Master of Science in Data Analytics

Dublin Business School

CA

Data Storage Solutions for Data Analytics

Course Code: B9DA111_2122_TMD3

Submitted By: Saima Saleem 10603707

Individual Report

Submitted To: Bernie Lydon

Table of Contents

Group Task

My group task is to work on Neo4J and to compare graph and relational databases with the consultation with other group member who is working on SQL part. Rest was to work on presentation collectively. I enjoyed my group task.

Database Description

The database "Bike Store" has the following tables:

- brand: brand_id, brand_name. There are 9 brands.
- category: category_id, category_name. There are 7 categories.
- customer: customer_id, first_name, last_name, phone, email, street, city, state, zip_code. There are 1445 customers based in 3 states i.e. NY, TX, CA
- employee: staff_id, first_name, last_name, email, phone, active, store_id, manager_id. Total number of employees is 10.
- order: order_id, customer_id, order_status, order_date, required_date, shipped_date, store_id, staff_id. Total number of orders are 1615. Ordered years are 2016, 2017, 2018
- product:product_id, product_name, brand_id, category_id, model_year, list_price. There are 321 products.
- store: store_id, store_name, phone. Email, street, city, state, zip_code. There are 3 stores with the names Santa Cruz Bikes, Baldwin Bikes, Rowlett Bikes.
- order_items: order_id, item_id, product_id, quantity, list_price, discount

Database Creation

Local Database "Graph DBMS_BikeStore" was created.

Importing The Tables

The tables csv files were placed in the import folder.

Loading Nodes

Store Node

```
neo4j$ LOAD CSV WITH HEADERS FROM 'file:///store.csv'AS    row CREATE
       (n:Store)SET n=row
```

neo4j$ MATCH (n:Store) RETURN n LIMIT 25

Graph

Table

Text

Code

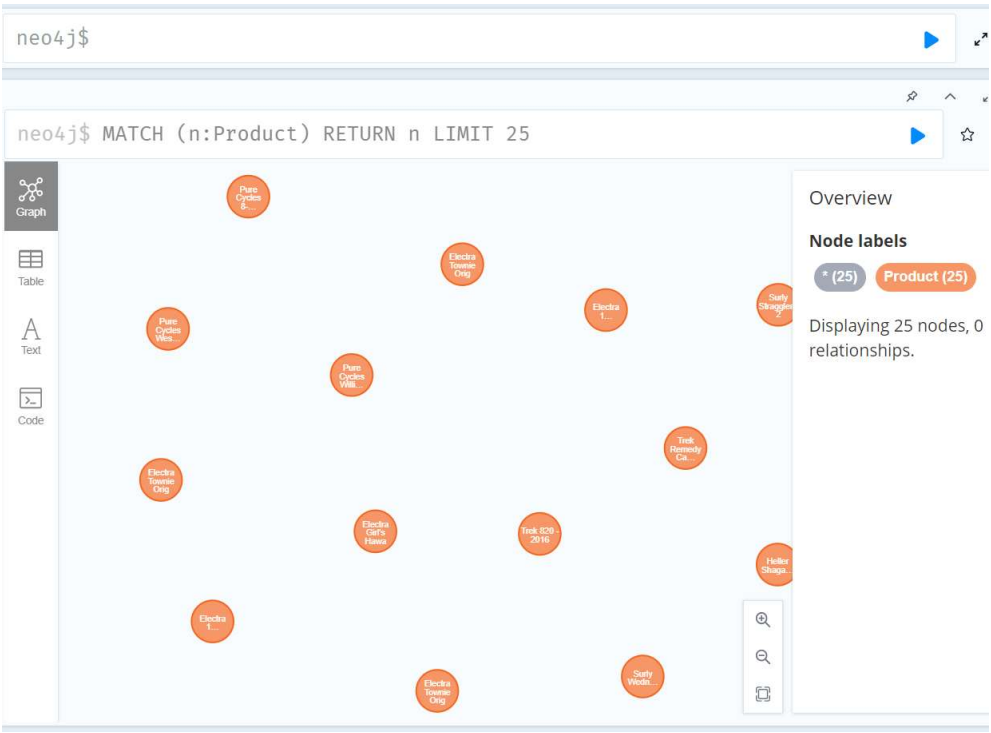
n

1


```
{
  "identity": 0,
  "labels": [
    "Store"
  ],
  "properties": {
"store_id": "1",
"phone": "(831) 476-4321",
"city": "Santa Cruz",
"street": "3700 Portola Drive",
"store_name": "Santa Cruz Bikes",
"state": "CA",
"email": "santacruz@bikes.shop",
"zip_code": "95060"
  }
}
```

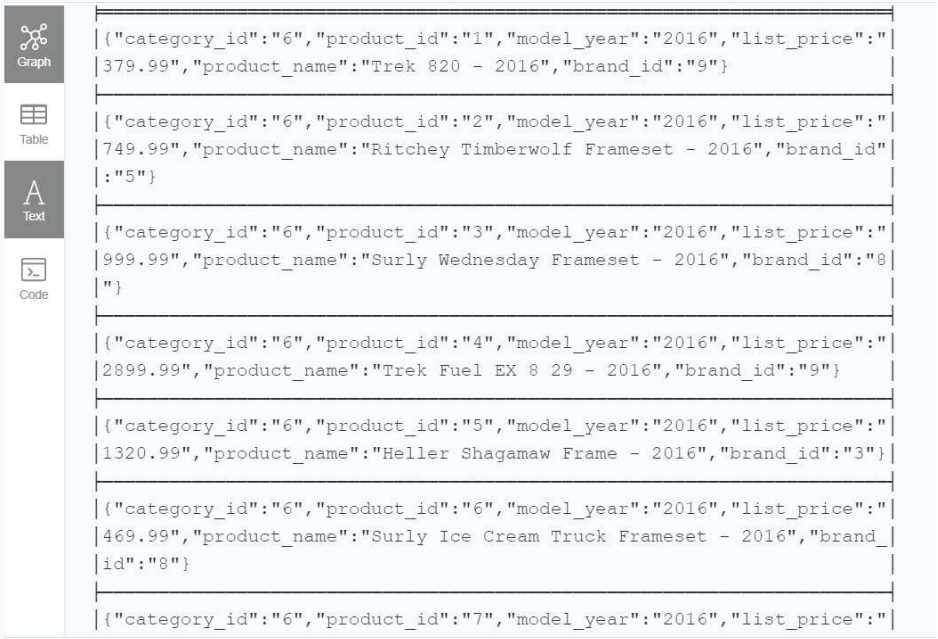
Product Node

```
neo4j$ LOAD CSV WITH HEADERS FROM 'file:///product.csv'AS row CREATE
       (n:Product)SET n=row
```



```
neo4j$ MATCH (n:Product) RETURN n LIMIT 25
```


```
neo4j$ MATCH (n:Product) RETURN n LIMIT 25
```



Order Node

```
neo4j$ LOAD CSV WITH HEADERS FROM 'file:///order.csv'AS row CREATE
       (n:Order)SET n=row
```

```
neo4j$
```

```
neo4j$ LOAD CSV WITH HEADERS FROM 'file:///order.csv'AS row CREATE (n:…
```

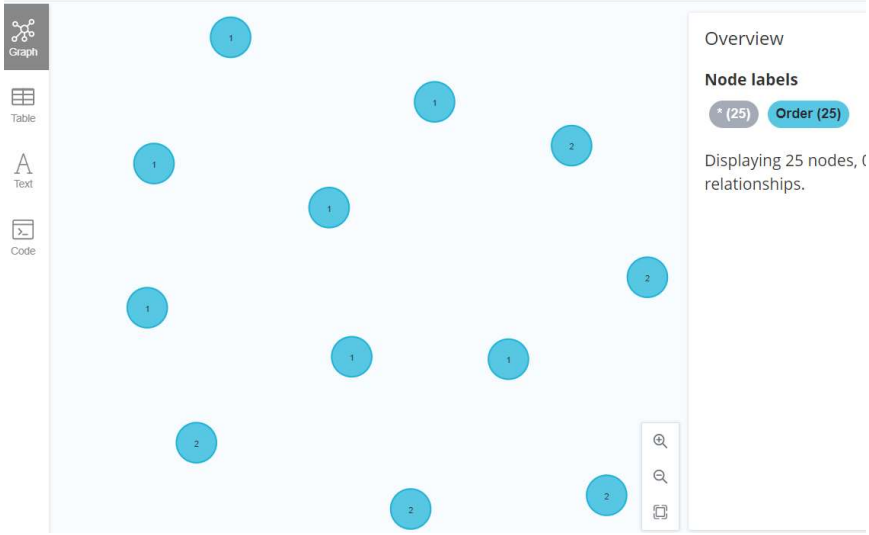
Table

Code

Added 1615 labels, created 1615 nodes, set 12920 properties, completed after 90 ms.

```
neo4j$
```

```
neo4j$ MATCH (n:Order) RETURN n LIMIT 25
```




```
neo4j$
```

```
neo4j$ MATCH (n:Order) RETURN n LIMIT 25
```

