

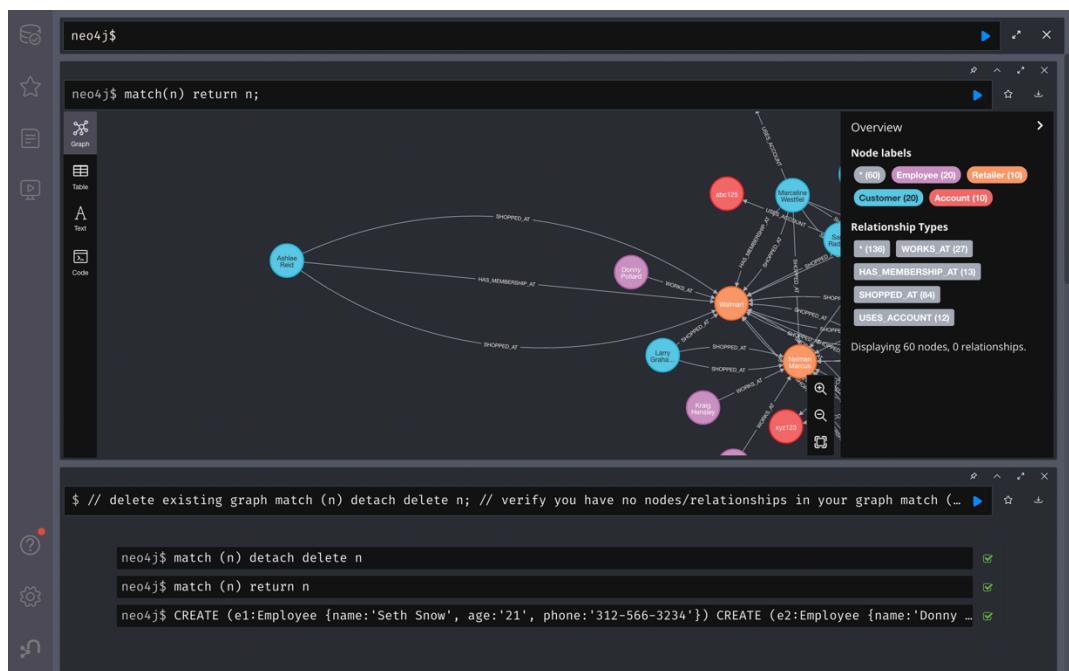
# Neo4j

## Cypher Query Language

### Fraud Detection

1.

- a. You could use any names for your project and the graph database
- b. Copy the **ENTIRE** Cypher code in the script and paste it in ne4oj\$ prompt and then click the blue play button on the right.
- c. (DO NOT copy and paste one line at a time)
- d. Run the command below. Find the Customer Ashlee Reid and pull the node to the far left of the screen. Include a screencapture of this view to show you were able to load the database (**6 points**)



2. Execute the following Cypher code to get the list of retailers: (1 point) **MATCH (r:Retailer) RETURN (r);**



3. Execute the following Cypher code to get the list of employees: (1 point) **MATCH (e:Employee) RETURN (e);**



4. Execute the following Cypher code to get the list of customers: (1 point) `MATCH (c:Customer)RETURN (c);`



5. Execute the following Cypher code to get the list of all disputed transactions: (1 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 `TransactionAmount`, transaction.date AS `Transaction date` ORDER BY `Transaction date` DESC`

neo4j\$

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" RETURN cus...
```

Customer Name	Retailer Name	TransactionAmount	Transaction date
" Nicola Castanon "	"Coach"	"721"	"7/17/2020"
" Zenaida Sitzes "	"Express"	"1884"	"5/7/2020"
" Marceline Westfield "	"Express"	"533"	"6/6/2020"
"Edgar Haroop"	"Neiman Marcus"	"1732"	"5/26/2020"
"Edgar Haroop"	"Kohls"	"1021"	"5/23/2020"
"Lucy Scheller"	"BestBuy"	"424"	"5/20/2020"

Started streaming 33 records after 6 ms and completed after 16 ms.

neo4j\$ MATCH (n) RETURN (n);

Overview

Node labels

- (60) Employee (20)
- Retailer (10)
- Customer (20)
- Account (10)

Relationship Types

- (136) WORKS\_AT (27)

**6. Write the Cypher code to get the number of disputed transactions for every retailer (6 points)**

The screenshot shows the Neo4j browser interface with two separate query windows.

**Query 1:**

```
1 MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(retailer) WHERE transaction.status = "Disputed"
2 RETURN retailer.name, count(*)
```

**Result 1:**

retailer.name	count(*)
"Walmart"	7
"Nordstrom"	6
"Neiman Marcus"	4
"Kohls"	4
"Foot Locker"	2
"Target"	1

Started streaming 9 records after 2 ms and completed after 4 ms.

**Query 2:**

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(retailer) WHERE transaction.status = "Disputed" RETURN cus...
```

**Result 2:**

Customer Name	Retailer Name	TransactionAmount	Transaction date
"Jonathan Rinka"	"Neiman Marcus"	"375"	"4/19/2020"
"Torri Pettway"	"Foot Locker"	"62"	"4/17/2020"

```

neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(retailer) WHERE transaction.status = "Disputed" RETURN retailer.name, count(*)
  
```

retailer.name	count(*)
"Kohls"	4
"Foot Locker"	2
"Target"	1
"Express"	4
"BestBuy"	3
"Coach"	2

