Practical-9

Aim: Import files from Neo4j and complete relational graph from it

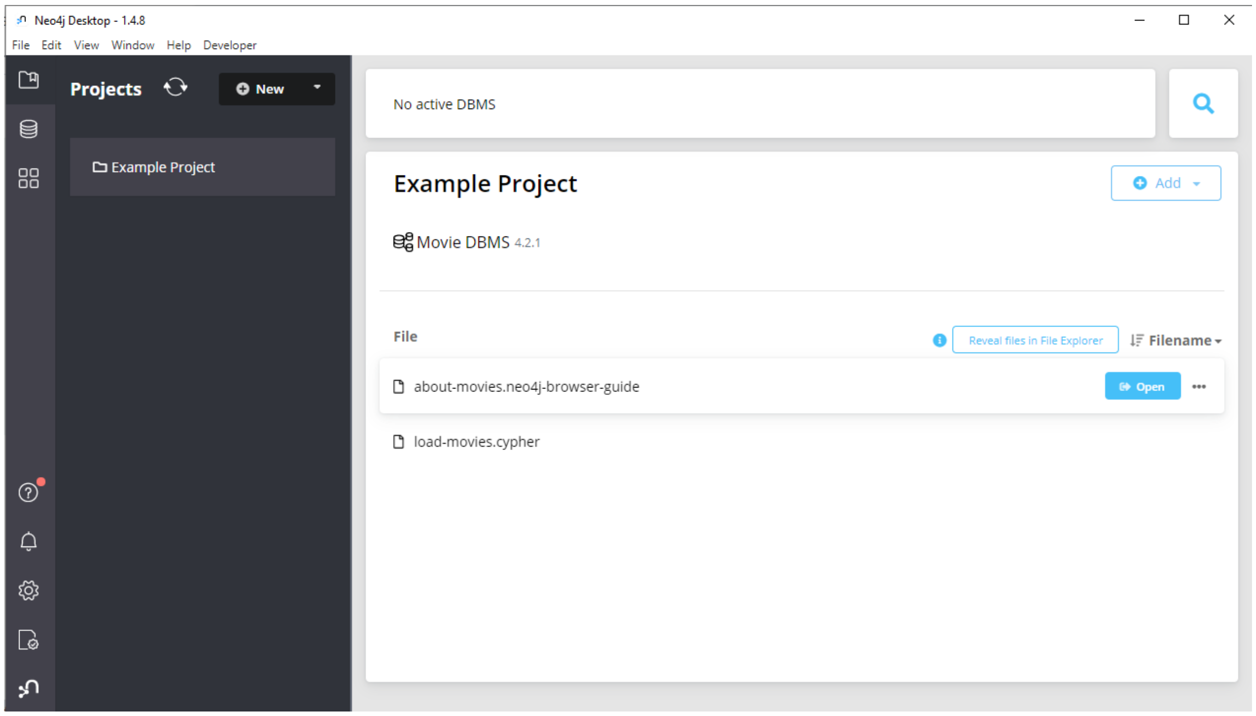
Theory:

Neo4j:
Neo4j facilitates personal data storage and management: it allows you to track where private information is stored and which systems, applications, and users access it. The graph data model helps visualize personal data and allows for data analysis and pattern detection.

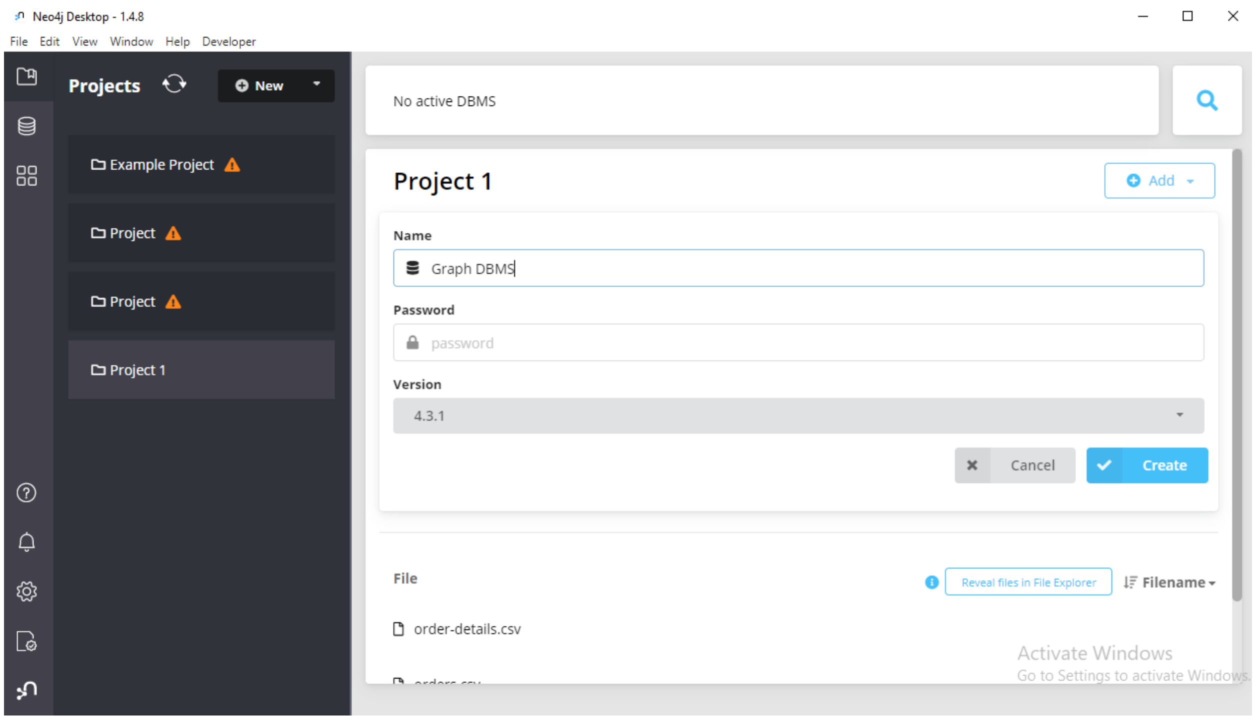
Implementation:

For Installation go to the link: https://neo4j.com/download/ and download Neo4j Desktop.Open the Neo4j Desktop installer. On your computer, locate the file you just downloaded and double click to begin the install.Install Neo4j Desktop. Follow the on-screen steps to complete the installation.