Graph

Table

Text

Code

n

1


```
{
  "identity": 324,
  "labels": [
    "Order"
  ],
  "properties": {
"store_id": "1",
"order_status": "4",
"order_date": "2016-01-01",
"required_date": "2016-01-03",
"shipped_date": "2016-01-03",
"staff_id": "2",
"customer_id": "259",
"order_id": "1"
  }
}
```

```
neo4j$ MATCH (n:Order) RETURN n LIMIT 25
```

Graph | Table | Text | Code

```
╒══════════════════════════════════════════════════════════════════════════════╕
|"n"                                                                           |
╞══════════════════════════════════════════════════════════════════════════════╡
|{"store_id":"1","order_status":"4","order_date":"2016-01-01","required        |
|_date":"2016-01-03","shipped_date":"2016-01-03","staff_id":"2","custom        |
|er_id":"259","order_id":"1"}                                                  |
├──────────────────────────────────────────────────────────────────────────────┤
|{"store_id":"2","order_status":"4","order_date":"2016-01-01","required        |
|_date":"2016-01-04","shipped_date":"2016-01-03","staff_id":"6","custom        |
|er_id":"1212","order_id":"2"}                                                 |
├──────────────────────────────────────────────────────────────────────────────┤
|{"store_id":"2","order_status":"4","order_date":"2016-01-02","required        |
|_date":"2016-01-05","shipped_date":"2016-01-03","staff_id":"7","custom        |
|er_id":"523","order_id":"3"}                                                  |
├──────────────────────────────────────────────────────────────────────────────┤
|{"store_id":"1","order_status":"4","order_date":"2016-01-03","required        |
|_date":"2016-01-04","shipped_date":"2016-01-05","staff_id":"3","custom        |
|er_id":"175","order_id":"4"}                                                  |
├──────────────────────────────────────────────────────────────────────────────┤
|{"store_id":"2","order_status":"4","order_date":"2016-01-03","required        |
|_date":"2016-01-06","shipped_date":"2016-01-06","staff_id":"6","custom        |
|er_id":"1324","order_id":"5"}                                                 |
└──────────────────────────────────────────────────────────────────────────────┘
```

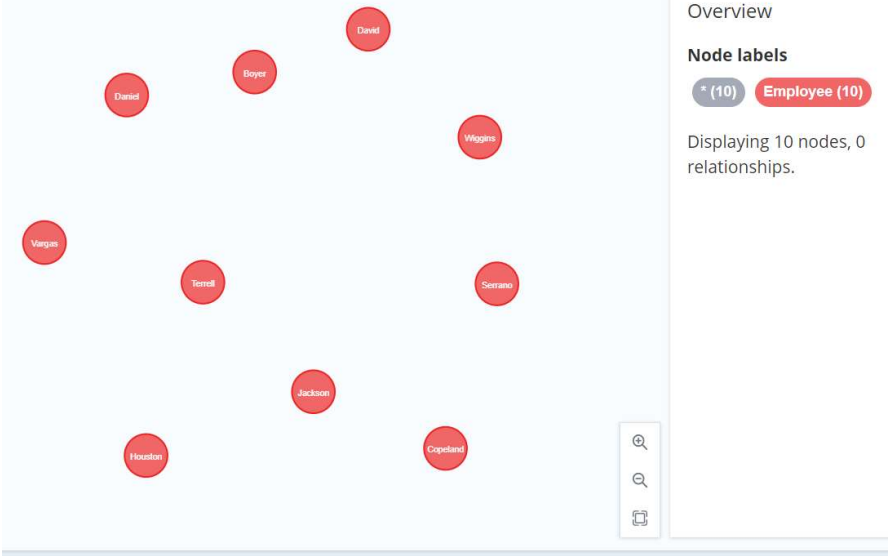
Employee Node

```
neo4j$ LOAD CSV WITH HEADERS FROM 'file:///employee.csv'AS row
       CREATE (n:Employee)SET n=row
```

Table

Added 10 labels, created 10 nodes, set 79 properties, completed after 24 ms.

```
$ MATCH (n:Employee) RETURN n LIMIT 25
```



```
neo4j$ MATCH (n:Employee) RETURN n LIMIT 25
```


```
neo4j$ MATCH (n:Employee) RETURN n LIMIT 25
```

Customer Node

```
neo4j$ LOAD CSV WITH HEADERS FROM 'file:///customer.csv'AS row
       CREATE (n:Customer)SET n=row
```

Table

Added 1445 labels, created 1445 nodes, set 13005 properties, completed after 90 ms.

```
neo4j$ MATCH (n:Customer) RETURN n LIMIT 25
```

Graph | Table | Text | Code

n

1


```
{
  "identity": 1949,
  "labels": [
    "Customer"
  ],
  "properties": {
"city": "Orchard Park",
"street": "9273 Thorne Ave. ",
"last_name": "Burks",
"state": "NY",
"customer_id": "1",
"first_name": "Debra",
"email": "debra.burks@yahoo.com",
"zip_code": "14127"
  }
}
```


Graph | Table | Text | Code

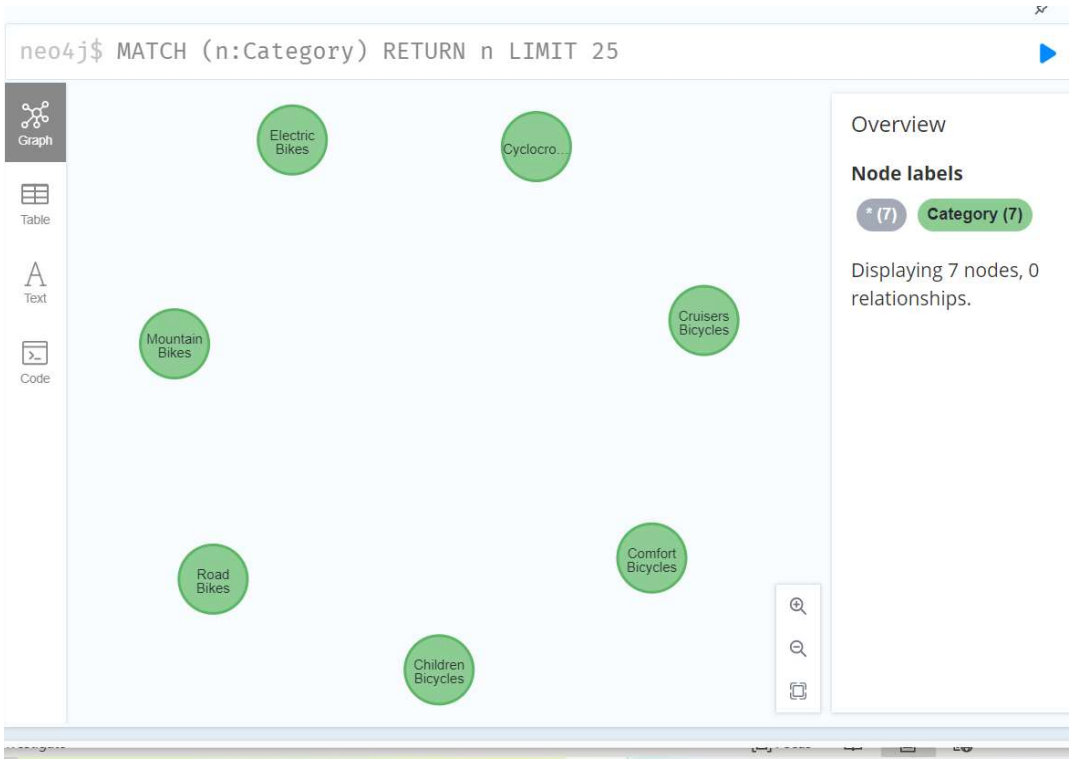

```
╟──────────────────────────────────────────────────────────────────────────────╢
|{"city":"Orchard Park","street":"9273 Thorne Ave. ","last_name":"Burks|
|","state":"NY","customer_id":"1","first_name":"Debra","email":"debra.b|
|urks@yahoo.com","zip_code":"14127"}                                          |
╟──────────────────────────────────────────────────────────────────────────────╢
|{"city":"Campbell","street":"910 Vine Street ","last_name":"Todd","sta|
|te":"CA","customer_id":"2","first_name":"Kasha","email":"kasha.todd@ya|
|hoo.com","zip_code":"95008"}                                                 |
╟──────────────────────────────────────────────────────────────────────────────╢
|{"city":"Redondo Beach","street":"769C Honey Creek St. ","last_name":"|
|Fisher","state":"CA","customer_id":"3","first_name":"Tameka","email":"|
|tameka.fisher@aol.com","zip_code":"90278"}                                   |
╟──────────────────────────────────────────────────────────────────────────────╢
|{"city":"Uniondale","street":"988 Pearl Lane ","last_name":"Spence","s|
|tate":"NY","customer_id":"4","first_name":"Daryl","email":"daryl.spenc|
|e@aol.com","zip_code":"11553"}                                               |
╟──────────────────────────────────────────────────────────────────────────────╢
|{"phone":"(916) 381-6003","city":"Sacramento","street":"107 River Dr. |
|","last_name":"Rice","state":"CA","customer_id":"5","first_name":"Char|
|olette","email":"charolette.rice@msn.com","zip_code":"95820"}                |
╟──────────────────────────────────────────────────────────────────────────────╢
```

Category Node

```
neo4j$ LOAD CSV WITH HEADERS FROM 'file:///category.csv'AS row
       CREATE (n:Category)SET n=row
```

Table

Added 7 labels, created 7 nodes, set 14 properties, completed after 16 ms.




