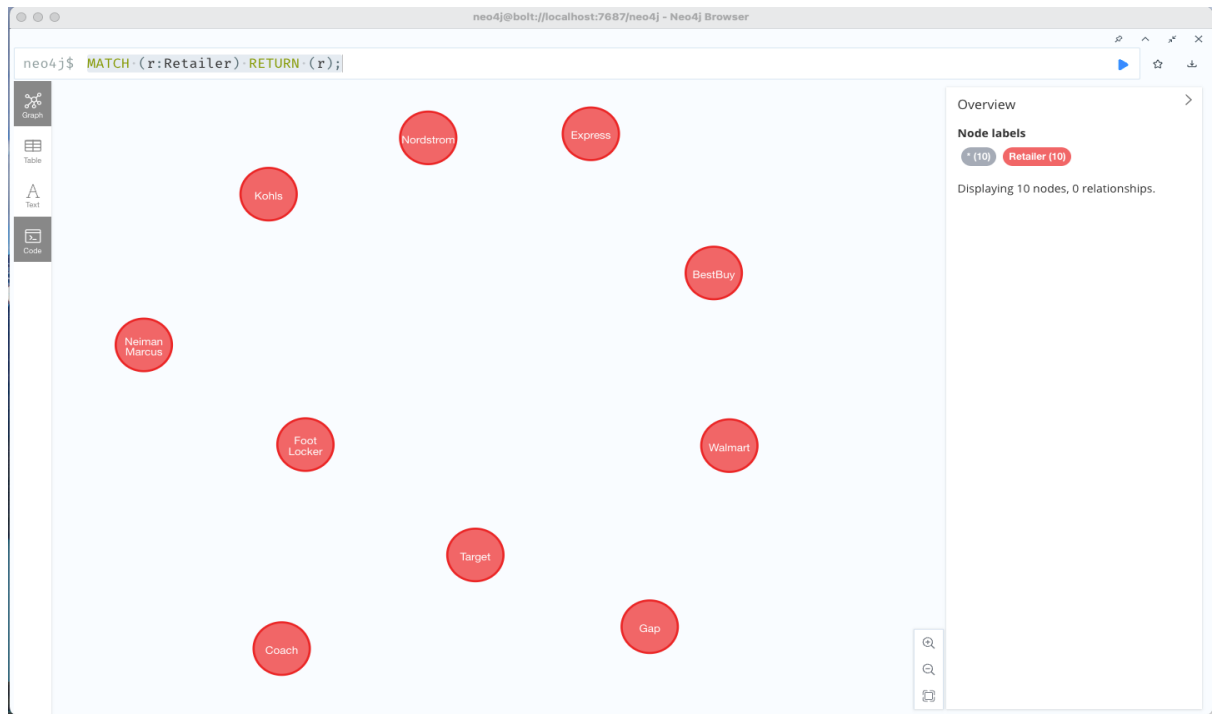


Fig 5: Screenshot of graph result contains nodes of all Retailers

3. Execute the following Cypher code to the get the list of employees. (0 point)

```
MATCH (e:Employee) RETURN (e);
```

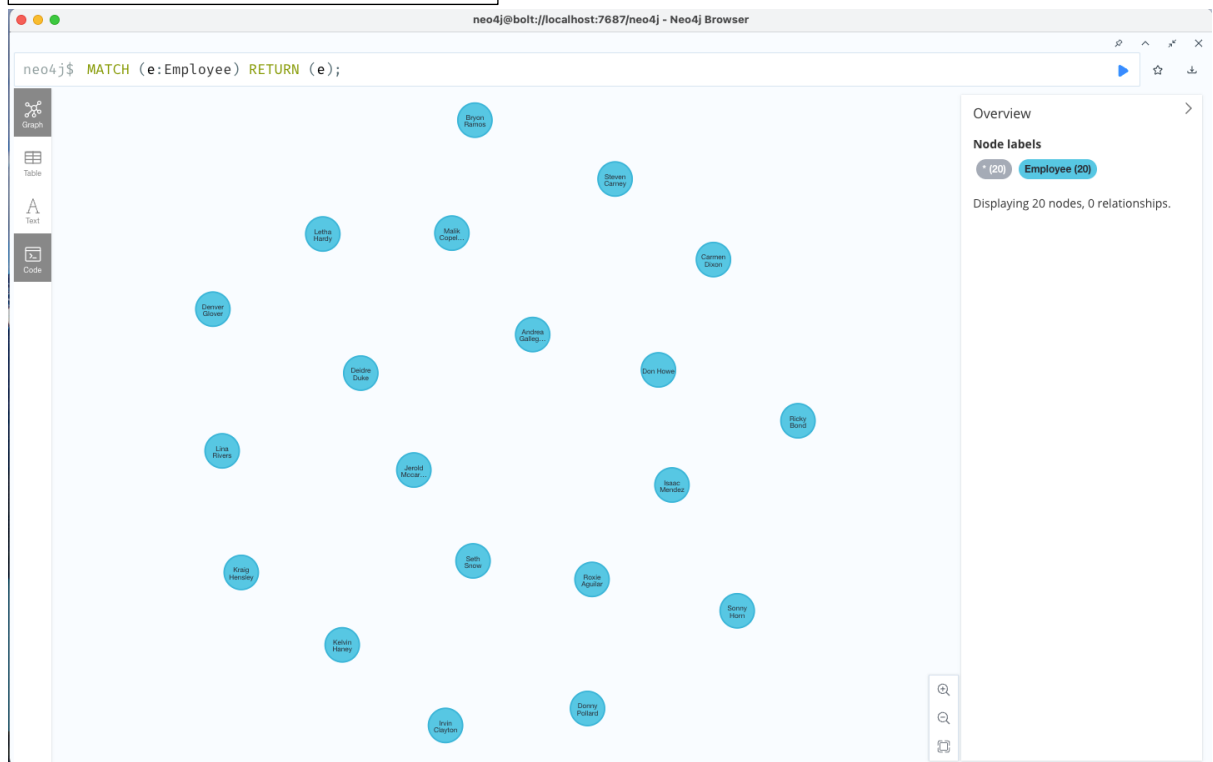


Fig 6: Screenshot of graph result contains nodes of all Employees

4. Execute the following Cypher code to get the list of customers. (0 point)

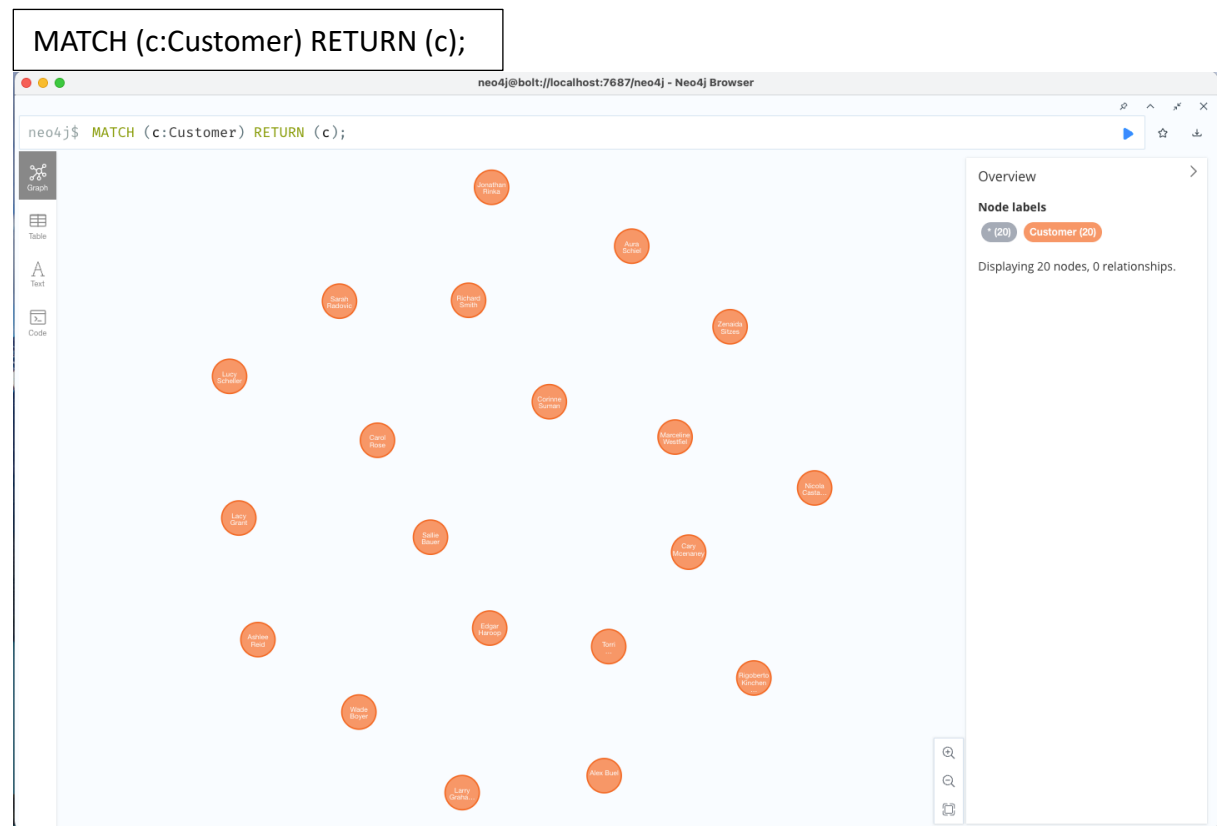


Fig 7: Screenshot of graph result contains nodes of all Customers

5. Execute the following Cypher code to get the list of all disputed transactions. (0 point)

```