First step after launching the Neo4j is to set the path for storing the application data on the computer.Then Enter the software key or the credential for getting started with the software:



Click on New Button in the left panel in projects.Thus a New project will be created and there click on the Add button and click on Local DBMS and enter the password:



Project 1

Name: Graph DBMS

Password: password

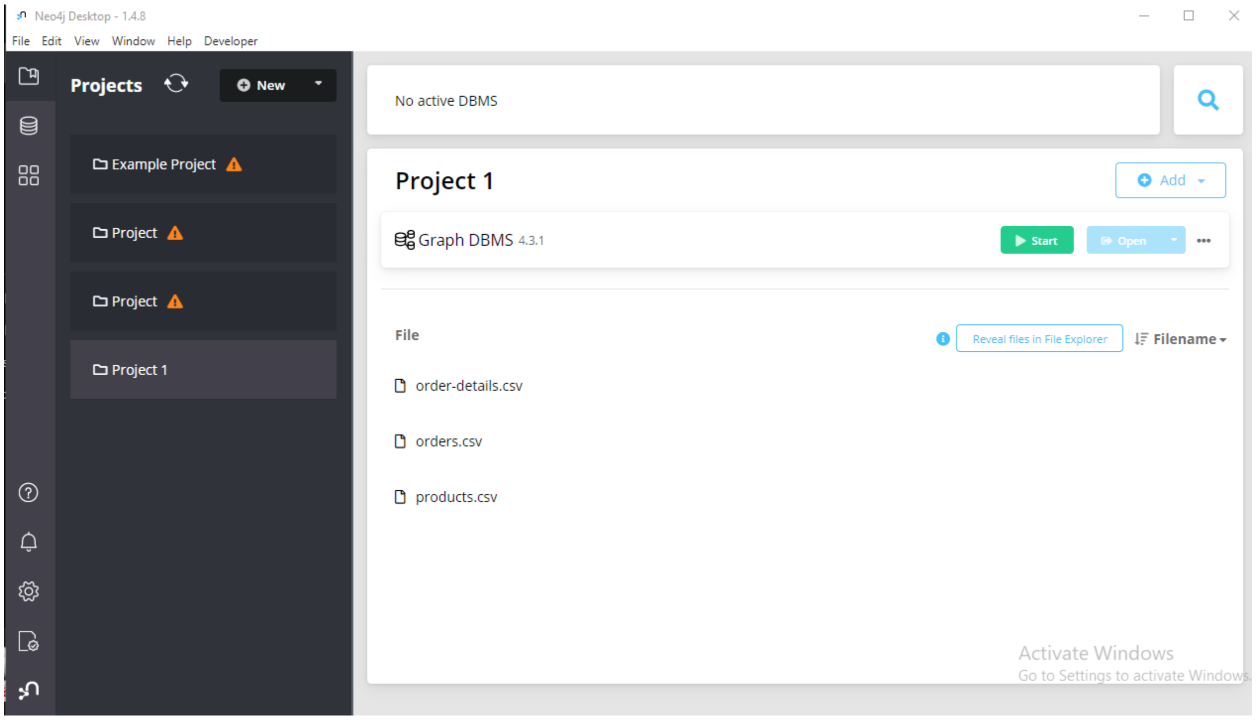
Version: 4.3.1

File

order-details.csv

orders.csv

And then click on Create.



Project 1

Graph DBMS 4.3.1

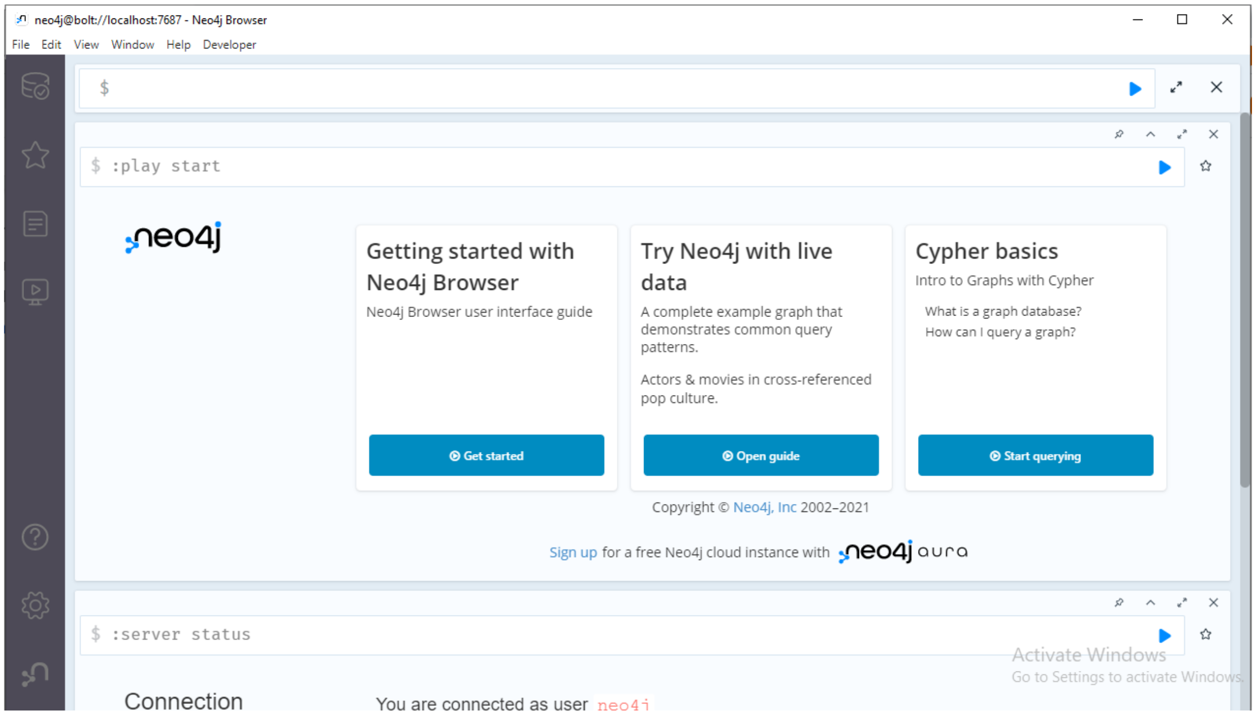
File

order-details.csv

orders.csv

products.csv

Then click on Start and then click on Open. This will launch the Neo4j browser.


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

File Edit View Window Help Developer

```
$ :play start
```

Getting started with Neo4j Browser
Neo4j Browser user interface guide

Try Neo4j with live data
A complete example graph that demonstrates common query patterns.
Actors & movies in cross-referenced pop culture.

Cypher basics
Intro to Graphs with Cypher
What is a graph database?
How can I query a graph?