Graph
Table
Text
Code
n
1
{
"identity": 3394,
"labels": [
"Category"
],
"properties": {
"category_name": "Children Bicycles",
"category_id": "1"
}
}
2
{
"identity": 3395,
"labels": [
"Category"


eo4j$ MATCH (n:Category) RETURN n LIMIT 25
"n"
{"category_name":"Children Bicycles","category_id":"1"}
{"category_name":"Comfort Bicycles","category_id":"2"}
{"category_name":"Cruisers Bicycles","category_id":"3"}
{"category_name":"Cyclocross Bicycles","category_id":"4"}
{"category_name":"Electric Bikes","category_id":"5"}
{"category_name":"Mountain Bikes","category_id":"6"}
{"category_name":"Road Bikes","category_id":"7"}
MAX COLUMN WIDTH:

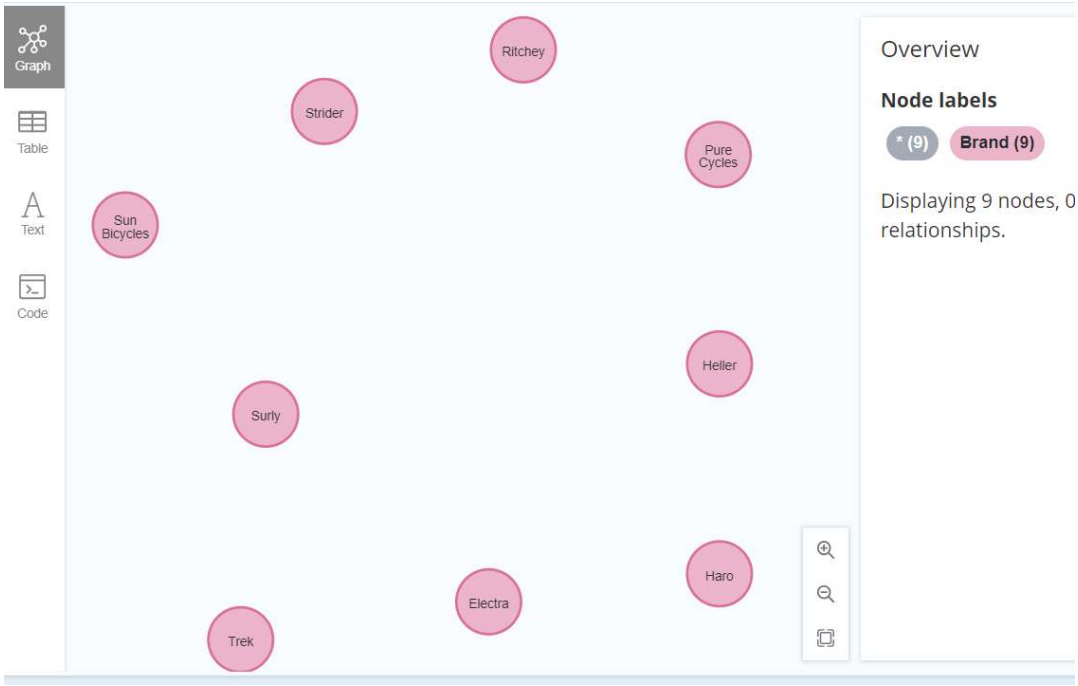
Brand Node

```
neo4j$ LOAD CSV WITH HEADERS FROM 'file:///brand.csv'AS row CREATE
       (n:Brand)SET n=row
```

Table

Added 9 labels, created 9 nodes, set 18 properties, completed after 16 ms.

```
neo4j$ MATCH (n:Brand) RETURN n LIMIT 25
```



```
neo4j$ MATCH (n:Brand) RETURN n LIMIT 25
```

n

1


```
{
  "identity": 3401,
  "labels": [
    "Brand"
  ],
  "properties": {
"brand_name": "Electra",
"brand_id": "1"
  }
}
```


2


```
{
  "identity": 3402,
  "labels": [
    "Brand"
  ],
```

```
neo4j$ MATCH (n:Brand) RETURN n LIMIT 25
```

Graph
Table
Text
Code

```
╒═══════════════════════════════════════════════════╕
"n"                                                 
╞═══════════════════════════════════════════════════╡
{"brand_name":"Electra","brand_id":"1"}             
├───────────────────────────────────────────────────┤
{"brand_name":"Haro","brand_id":"2"}                
├───────────────────────────────────────────────────┤
{"brand_name":"Heller","brand_id":"3"}              
├───────────────────────────────────────────────────┤
{"brand_name":"Pure Cycles","brand_id":"4"}         
├───────────────────────────────────────────────────┤
{"brand_name":"Ritchey","brand_id":"5"}             
├───────────────────────────────────────────────────┤
{"brand_name":"Strider","brand_id":"6"}             
├───────────────────────────────────────────────────┤
{"brand_name":"Sun Bicycles","brand_id":"7"}        
├───────────────────────────────────────────────────┤
{"brand_name":"Surly","brand_id":"8"}               
├───────────────────────────────────────────────────┤
{"brand_name":"Trek","brand_id":"9"}                
└───────────────────────────────────────────────────┘
```

Order_Items Node

```
LOAD CSV WITH HEADERS FROM 'file:///order_items.csv'AS row
CREATE (n:Order_Items)SET n=row
```

Added 4722 labels, created 4722 nodes, set 28332 properties, completed after 938 ms.


```
{"quantity":"1","item_id":"1","product_id":"20","discount":".20","list_price":"599.99","order_id":"1"}
{"quantity":"2","item_id":"2","product_id":"8","discount":".07","list_price":"1799.99","order_id":"1"}
{"quantity":"2","item_id":"3","product_id":"10","discount":".05","list_price":"1549.00","order_id":"1"}
{"quantity":"2","item_id":"4","product_id":"16","discount":".05","list_price":"599.99","order_id":"1"}
{"quantity":"1","item_id":"5","product_id":"4","discount":".20","list_price":"2899.99","order_id":"1"}
{"quantity":"1","item_id":"1","product_id":"20","discount":".07","list_price":"599.99","order_id":"2"}
{"quantity":"2","item_id":"2","product_id":"16","discount":".05","list
```


Now all the eight nodes have been loaded as shown in the fig. below:

Indexes

Product (product_id)

```
neo4j$ CREATE INDEX ON : Product(product_id)
```

Added 1 index, completed after 89 ms.

Store(store_id)

```
eo4j$ CREATE INDEX ON : Store(store_id)
```

Added 1 index, completed after 8 ms.

Order(order_id)

```
$ CREATE INDEX ON : Order(order_id)
```

Added 1 index, completed after 8 ms.

Employee(staff_id)

```
 CREATE INDEX ON : Employee(staff_id)
```

dded 1 index, completed in less than 1 ms.

Customer(customer_id)

```
 CREATE INDEX ON : Customer(customer_id)
```

Added 1 index, completed in less than 1 ms.

Category(category_id)

```
j$ CREATE INDEX ON : Category(category_id)
```

Added 1 index, completed in less than 1 ms.

Brand(brand_id)

```
j$ CREATE INDEX ON : Brand(brand_id)
```

Added 1 index, completed after 16 ms.

Order_Items(item_id)

```
$ CREATE INDEX ON :Order_Items(item_id)
```

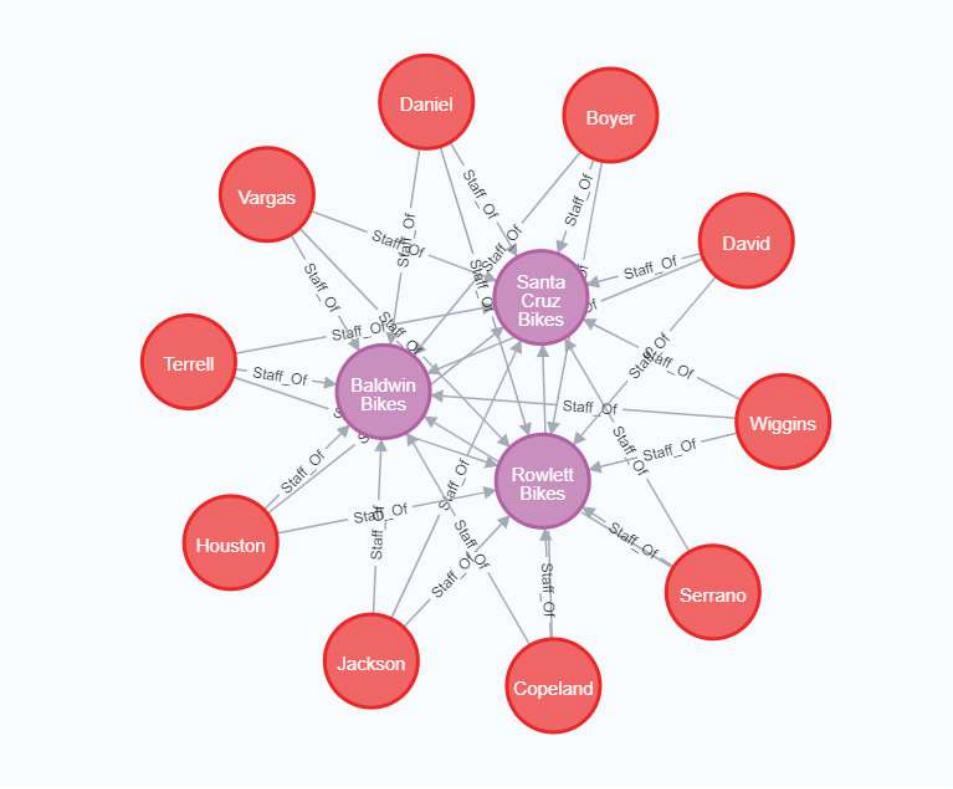
Added 1 index, completed after 112 ms.