MATCH (customer:Customer)-[transaction:SHOPPED_AT]->(retailer) WHERE
transaction.status = "Disputed"
RETURN customer.name AS `Customer Name`, retailer.name AS `Retailer Name`,
transaction.amount AS `Transaction Amount`,
transaction.date AS `Transaction date`
ORDER BY `Transaction date` DESC
```

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

1 MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(retailer) WHERE transaction.status = "Disputed"

2 RETURN customer.name AS `Customer Name`, retailer.name AS `Retailer Name`, transaction.amount AS `Transaction Amount`,

3 transaction.date AS `Transaction date`

4 ORDER BY `Transaction date` DESC

Table

Text

Code

	Customer Name	Retailer Name	Transaction Amount	Transaction date
1	" Nicola Castanon "	"Coach"	"721"	"7/17/2020"
2	" Zenaida Sitzes "	"Express"	"1884"	"5/7/2020"
3	" Marceline Westfield "	"Express"	"533"	"5/6/2020"
4	"Edgar Haroop"	"Neiman Marcus"	"1732"	"5/26/2020"
5	"Edgar Haroop"	"Kohls"	"1021"	"5/23/2020"
6	"Lucy Scheller"	"BestBuy"	"424"	"5/20/2020"
7	"Larry Grahamr"	"Walmart"	"425"	"5/19/2020"
8	"Larry Grahamr"	"Neiman Marcus"	"475"	"5/19/2020"
9	"Richard Smith"	"Kohls"	"875"	"5/13/2020"
10	" Rigoberto Kinchen "	"BestBuy"	"424"	"5/10/2020"
11	"Jonathan Rlinka"	"Neiman Marcus"	"375"	"4/19/2020"