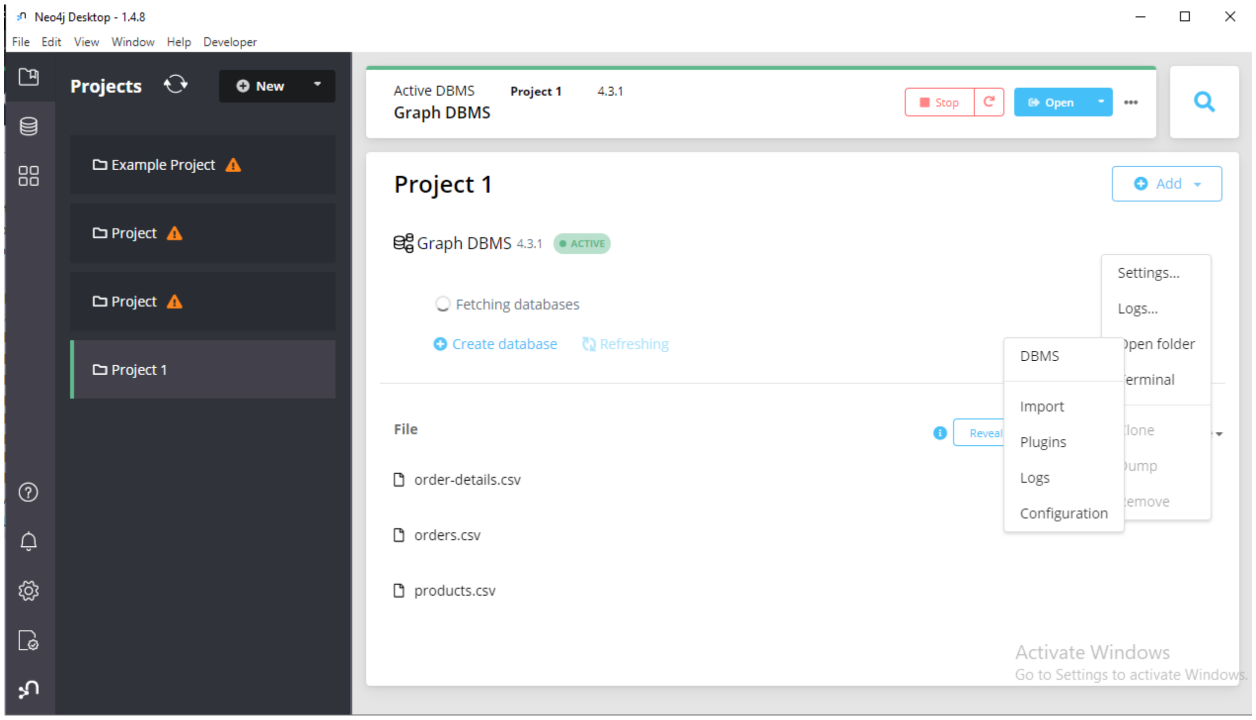
Copyright © Neo4j, Inc 2002–2021

Sign up for a free Neo4j cloud instance with neo4j aura

```
$ :server status
```

Connection   You are connected as user neo4j

Click Import in the project and it will open the import folder in the file Explorer:


Neo4j Desktop - 1.4.8

File Edit View Window Help Developer

Active DBMS Project 1 4.3.1
Graph DBMS

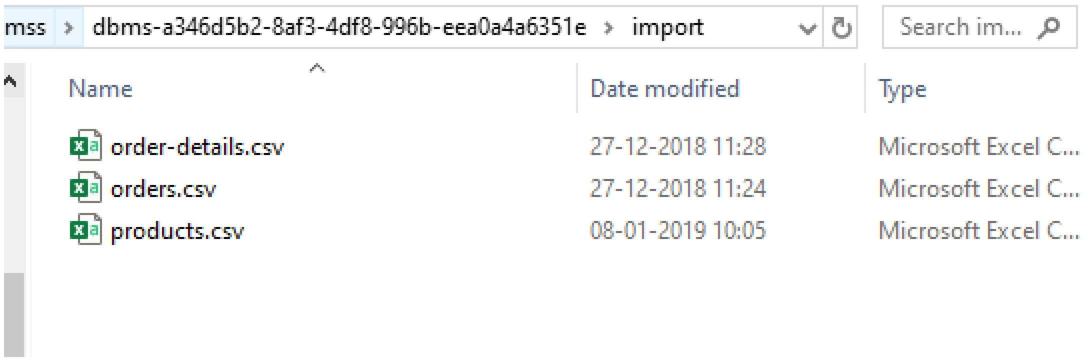
Project 1

Graph DBMS 4.3.1 ACTIVE

Fetching databases

File
order-details.csv
orders.csv
products.csv

And at the location of import copy the csv files:

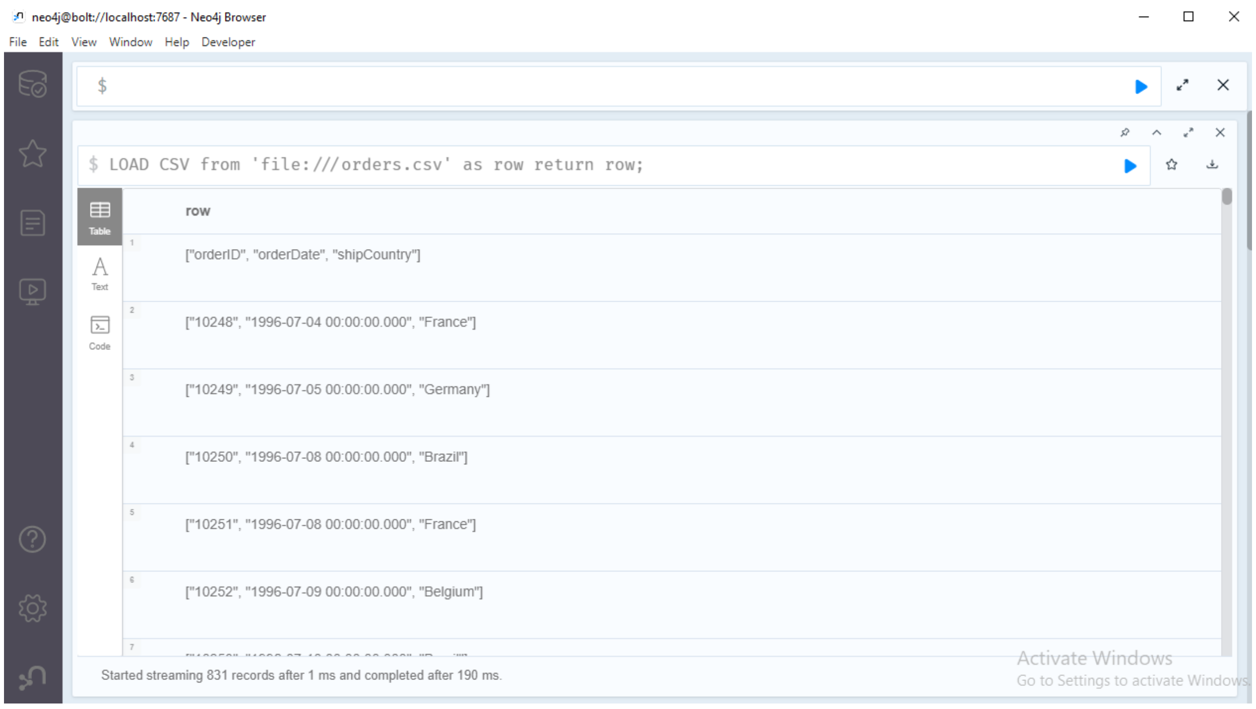


mss > dbms-a346d5b2-8af3-4df8-996b-eea0a4a6351e > import

Name	Date modified	Type
order-details.csv	27-12-2018 11:28	Microsoft Excel C...
orders.csv	27-12-2018 11:24	Microsoft Excel C...
products.csv	08-01-2019 10:05	Microsoft Excel C...

To read data from csv file:

Command:
LOAD CSV from 'file:///orders.csv' as row return row;



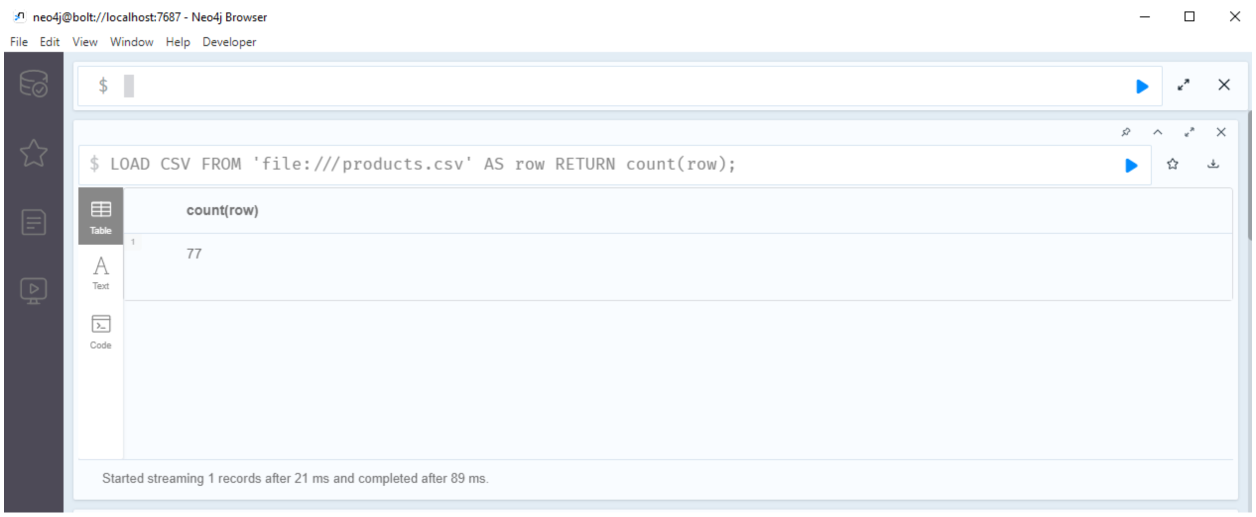
neo4j@bolt://localhost:7687 - Neo4j Browser
File Edit View Window Help Developer
Table
Text
Code

```
$ LOAD CSV from 'file:///orders.csv' as row return row;
```

```
row
["orderID", "orderDate", "shipCountry"]
["10248", "1996-07-04 00:00:00.000", "France"]
["10249", "1996-07-05 00:00:00.000", "Germany"]
["10250", "1996-07-08 00:00:00.000", "Brazil"]
["10251", "1996-07-08 00:00:00.000", "France"]
["10252", "1996-07-09 00:00:00.000", "Belgium"]
Started streaming 831 records after 1 ms and completed after 190 ms.
```

To count row from data :

Command:
LOAD CSV FROM 'file:///products.csv' AS row RETURN count(row);

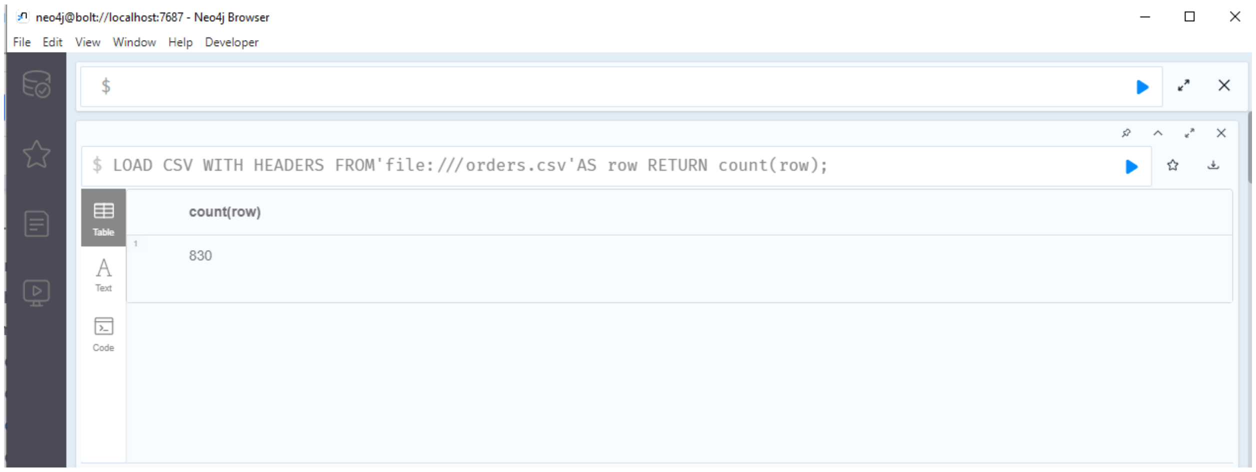

neo4j@bolt://localhost:7687 - Neo4j Browser
File Edit View Window Help Developer
Table
Text
Code

```
$ LOAD CSV FROM 'file:///products.csv' AS row RETURN count(row);
```

```
count(row)
77
```

```
Started streaming 1 records after 21 ms and completed after 89 ms.
```