Relationships

Employee and Store

```
MATCH(e:Employee),(s:Store) WHERE e: store_id=s:store_id
CREATE (e) - [ :Staff_Of]→(s)
```

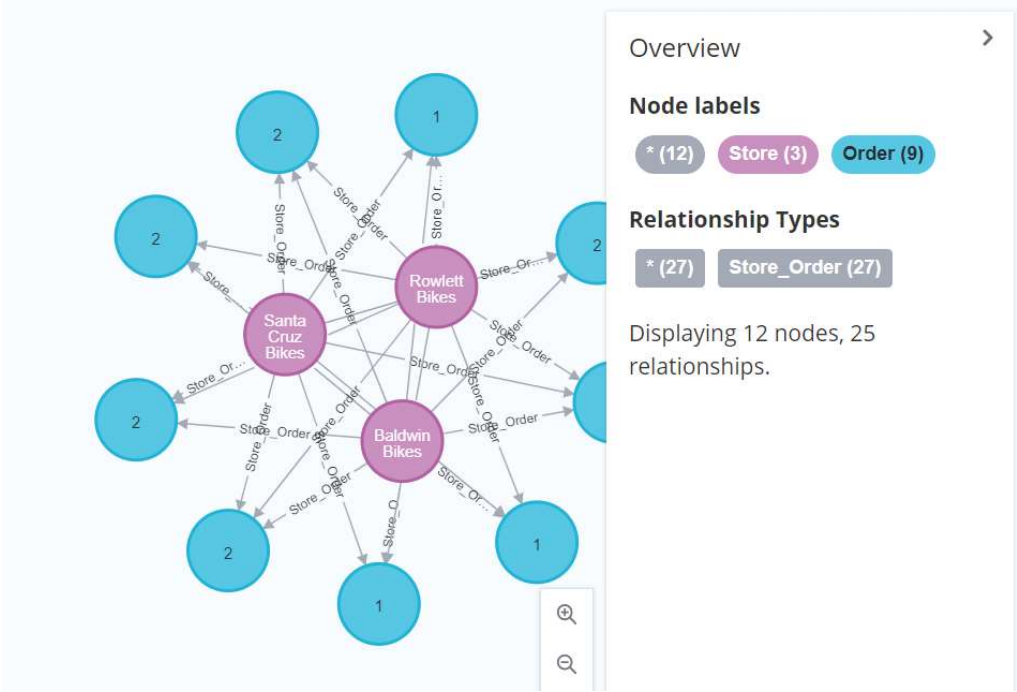
Created 30 relationships, completed after 16 ms.



Store and Order

```
MATCH(s:Store),(o:Order) WHERE s: store_id=o:store_id CREATE
(s) - [ :Store_Order]→(o)
```

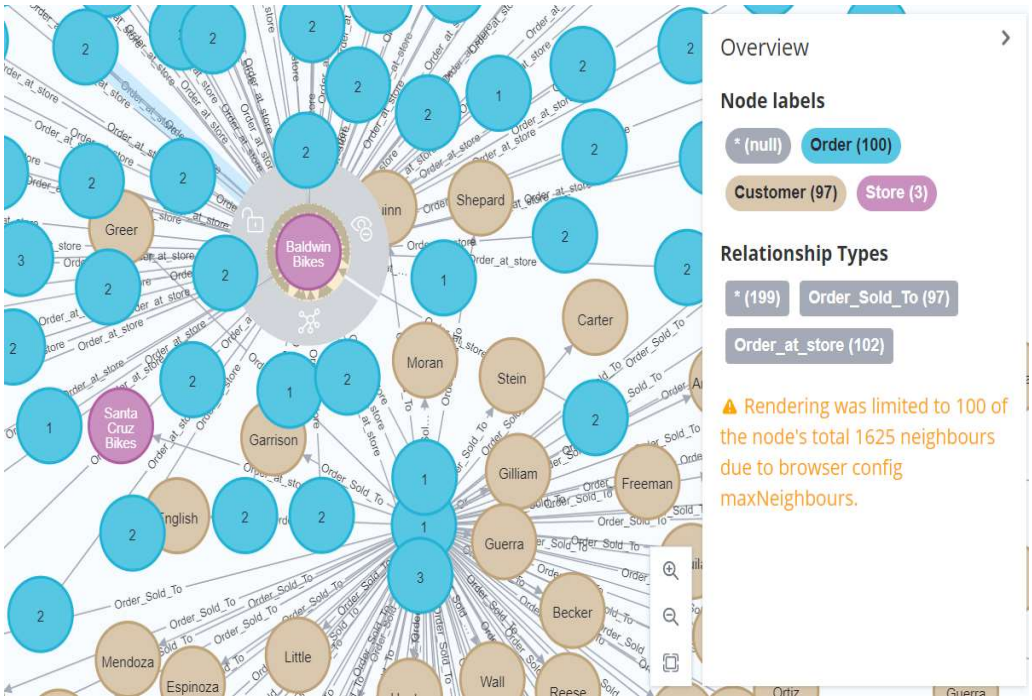
Created 4845 relationships, completed after 54 ms.



Order and Customer (Order_Sold_To)

```
MATCH(c:Customer),(o:Order) WHERE o:customer_id=c:customer_id
CREATE (o) - [ :Order_Sold_To]→(c)
```

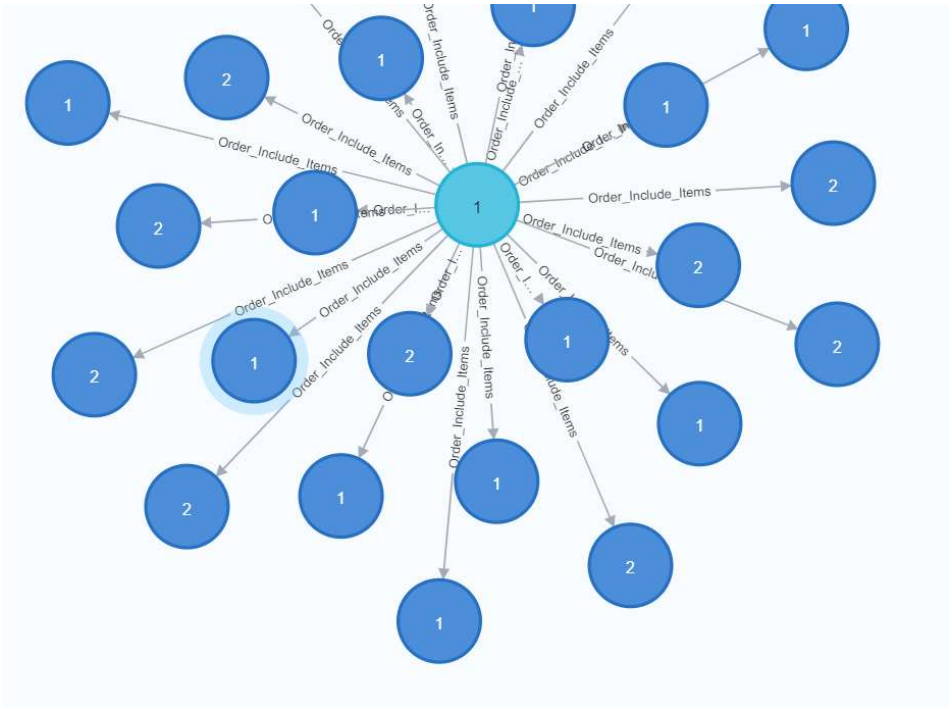
reated 2333675 relationships, completed after 8531 ms.



Order and Order_Item (Order_Include_Items)

```
MATCH(f:Order),(d:Order_Items) WHERE d: order_id=f:order_id CREATE (f) - [
:Order_Include_Items]→(d)
```

reated 7626030 relationships, completed after 14646 ms.