12	"Torri Pettway "	"Foot Locker"	"62"	"4/17/2020"
13	"Carol Rose"	"Express"	"721"	"4/13/2020"
14	"Edgar Haroop"	"Nordstrom"	"1415"	"4/1/2020"
15	" Rigoberto Kinchen "	"Express"	"721"	"4/1/2020"
16	"Edgar Haroop"	"Walmart"	"654"	"3/20/2020"
17	" Rigoberto Kinchen "	"Walmart"	"914"	"3/18/2020"
18	" Zenaída Sitzes "	"Walmart"	"1149"	"3/18/2020"
19	"Richard Smith"	"Coach"	"1145"	"3/18/2020"
20	"Ashlee Reid"	"Walmart"	"1149"	"3/18/2020"
21	"Sarah Radovic"	"Nordstrom"	"516"	"3/15/2020"
22	" Aura Schiel "	"Neiman Marcus"	"830"	"3/13/2020"
23	" Cary Mcenaney "	"Kohls"	"468"	"2/29/2020"
24	"Edgar Haroop"	"Walmart"	"1849"	"2/20/2020"
25	" Rigoberto Kinchen "	"Nordstrom"	"1003"	"2/20/2020"
26	" Corinne Surman "	"Nordstrom"	"816"	"2/20/2020"
27	"Lacy Grant"	"Nordstrom"	"1003"	"2/20/2020"
28	"Jonathan Rinka"	"Kohls"	"1345"	"2/18/2020"
29	" Zenaída Sitzes "	"BestBuy"	"378"	"2/10/2020"
30	"Sallie Bauer"	"Foot Locker"	"378"	"2/10/2020"
31	" Torri Pettway "	"Target"	"605"	"1/27/2020"
32	"Jonathan Rinka"	"Walmart"	"945"	"1/27/2020"
33	" Zenaída Sitzes "	"Nordstrom"	"1790"	"1/20/2020"

Started streaming 33 records after 1 ms and completed after 3 ms.

Fig 8: Screenshot of list of all disputed transactions

6. Write the Cypher code to get the number of disputed transactions for every retailer.

The output should show the Retailer name and the number of disputes. Sort with highest number of disputes on top. (10 points)

```

MATCH (customer:Customer)-[transaction:SHOPPED_AT]->(r:Retailer) WHERE
transaction.status = "Disputed"
RETURN r.name AS `Retailer Name`, count(transaction) AS `Disputed Transaction
Count`
ORDER BY `Disputed Transaction Count` DESC

```

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

```

1 MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(r:Retailer) WHERE transaction.status = "Disputed"
2 RETURN r.name AS `Retailer Name`, count(transaction) AS `Disputed Transaction Count`
3 ORDER BY `Disputed Transaction Count` DESC

```

	Retailer Name	Disputed Transaction Count
1	"Walmart"	7
2	"Nordstrom"	6
3	"Express"	4
4	"Kohls"	4
5	"Neiman Marcus"	4
6	"BestBuy"	3
7	"Foot Locker"	2
8	"Coach"	2
9	"Target"	1

Started streaming 9 records in less than 1 ms and completed after 2 ms.

Fig 9: Screenshot of disputed transactions for every retailer

7. Write the Cypher code to get the number of disputed transactions and the list of customer names for these disputed transactions for every retailer. The output should show the Retailer and the customer name(s). You can consider using a collect() container, but it is not required. (10 points)

```

MATCH (c:Customer)-[transaction:SHOPPED_AT]-(r:Retailer) WHERE
transaction.status = "Disputed"
RETURN r.name AS `Retailer Name`, collect(c.name) AS `Customer Names`,
count(transaction) AS `Disputed Transaction Count`
ORDER BY `Disputed Transaction Count` DESC

```


neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

```

1 MATCH (c:Customer)-[transaction:SHOPPED_AT]->(r:Retailer) WHERE transaction.status = "Disputed"
2 RETURN r.name AS `Retailer Name`, collect(c.name) AS `Customer Names`, count(transaction) AS `Disputed Transaction Count`
3 ORDER BY `Disputed Transaction Count` DESC