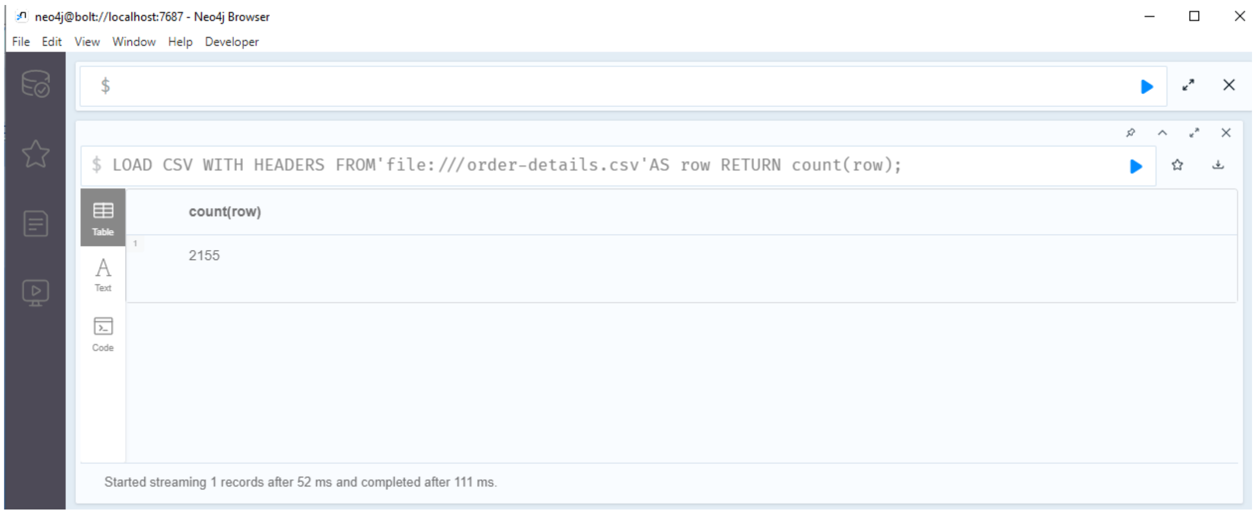
Command:
LOAD CSV WITH HEADERS FROM'file:///orders.csv'AS row RETURN count(row);


neo4j@bolt://localhost:7687 - Neo4j Browser
File Edit View Window Help Developer
Table
Text
Code

```
$ LOAD CSV WITH HEADERS FROM'file:///orders.csv'AS row RETURN count(row);
```

```
count(row)
830
```

Command:
LOAD CSV WITH HEADERS FROM'file:///order-details.csv'AS row RETURN count(row);


neo4j@bolt://localhost:7687 - Neo4j Browser
File Edit View Window Help Developer
Table
Text
Code

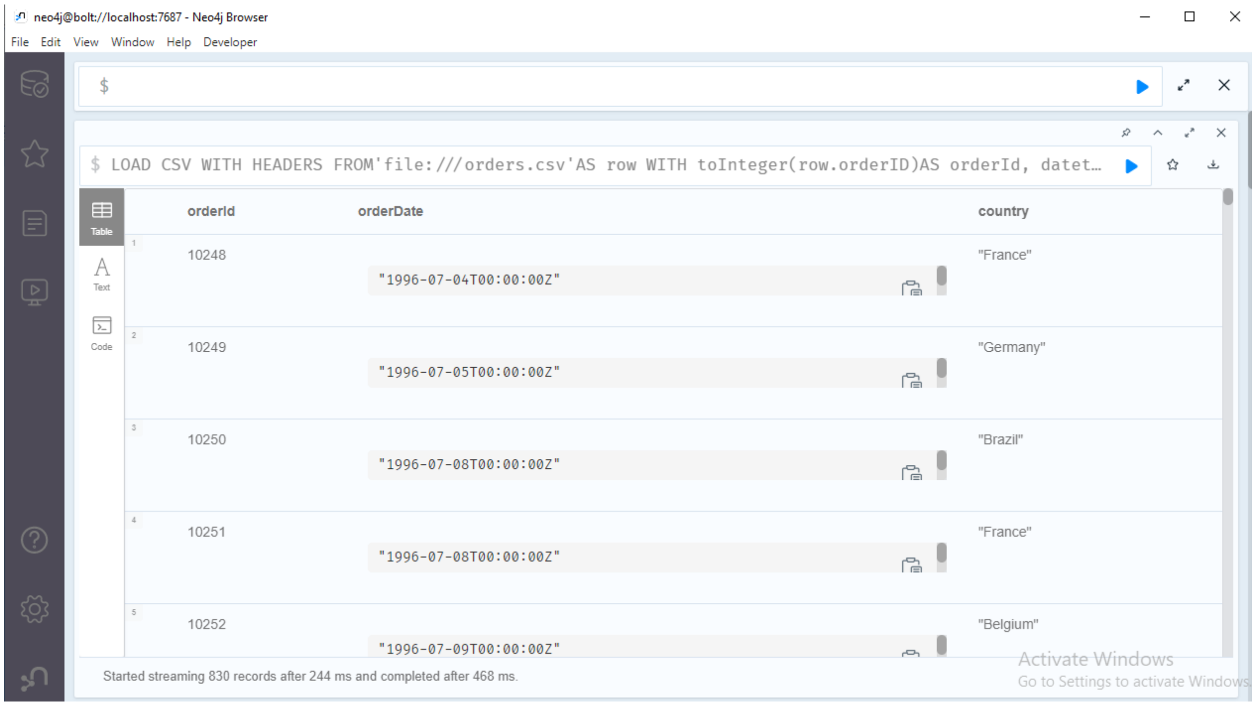
```
$ LOAD CSV WITH HEADERS FROM'file:///order-details.csv'AS row RETURN count(row);
```

```
count(row)
2155
```

```
Started streaming 1 records after 52 ms and completed after 111 ms.
```

Here Data in string format by default
Convert in appropriate format
toInteger(): converts a value to an integer.
toFloat(): converts a value to a float (in this case, for monetary amounts).
datetime(): converts a value to a datetime.