Started streaming 9 records after 2 ms and completed after 4 ms.

```

neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(retailer) WHERE transaction.status = "Disputed" RETURN customer.name, retailer.name, transaction.amount, transaction.date
  
```

Customer Name	Retailer Name	TransactionAmount	Transaction date
"Jonathan Rinka"	"Neiman Marcus"	"375"	"4/19/2020"
" Torri Pettway "	"Foot Locker"	"62"	"4/17/2020"

## 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

```

neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(retailer) WHERE transaction.status = "Disputed" RETURN customer.name, retailer.name, count(*)
  
```

customer.name	retailer.name	count(*)
"Edgar Haroop"	"Walmart"	2
"Edgar Haroop"	"Nordstrom"	1
"Edgar Haroop"	"Neiman Marcus"	1
"Edgar Haroop"	"Kohls"	1
" Torri Pettway "	"Foot Locker"	1
" Torri Pettway "	"Target"	1

Started streaming 32 records after 4 ms and completed after 24 ms.

```

neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]→(retailer) WHERE transaction.status = "Disputed" RETURN retailer.name, count(*)
  
```

retailer.name	count(*)
"Kohls"	4
"Foot Locker"	2

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" RETURN customer.name,retailer.name,count(*)
```

customer.name	retailer.name	count(*)
" Rigoberto Kinchen "	"Nordstrom"	1
" Rigoberto Kinchen "	"Express"	1
" Rigoberto Kinchen "	"BestBuy"	1
" Rigoberto Kinchen "	"Walmart"	1
" Cary Mcenaney "	"Kohls"	1
" Nicola Castanon "	"Coach"	1

Started streaming 32 records after 4 ms and completed after 24 ms.

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" RETURN retailer.name, count(*)
```

retailer.name	count(*)
"Kohls"	4
"Foot Locker"	2

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" RETURN customer.name,retailer.name,count(*)
```

customer.name	retailer.name	count(*)
" Corinne Suman "	"Nordstrom"	1
" Jonathan Rinka "	"Neiman Marcus"	1
" Jonathan Rinka "	"Kohls"	1
" Jonathan Rinka "	"Walmart"	1
" Richard Smith "	"Coach"	1
" Richard Smith "	"Kohls"	1

Started streaming 32 records after 4 ms and completed after 24 ms.

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" RETURN retailer.name, count(*)
```

retailer.name	count(*)
"Kohls"	4
"Foot Locker"	2

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" RETURN customer.name,retailer.name,count(*)
```

customer.name	retailer.name	count(*)
"Corinne Suman"	"Nordstrom"	1
"Jonathan Rinka"	"Neiman Marcus"	1
"Jonathan Rinka"	"Kohls"	1
"Jonathan Rinka"	"Walmart"	1
"Richard Smith"	"Coach"	1
"Richard Smith"	"Kohls"	1

Started streaming 32 records after 4 ms and completed after 24 ms.

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" RETURN retailer.name, count(*)
```

retailer.name	count(*)
"Kohls"	4
"Foot Locker"	2

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" RETURN customer.name,retailer.name,count(*)
```

customer.name	retailer.name	count(*)
"Sarah Radovic"	"Nordstrom"	1
"Lucy Scheller"	"BestBuy"	1
"Carol Rose"	"Express"	1
"Lacy Grant"	"Nordstrom"	1
"Ashlee Reid"	"Walmart"	1
"Sallie Bauer"	"Foot Locker"	1

Started streaming 32 records after 4 ms and completed after 24 ms.

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" RETURN retailer.name, count(*)
```

retailer.name	count(*)
"Kohls"	4
"Foot Locker"	2

```

neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed"
2 RETURN customer.name, retailer.name, count(*)

```

	customer.name	retailer.name	count(*)
27	"Carol Rose"	"Express"	1
28	"Lacy Grant"	"Nordstrom"	1
29	"Ashlee Reid"	"Walmart"	1
30	"Sallie Bauer"	"Foot Locker"	1
31	"Larry Graham"	"Walmart"	1
32	"Larry Graham"	"Neiman Marcus"	1

Started streaming 32 records after 4 ms and completed after 24 ms.

```

neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" RETURN retailer.name, count(*)

```

	retailer.name	count(*)
4	"Kohls"	4
5	"Foot Locker"	2

**8. Write the Cypher code to get the number of disputed transactions for every customer that has more than one disputed transaction (6 points)**

9. **MATCH** (customer:Customer)-[transaction:SHOPPED\_AT]->(retailer)
10. **WHERE** transaction.status = "Disputed"
11. **WITH** customer.name **AS** `Customer Name`, **COUNT (\*) AS** `Number of Disputed Transactions`
12. **WHERE** `Number of Disputed Transactions` > 1
13. **RETURN** `Customer Name`, `Number of Disputed Transactions`

```
neo4j$ MATCH (customer:Customer)-[transaction:SHOPPED_AT]-(retailer) WHERE transaction.status = "Disputed" WITH custo... ▶
```

	Customer Name	# of Disputed Transactions
A	" Torri Pettway "	2
B	" Rigoberto Kinchen "	4
C	" Zenaida Sitzes "	4
D	"Jonathan Rinka"	3
E	"Richard Smith"	2
F	"Larry Grahamr"	2

Started streaming 7 records after 1 ms and completed after 3 ms.

9. Write the Cypher code to get the list of stores on La Salle 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.