```

	Retailer Name	Customer Names	Disputed Transaction Count
1	"Walmart"	["Ashlee Reid", "Edgar Haroop", "Jonathan Rinka", "Edgar Haroop", "Zenaída Sitzes", "Larry Grahamr", "Rigoberto Kinchen "]	7
2	"Nordstrom"	[" Zenaída Sitzes ", " Rigoberto Kinchen ", " Corinne Suman ", "Edgar Haroop", "Sarah Radovic", "Lacy Grant"]	6
3	"Express"	[" Marceline Westfield ", " Rigoberto Kinchen ", " Zenaída Sitzes ", "Carol Rose"]	4
4	"Kohls"	["Jonathan Rinka", "Richard Smith", " Cary Mcenaney ", "Edgar Haroop"]	4
5	"Neiman Marcus"	["Jonathan Rinka", "Edgar Haroop", " Aura Schiel ", "Larry Grahamr"]	4
6	"BestBuy"	["Lucy Scheller", " Rigoberto Kinchen ", " Zenaída Sitzes "]	3
7	"Foot Locker"	[" Torri Pettway ", "Sallie Bauer"]	2
8	"Coach"	[" Nicola Castanon ", "Richard Smith"]	2
9	"Target"	[" Torri Pettway "]	1

Started streaming 9 records in less than 1 ms and completed after 1 ms.

Fig 10: Screenshot of disputed transactions for every customer for every retailer

8. Write the Cypher code to get the number of disputed transactions for every customer that has more than one disputed transaction. (10 points)

```

MATCH (c:Customer)-[transaction:SHOPPED_AT]->(r:Retailer) WHERE
transaction.status = "Disputed"
WITH c.name AS `Customer Name`, count(transaction) AS `Disputed Transaction Count`
WHERE `Disputed Transaction Count` > 1
RETURN `Customer Name`, `Disputed Transaction Count`
ORDER BY `Disputed Transaction Count` DESC

```

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

```

1 MATCH (c:Customer)-[transaction:SHOPPED_AT]->(r:Retailer) WHERE transaction.status = "Disputed"
2 WITH c.name AS `Customer Name`, count(transaction) AS `Disputed Transaction Count`
3 WHERE `Disputed Transaction Count` > 1
4 RETURN `Customer Name`,`Disputed Transaction Count`
5 ORDER BY `Disputed Transaction Count` DESC

```

	Customer Name	Disputed Transaction Count
1	"Edgar Haroop"	5
2	"Zenaida Sitzes "	4
3	" Rigoberto Kinchen "	4
4	"Jonathan Rlinka"	3
5	" Torri Pettway "	2
6	"Larry Grahamr"	2
7	"Richard Smith"	2

Started streaming 7 records in less than 1 ms and completed after 5 ms.

Fig 11: Screenshot of customers which have more than 1 disputed transaction

9. Write the Cypher code to get the list of stores on LaSalle street that have disputed transactions and the number of disputed transactions for every store; the store list must be sorted by store name in ascending order. (10 points)

```

MATCH (c:Customer)-[transaction:SHOPPED_AT]->(r:Retailer)
WHERE transaction.status = "Disputed" AND r.street CONTAINS "LaSalle"
RETURN r.name AS `Retailer Name`,count(transaction) AS `Disputed Transaction Count`
ORDER BY `Retailer Name` ASC

```

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

```

1 MATCH (c:Customer)-[transaction:SHOPPED_AT]->(r:Retailer)
2 WHERE transaction.status = "Disputed" AND r.street CONTAINS "LaSalle"
3 RETURN r.name AS `Retailer Name`, count(transaction) AS `Disputed Transaction Count`
4 ORDER BY `Retailer Name` ASC
5

```

	Retailer Name	Disputed Transaction Count
1	"Neiman Marcus"	4
2	"Nordstrom"	6

Started streaming 2 records after 7 ms and completed after 8 ms.