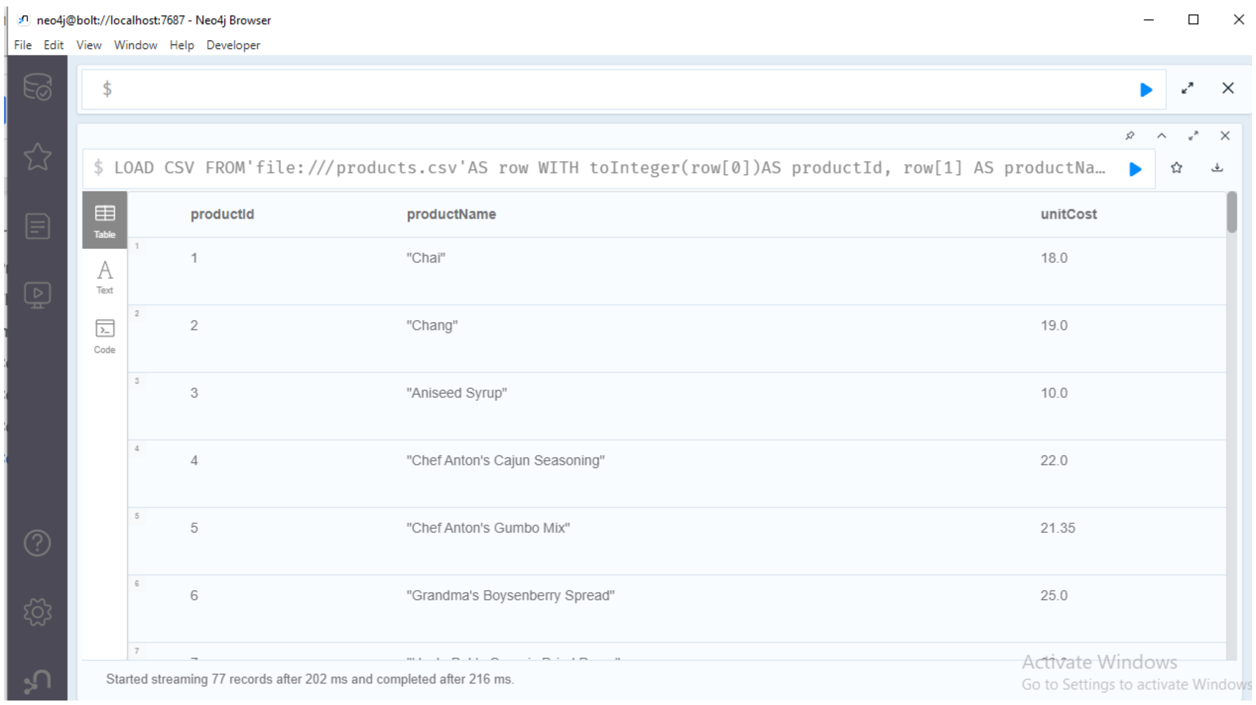
Command:
LOAD CSV WITH HEADERS FROM'file:///orders.csv'AS row
WITH toInteger(row.orderID)AS orderId, datetime(replace(row.orderDate,' ','T')) AS orderDate, row.shipCountry AS country
RETURN orderId, orderDate, country


neo4j@bolt://localhost:7687 - Neo4j Browser
File Edit View Window Help Developer
Table
Text
Code

```
$ LOAD CSV WITH HEADERS FROM'file:///orders.csv'AS row WITH toInteger(row.orderID)AS orderId, datet…
```

```
orderId   orderDate                  country
10248     "1996-07-04T00:00:00Z"     "France"
10249     "1996-07-05T00:00:00Z"     "Germany"
10250     "1996-07-08T00:00:00Z"     "Brazil"
10251     "1996-07-08T00:00:00Z"     "France"
10252     "1996-07-09T00:00:00Z"     "Belgium"
Started streaming 830 records after 244 ms and completed after 468 ms.
```

Command:
LOAD CSV FROM'file:///products.csv'AS row
WITH toInteger(row[0])AS productId, row[1] AS productName,toFloat(row[2])AS unitCost
RETURN productId, productName, unitCost;



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

```
$ LOAD CSV FROM'file:///products.csv'AS row WITH toInteger(row[0])AS productId, row[1] AS productNa…
```

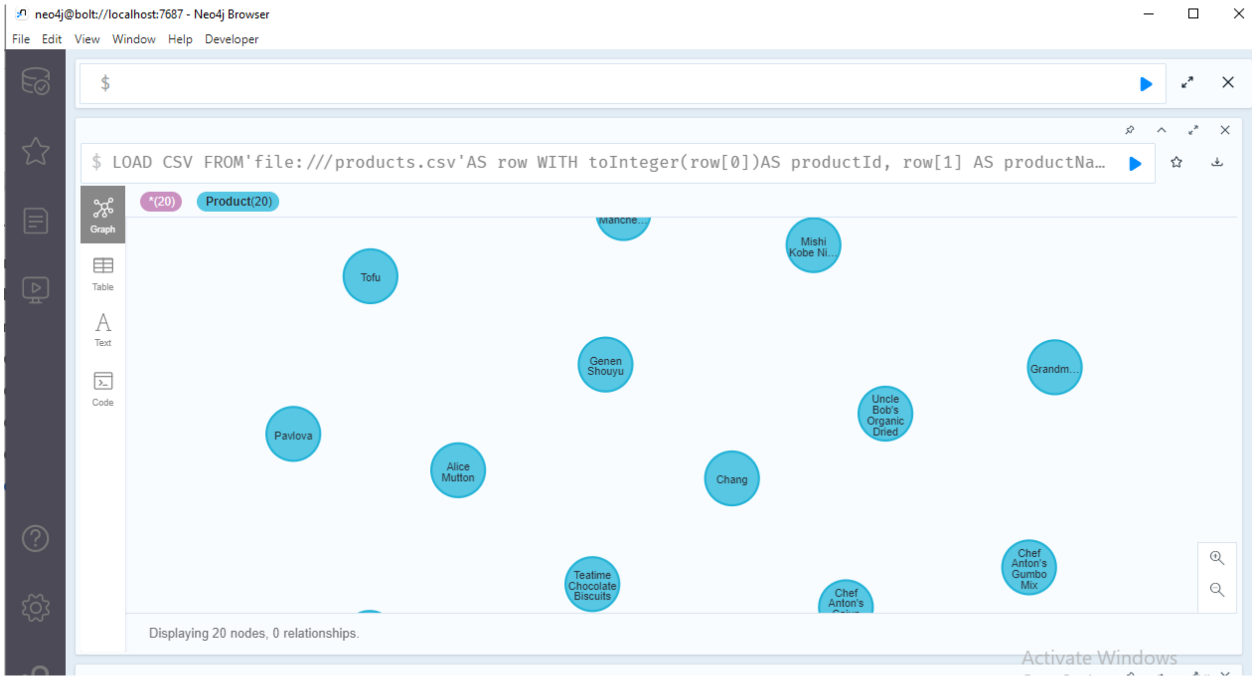
productId	productName	unitCost
1	"Chai"	18.0
2	"Chang"	19.0
3	"Aniseed Syrup"	10.0
4	"Chef Anton's Cajun Seasoning"	22.0
5	"Chef Anton's Gumbo Mix"	21.35
6	"Grandma's Boysenberry Spread"	25.0

Started streaming 77 records after 202 ms and completed after 216 ms.