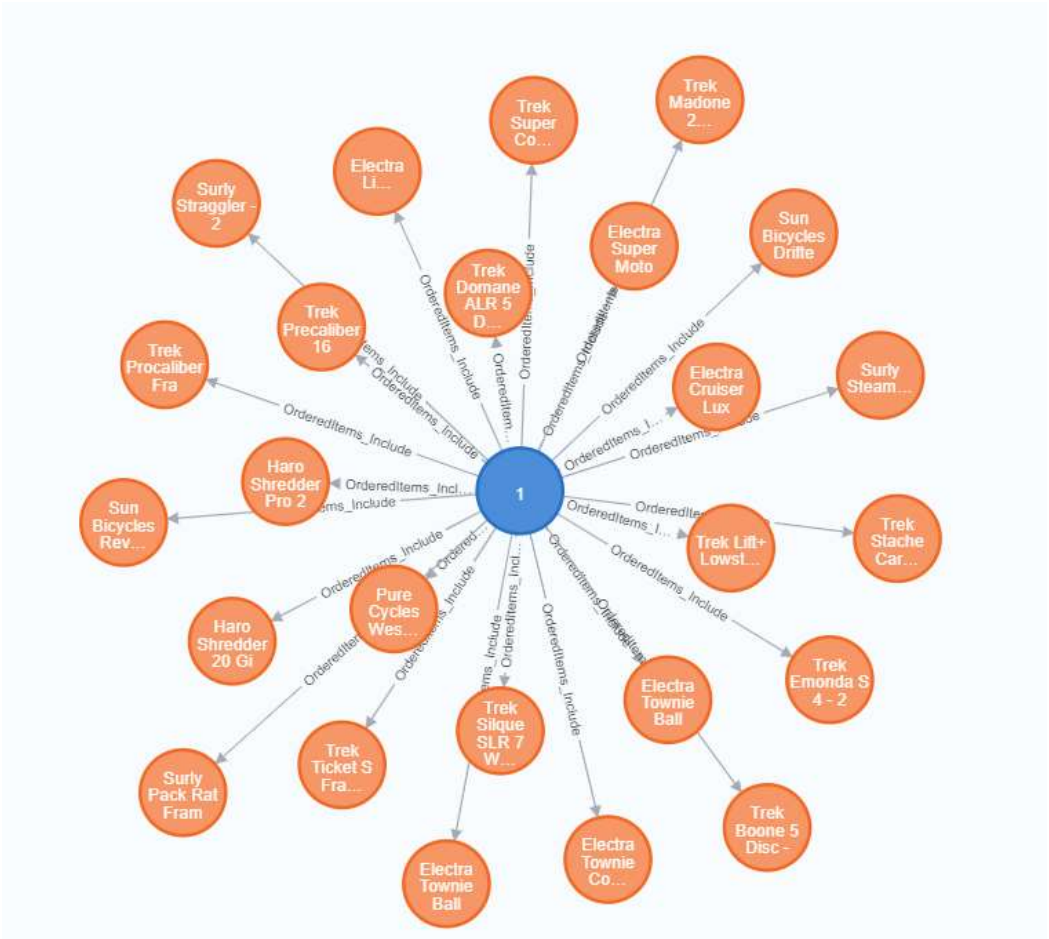

```
j$ MATCH p=()-[r:Order_Include_Items]→() RETURN p LIMIT 25
```

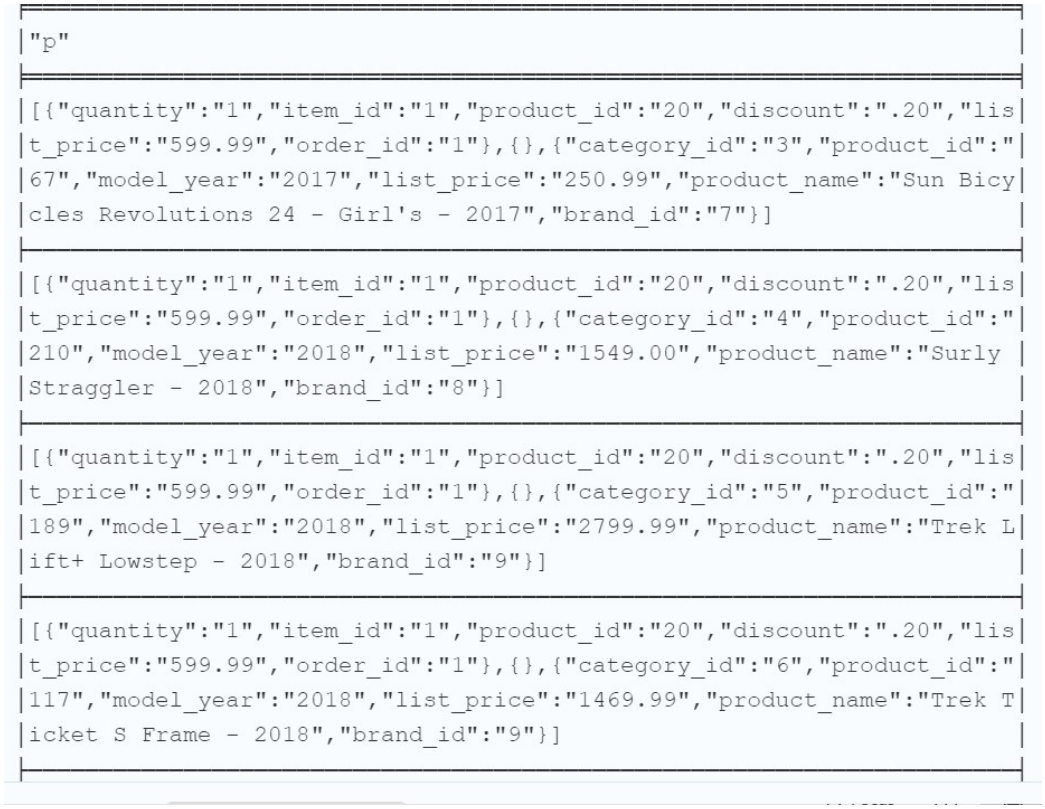
```
╒══════════════════════════════════════════════════════════════════════════════╕
"p"
╞══════════════════════════════════════════════════════════════════════════════╡
[{"store_id":"1","order_status":"4","order_date":"2016-01-01","require
d_date":"2016-01-03","shipped_date":"2016-01-03","staff_id":"2","custo
mer_id":"259","order_id":"1"},{},{"quantity":"1","item_id":"2","produc
t_id":"16","discount":".20","list_price":"599.99","order_id":"172"}]
├──────────────────────────────────────────────────────────────────────────────┤
[{"store_id":"1","order_status":"4","order_date":"2016-01-01","require
d_date":"2016-01-03","shipped_date":"2016-01-03","staff_id":"2","custo
mer_id":"259","order_id":"1"},{},{"quantity":"1","item_id":"2","produc
t_id":"44","discount":".07","list_price":"539.99","order_id":"936"}]
├──────────────────────────────────────────────────────────────────────────────┤
[{"store_id":"1","order_status":"4","order_date":"2016-01-01","require
d_date":"2016-01-03","shipped_date":"2016-01-03","staff_id":"2","custo
mer_id":"259","order_id":"1"},{},{"quantity":"1","item_id":"2","produc
t_id":"2","discount":".07","list_price":"749.99","order_id":"97"}]
├──────────────────────────────────────────────────────────────────────────────┤
[{"store_id":"1","order_status":"4","order_date":"2016-01-01","require
d_date":"2016-01-03","shipped_date":"2016-01-03","staff_id":"2","custo
mer_id":"259","order_id":"1"},{},{"quantity":"1","item_id":"2","produc
t_id":"54","discount":".07","list_price":"3199.99","order_id":"800"}]
└──────────────────────────────────────────────────────────────────────────────┘
```

Order_Items and Product (OrderedItems_Includes)