Fig 12: Screenshot of list of Retailers on LaSalle street that have disputed transactions

10. Write the Cypher code to get the list of Employees who work in at least 2 stores where disputed transactions reported in these retailers. (10 points)

```

MATCH (e:Employee)-[:WORKS_AT]->(r:Retailer)<-[:SHOPPED_AT {status:
'Disputed'}]-(:Customer)
WITH e, COUNT(DISTINCT r) AS numStores
WHERE numStores >= 2
RETURN e.name AS `Employee Name`
ORDER BY `Employee Name`

```

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

```

1 MATCH (e:Employee)-[:WORKS_AT]-(r:Retailer)←[:SHOPPED_AT {status: 'Disputed'}]-(c:Customer)
2 WITH e, COUNT(DISTINCT r) AS numStores
3 WHERE numStores ≥ 2
4 RETURN e.name AS `Employee Name`
5 ORDER BY `Employee Name`
6

```

	Employee Name
1	"Bryon Ramos"
2	"Carmen Dixon"
3	"Irvin Clayton"
4	"Ricky Bond"

Started streaming 4 records after 1 ms and completed after 5 ms.

Fig 13: Screenshot of Employees who work in at least 2 stores where disputed transactions reported

11. Write the Cypher code to show the total amount customers spent shopping at retailers. List the customer's name and the total amount spent. (10 points)

```

MATCH (c:Customer)-[t:SHOPPED_AT]->(r:Retailer)
RETURN c.name AS `Customer Name`, SUM(toInteger(t.amount)) AS `Total Amount Spent`
ORDER BY `Total Amount Spent` DESC

```

neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

```

1 MATCH (c:Customer)-[t:SHOPPED_AT]-(r:Retailer)
2 RETURN c.name AS `Customer Name`, SUM(toInteger(t.amount)) AS `Total Amount Spend`
3 ORDER BY `Total Amount Spend` DESC

```

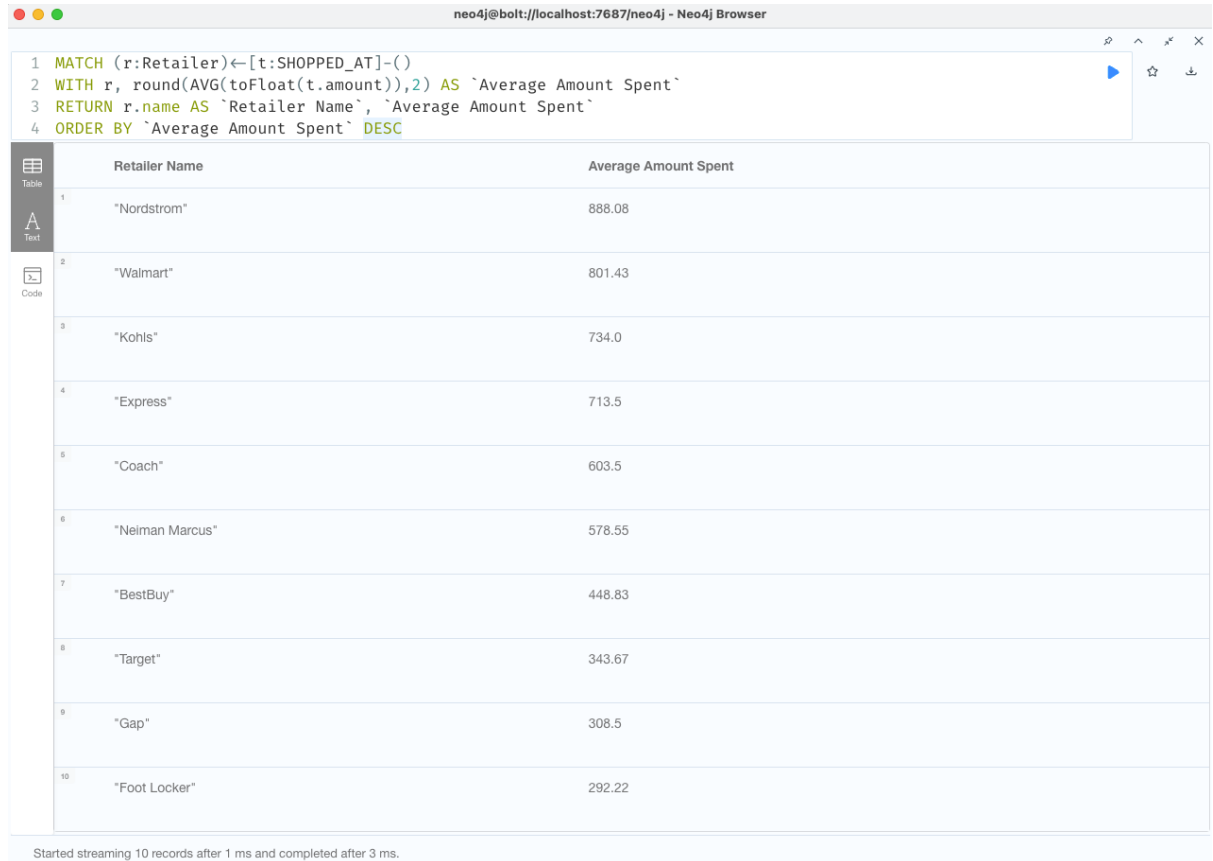
	Customer Name	Total Amount Spend
1	"Edgar Haroop"	8371
2	"Zenaïda Sitzes "	7172
3	" Rigoberto Kinchen "	4937
4	" Corinne Suman "	3722
5	" Cary Mcenaney "	3159
6	" Nicola Castanon "	2738
7	"Jonathan Rinka"	2665
8	" Marceline Westfield "	2651
9	" Alex Buel "	2551
10	"Richard Smith"	2285
11	" Aura Schiel "	2043
12	"Wade Boyer"	1884
13	"Ashlee Reid"	1762
14	"Lucy Scheller"	1272
15	"Larry Grahamr"	1224
16	"Lacy Grant"	1003
17	" Torri Pettway "	843
18	"Carol Rose"	721
19	"Sallie Bauer"	721
20	"Sarah Radovic"	516