Now Create a Node:

Command:

LOAD CSV FROM'file:///products.csv'AS row
WITH toInteger(row[0])AS productId, row[1] AS productName,toFloat(row[2])AS unitCost
MERGE (p:Product {productId: productId})
SET p.productName = productName, p.unitCost = unitCost
RETURN p LIMIT 20



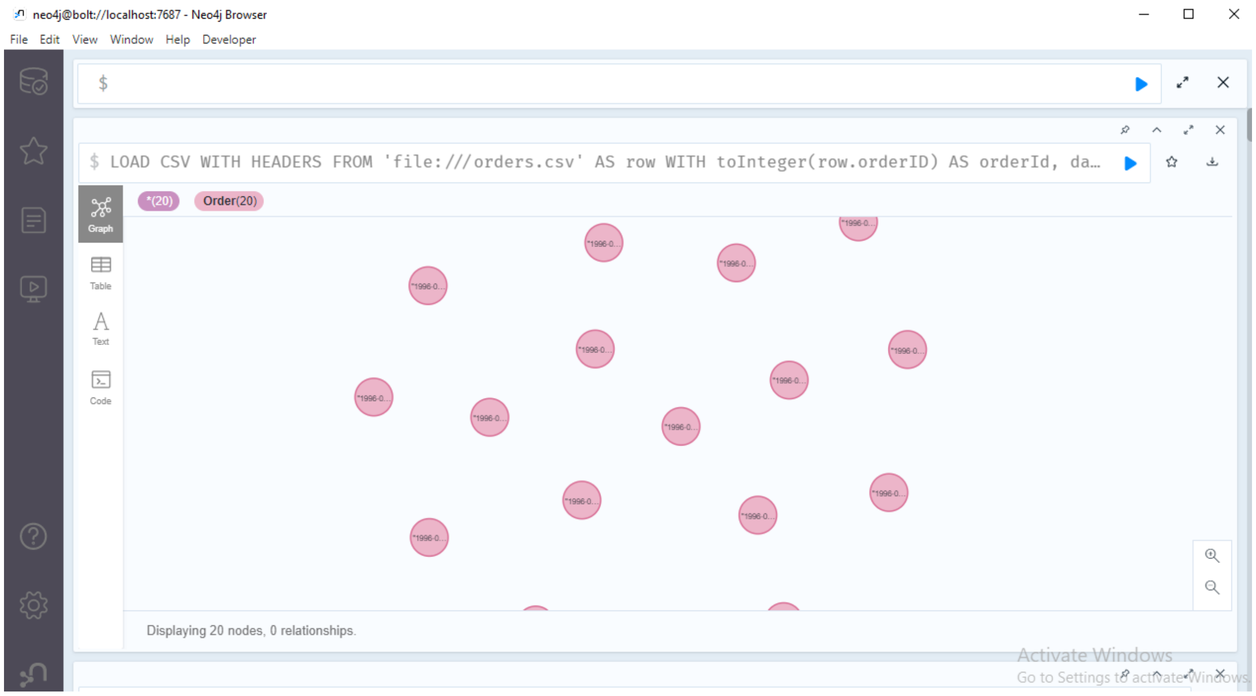
neo4j@bolt://localhost:7687 - Neo4j Browser

```
$ LOAD CSV FROM'file:///products.csv'AS row WITH toInteger(row[0])AS productId, row[1] AS productNa…
```

Displaying 20 nodes, 0 relationships.

Command:

```
LOAD CSV WITH HEADERS FROM 'file:///orders.csv' AS row
WITH toInteger(row.orderID) AS orderId, datetime(replace(row.orderDate,' ','T')) AS orderDate,
row.shipCountry AS country
MERGE (o:Order {orderId: orderId})
 SET o.orderDateTime = orderDate, o.shipCountry = country
RETURN o LIMIT 20
```


neo4j@bolt://localhost:7687 - Neo4j Browser
File Edit View Window Help Developer
Graph
Table
Text
Code
*(20)
Order(20)
"1996-0...

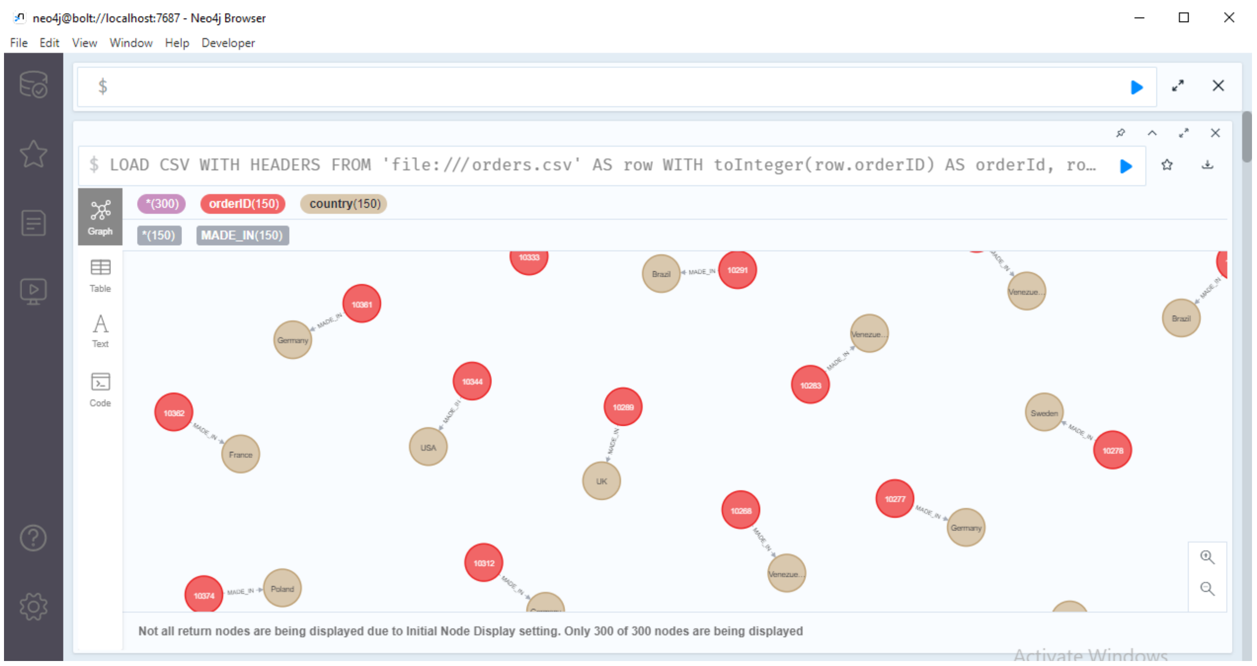
```
$ LOAD CSV WITH HEADERS FROM 'file:///orders.csv' AS row WITH toInteger(row.orderID) AS orderId, da…
```

Displaying 20 nodes, 0 relationships.