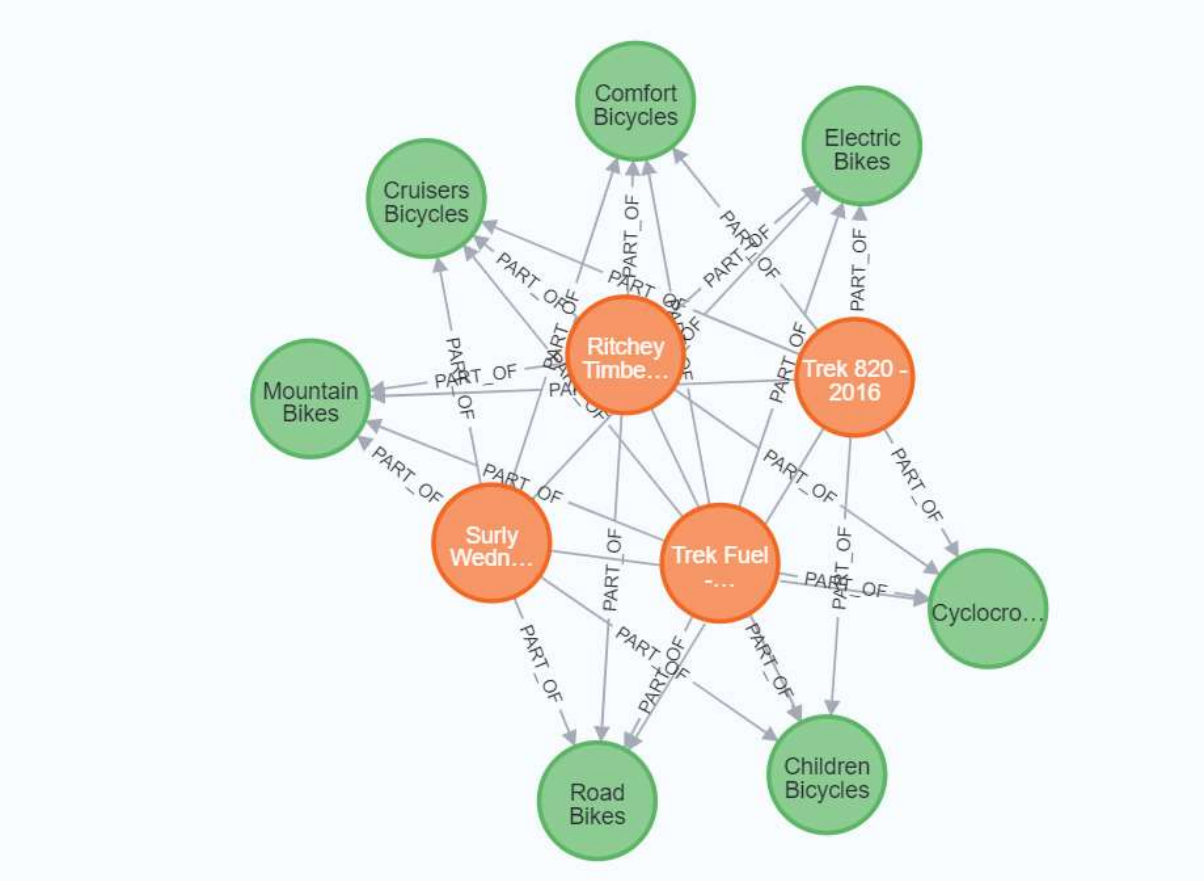
```
MATCH(d:Order_Items),(p:Product) WHERE
d:product_id=p:product_id CREATE (d) - [
:OrderedItems_Include]→(p)
```

Created 1515762 relationships, completed after 3455 ms.





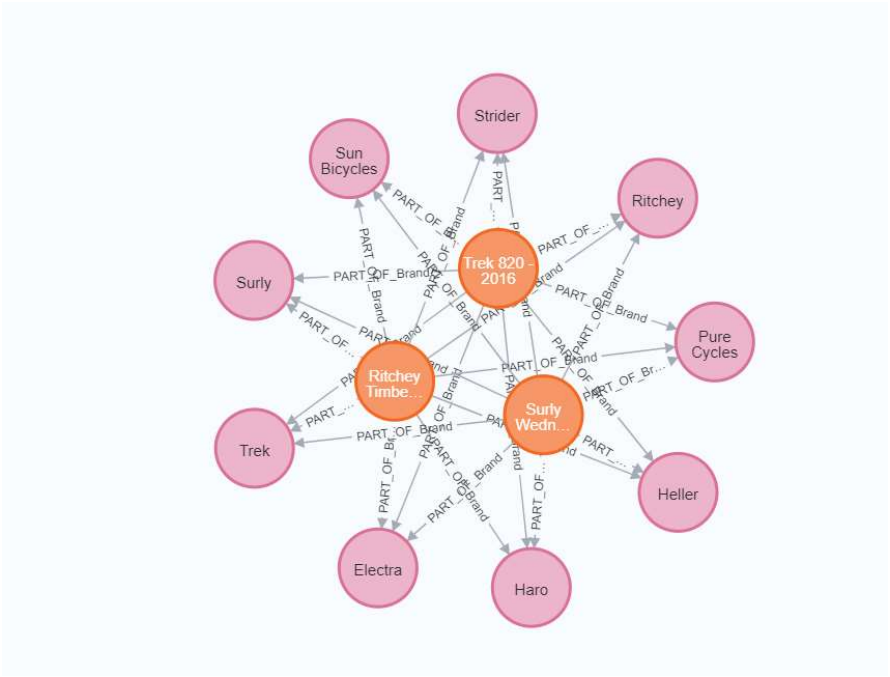
Product and Category (PART_OF)



Product and Brand (PART_OF_Brand)

```
MATCH(p:Product),(b:Brand) WHERE p: brand_id=b:brand_id
CREATE (p)-[ :PART_OF_Brand]→(b)
```

Created 2889 relationships, completed after 8 ms.



Overview

Node labels

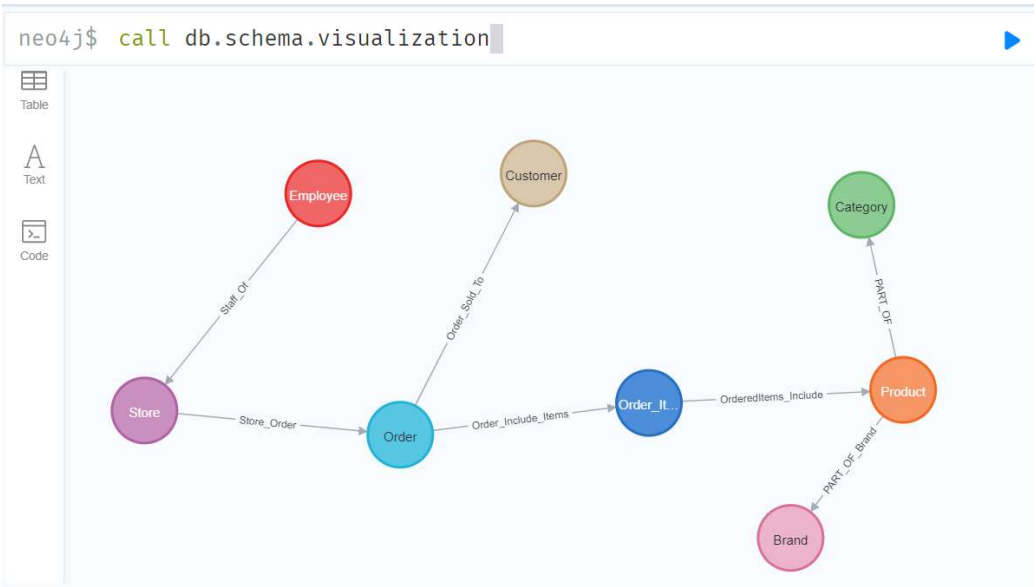
* (12) Product (3) Brand (9)

Relationship Types

* (27) PART_OF_Brand (27)

Displaying 12 nodes, 25 relationships.

Database Schema Visualization



Node 1	Node 2	Relationship
Employee	Store	Staff_Of
Store	Order	Store_Order
Order	Customer	Order_Sold_To
Order	Order_Items	Order_Include_Items
Order_Items	Product	OrderedItems_Include
Product	Category	PART_OF
Product	Brand	PART_OF_Brand

Queries: Cypher Versus SQL

Query1

To know the category of a product

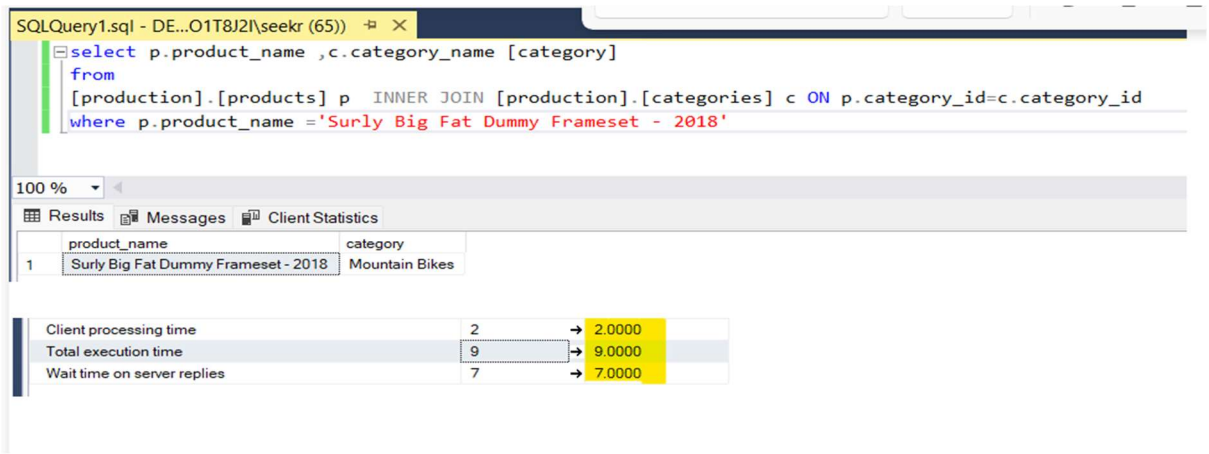
```
neo4j$
```

```
neo4j$ MATCH(p:Product{product_name:'Surly Big Fat Dummy Frameset - 2018'})⟶(c:Category) WHERE
       EXISTS{MATCH(p)-[PART_OF]→(c) WHERE p.category_id=c.category_id} RETURN p.product_name AS
       ProductName, c.category_name AS CATEGORIES
```