Started streaming 20 records in less than 1 ms and completed after 1 ms.

Fig 14: Screenshot of total amount spent by customers on shopping at retailers

12. Write the Cypher code to show the average amount spent at each Retailer. List the Retailer and the average amount spent. Sort with highest amount on top. (10 points)

```
MATCH (r:Retailer) <- [t:SHOPPED_AT] - ()
WITH r, round(AVG(toFloat(t.amount)), 2) AS `Average Amount Spent`
RETURN r.name AS `Retailer Name`, `Average Amount Spent`
ORDER BY `Average Amount Spent` DESC
```



The screenshot shows the Neo4j Browser interface. At the top, the browser title is "neo4j@bolt://localhost:7687/neo4j - Neo4j Browser". Below the title bar, there is a code editor with the following Cypher query:

```
1 MATCH (r:Retailer) <- [t:SHOPPED_AT] - ()
2 WITH r, round(AVG(toFloat(t.amount)), 2) AS `Average Amount Spent`
3 RETURN r.name AS `Retailer Name`, `Average Amount Spent`
4 ORDER BY `Average Amount Spent` DESC
```

To the right of the code editor are icons for running the query (a blue play button), saving (a star), and downloading (a download icon). Below the code editor, there is a table view showing the results of the query. The table has two columns: "Retailer Name" and "Average Amount Spent". The results are ordered by "Average Amount Spent" in descending order.

	Retailer Name	Average Amount Spent
1	"Nordstrom"	888.08
2	"Walmart"	801.43
3	"Kohls"	734.0
4	"Express"	713.5
5	"Coach"	603.5
6	"Neiman Marcus"	578.55
7	"BestBuy"	448.83
8	"Target"	343.67
9	"Gap"	308.5
10	"Foot Locker"	292.22

At the bottom of the table view, there is a status message: "Started streaming 10 records after 1 ms and completed after 3 ms."

Fig 15: Screenshot to show the average amount spent at each Retailer