Table | Text | Code

	ProductName	CATEGORIES
1	"Surly Big Fat Dummy Frameset - 2018"	"Mountain Bikes"

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



Query 2

To retrieve the information about the staff working at one of the stores

```
neo4j$ MATCH (e:Employee)-[:Staff_Of]→(s:Store) WHERE s.store_name='Baldwin Bikes'RETURN
       s.store_name AS Store, s.email AS StoreEmail, e.email AS EmployeeEmail, e.first_name AS
       FirstName, e.last_name AS LastName
```

Table
Text
Code

"Store"	"StoreEmail"	"EmployeeEmail"	"FirstName"	"LastName"
"Baldwin Bikes"	"baldwin@bikes.shop"	"bernardine.houston@bikes.shop"	"Bernardine"	"Houston"
"Baldwin Bikes"	"baldwin@bikes.shop"	"layla.terrell@bikes.shop"	"Layla"	"Terrell"
"Baldwin Bikes"	"baldwin@bikes.shop"	"kali.vargas@bikes.shop"	"Kali"	"Vargas"
"Baldwin Bikes"	"baldwin@bikes.shop"	"venita.daniel@bikes.shop"	"Venita"	"Daniel"
"Baldwin Bikes"	"baldwin@bikes.shop"	"marcelene.boyer@bikes.shop"	"Marcelene"	"Boyer"
"Baldwin Bikes"	"baldwin@bikes.shop"	"jannette.david@bikes.shop"	"Jannette"	"David"
"Baldwin Bikes"	"baldwin@bikes.shop"	"virgie.wiggins@bikes.shop"	"Virgie"	"Wiggins"
"Baldwin Bikes"	"baldwin@bikes.shop"	"genna.serrano@bikes.shop"	"Genna"	"Serrano"

```
neo4j$ MATCH (e:Employee)-[:Staff_Of]→(s:Store) WHERE s.store_name='Baldwin Bikes'RETURN
       s.store_name AS Store, s.email AS StoreEmail, e.email AS EmployeeEmail, e.first_name AS
       FirstName, e.last_name AS LastName
```

Table | Text | Code

Server version	Neo4j/4.4.5
Server address	localhost:7687
Query	MATCH (e:Employee)-[:Staff_Of]->(s:Store) WHERE s.store_name='Baldwin Bikes'RETURN s.store_name AS Store, s.email AS StoreEmail, e.email AS EmployeeEmail, e.first_name AS FirstName, e.last_name AS LastName
Summary ▸	{, "query": {, "text": "MATCH (e:Employee)-[:Staff_Of]->(s:Store) WHERE s.store_name='Baldwin Bikes'RETURN s.store_name AS Store, s.email AS StoreEmail, e.email AS EmployeeEmail, e.first_name AS FirstName, e.last_name AS LastName", ...
Response ▸	[, {, "keys": [...

Started streaming 10 records in less than 1 ms and completed in less than 1 ms.

SQLQuery9.sql - DE...O1T8J2I\seekr (56))*

```
SELECT
st.store_name,st.email [store email],st.email [employee email],sf.first_name,sf.last_name
FROM

[sales].[staffs] sf JOIN [sales].[stores] st ON sf.store_id=st.store_id

WHERE st.store_name='Baldwin Bikes'
```

100 %

Results | Messages | Client Statistics

	store_name	store email	employee email	first_name	last_name
1	Baldwin Bikes	baldwin@bikes.shop	baldwin@bikes.shop	Jannette	David
2	Baldwin Bikes	baldwin@bikes.shop	baldwin@bikes.shop	Marcelene	Boyer
3	Baldwin Bikes	baldwin@bikes.shop	baldwin@bikes.shop	Venita	Daniel

Time Statistics					
Client processing time	17	↑	0	→	8.5000
Total execution time	23	↑	0	→	11.5000
Wait time on server replies	6	↑	0	→	3.0000

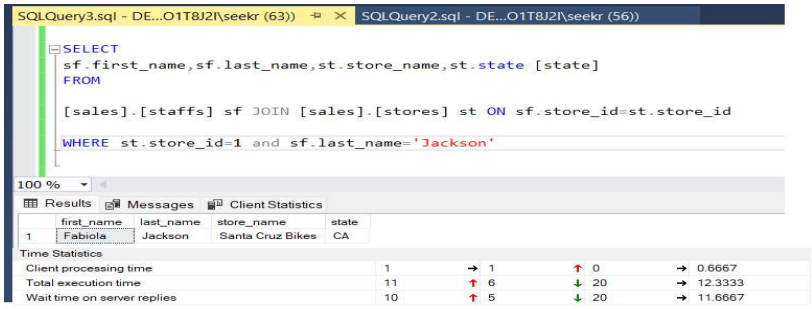
Query 3

To retrieve the information about employee with last_name working at a store

```
MATCH (e:Employee{store_id:'1'})⟶(s:Store{store_id:'1'})
WHERE EXISTS {
  MATCH (e)-[:Staff_Of]→(s)
  WHERE e.last_name = 'Jackson'
}
RETURN e.last_name AS LastName, e.first_name AS FirstName,s.store_name AS StoreName, s.state AS
State
```

	LastName	FirstName	StoreName	State
1	"Jackson"	"Fabiola"	"Santa Cruz Bikes"	"CA"

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



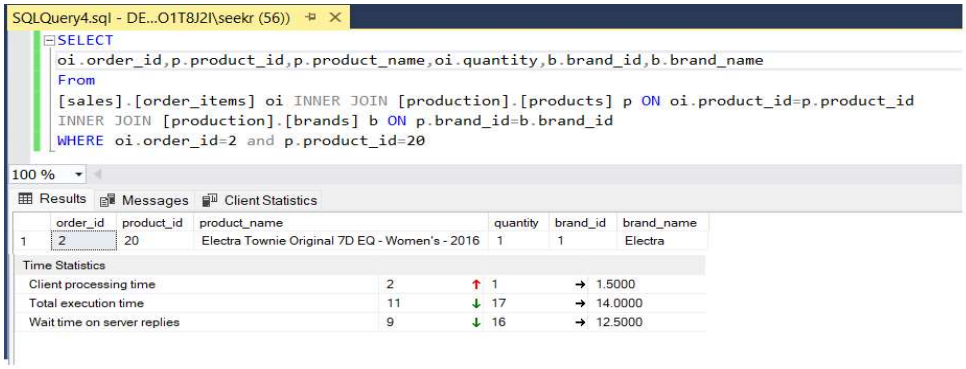
Query 4

To retrieve the information about an order_id with product_id

```
MATCH (o:Order{order_id:'2'})⟶(i:Order_Items{product_id:'20',order_id:'2'})⟶
(p:Product{product_id:'20'})⟶(b:Brand)
WHERE EXISTS {
  MATCH (o)-[:Order_Include_Items]→(i)-[:OrderedItems_Include]→(p)-[:PART_OF_Brand]→(b)
WHERE p.brand_id =b.brand_id
}
RETURN o.order_id AS OrderNum, p.product_id AS ProductNum,p.product_name AS
ProductName,i.quantity AS QTY,b.brand_id AS BRAND,b.brand_name AS BrandName
```

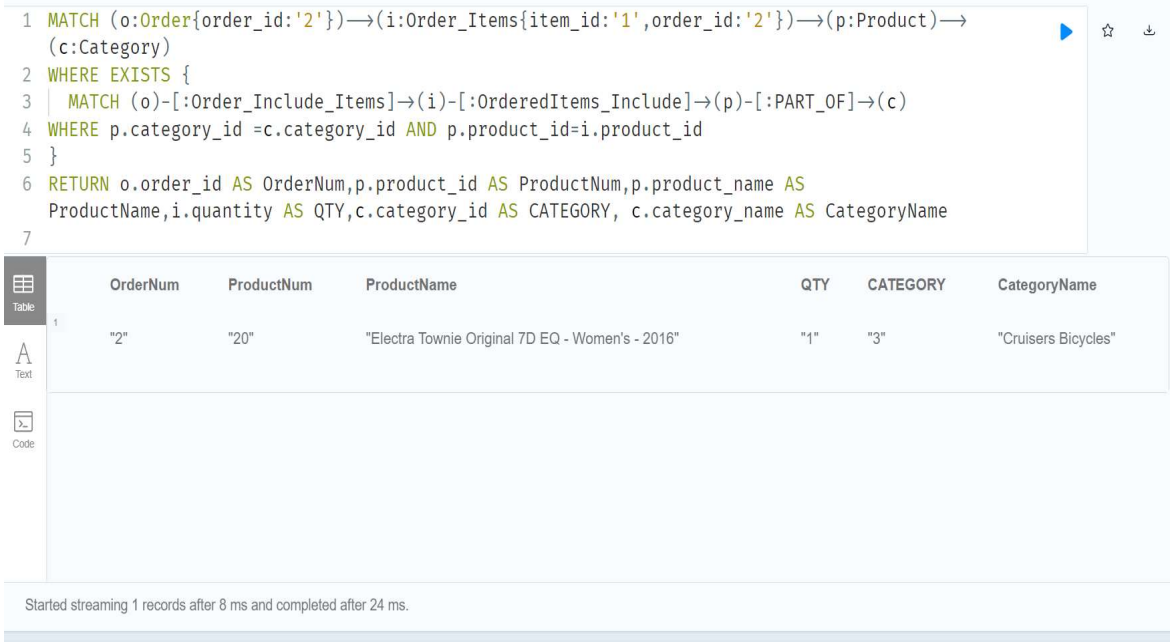
	OrderNum	ProductNum	ProductName	QTY	BRAND	BrandName
1	"2"	"20"	"Electra Townie Original 7D EQ - Women's - 2016"	"1"	"1"	"Electra"

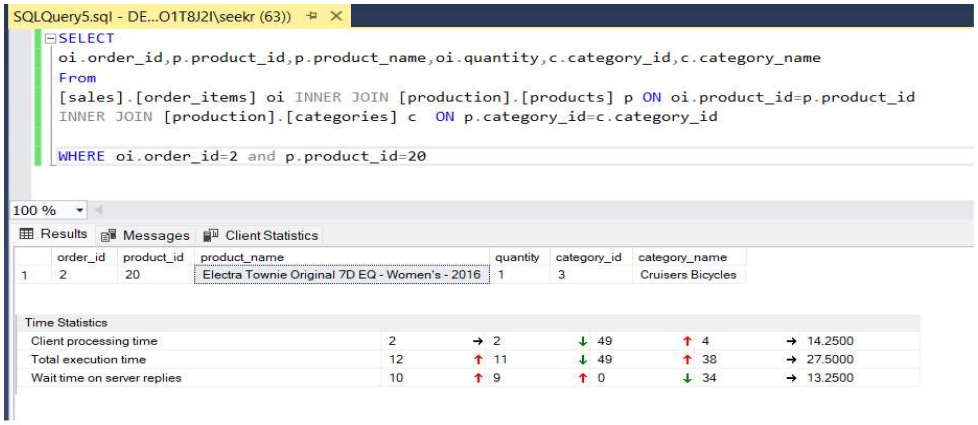
Started streaming 1 records in less than 1 ms and completed after 8 ms.



Query 5

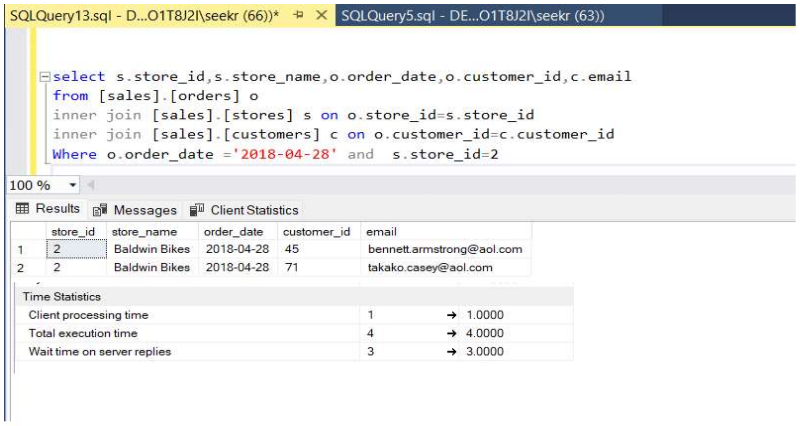
To retrieve the information including category, about order_id and product_id





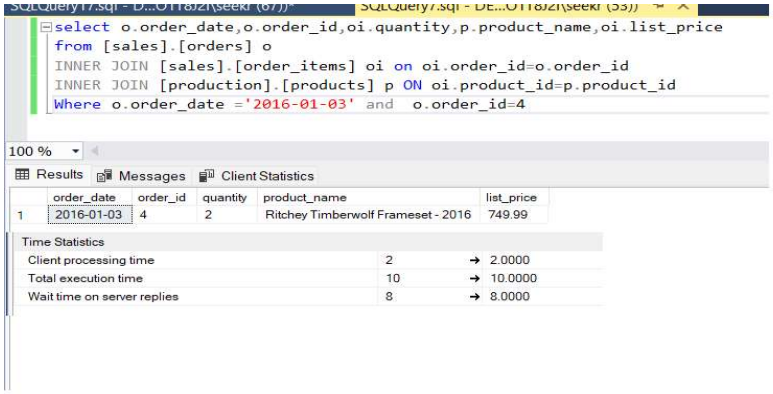
Query 6

To retrieve the information about a store_id e.g. '2', order_date e.g. '2018-04-28' about store name, order date, and customer's email.



Query 7

To retrieve the information about order_id e.g. '4' when the order date is e.g. '2016-01-03' about ordered quantity, product name and list price.



Conclusion

My Learning

MS SQL vs Cypher

Queries	Execution Time (ms)	
	MS SQL Server	Cypher
Query 1	9	After 1
Query 2	11.5	After 1
Query 3	12.3333	After 1
Query 4	14	After 8
Query 5	27.5	After 24
Query 6	4	After 24
Query 7	10	After 16

Table 1

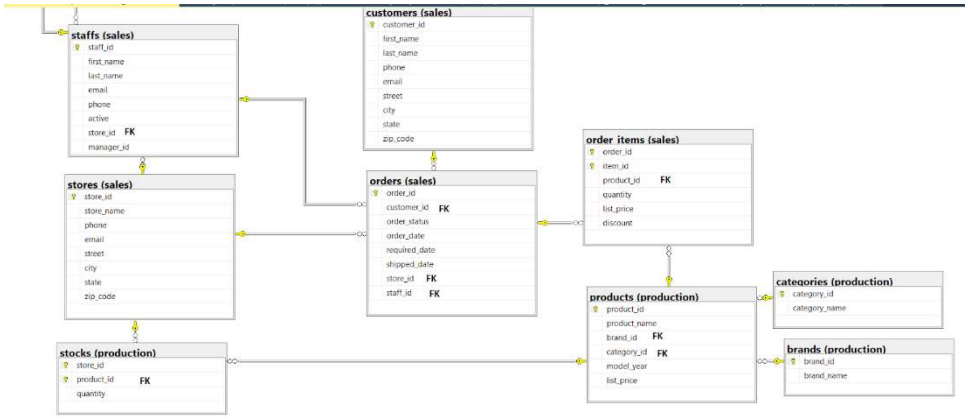
SQL queries for deep hierarchies are complex and slow while Neo4J queries are concise and fast. Table 1 shows that cypher is faster as compared to SQL.

Relational databases do not store the relationship between data. Graph Databases store relationship between data as entities.

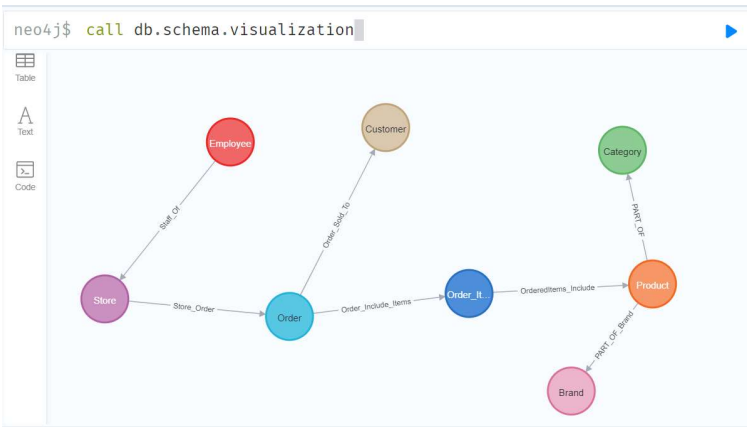
Relational Database	Graph Databases Neo4J
Tables	Graphs
Rows	Nodes
Constraints	Relationships
Columns	Properties
Use of joins	No use of joins
Use of Schema	Schema free
Structured Data	All
Secure	No security
Less or no scalable	High scalability
ACID compliant	ACID compliant

(Hunger, Boyd and Lyon, 2016)

ERD of the operational database



As shown in ERD, it is difficult to match in-memory data structures to the relational tables. While in graph database, as in figure below, it is easy to match the data structures to the relational tables.



References

‘Comparing SQL with Cypher’*Neo4J Developer*, Available at: Comparing SQL with Cypher - Developer Guides (neo4j.com)

Hunger, M., Boyd, R. and Lyon, W. (2016)’RDMS & Graphs: SQL vs. Cypher Query Languages’, *neo4j,* Available at: RDBMS & Graphs: SQL vs. Cypher Query Languages (neo4j.com)

Lydon, B. (2022), ‘On Demand Lesson 5 -NoSQL Databases: The Emergence of NoSQL’, *B9DA111_2122_TMD3: Data Storage Solutions for Data Analytics*. Dublin Business School. Available at:https://study.dbs.ie/2122/msc-data/B9DA111/Unit5/index.html#/lessons/KConnDRWyCDPTnj-mSGDRz2THNlFP4ab