Now for Creating Relationship:

Command:

```
LOAD CSV WITH HEADERS FROM 'file:///orders.csv' AS row
WITH toInteger(row.orderID) AS orderId, row.shipCountry AS country
MERGE (o:Order {orderId: orderId})
create (a:orderID {id:orderId})
create (b:country {cname:country})
create (a)-[:MADE_IN]->(b)
RETURN a , b;
```

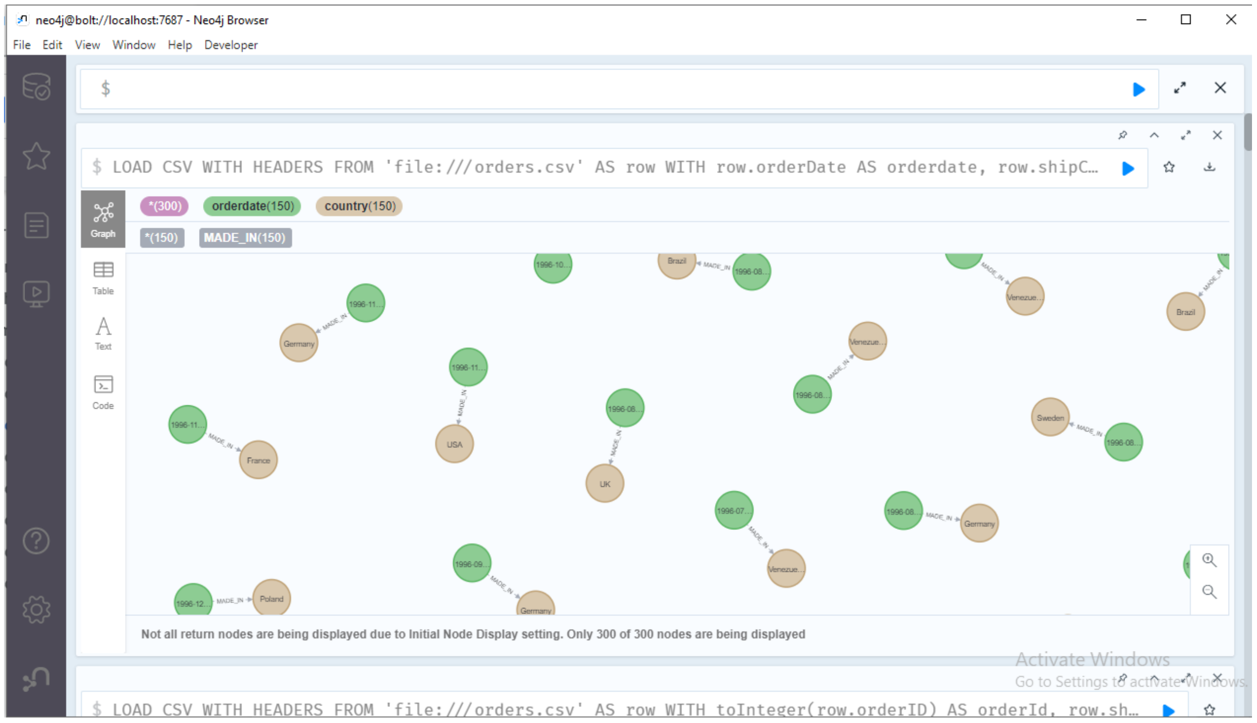


```
neo4j@bolt://localhost:7687 - Neo4j Browser
$ LOAD CSV WITH HEADERS FROM 'file:///orders.csv' AS row WITH toInteger(row.orderID) AS orderId, ro…
```

*(300) orderID(150) country(150)
*(150) MADE_IN(150)

Not all return nodes are being displayed due to Initial Node Display setting. Only 300 of 300 nodes are being displayed

For other Relationship:



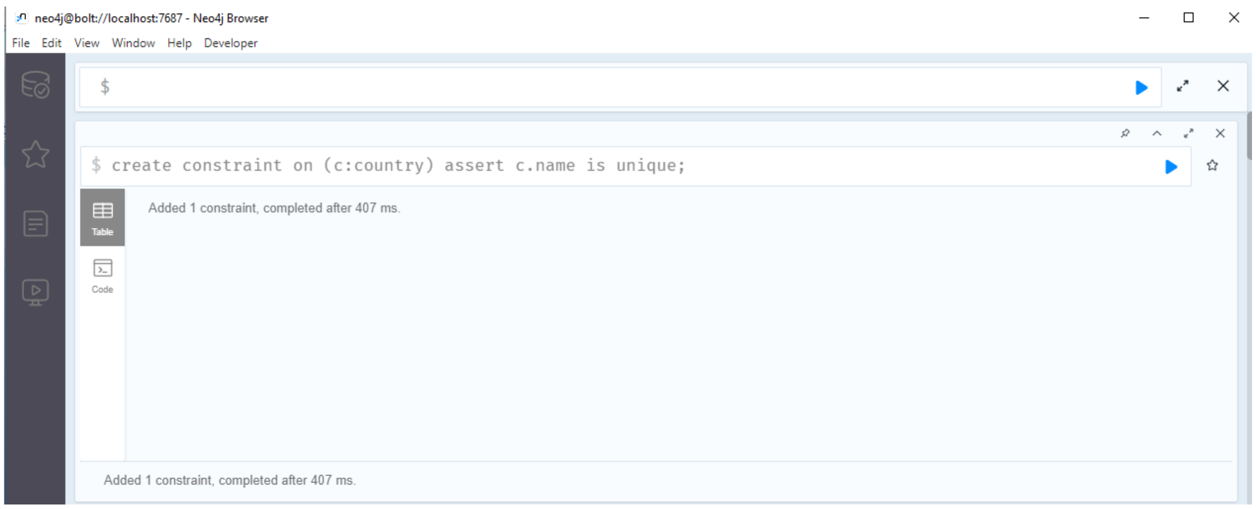
```
neo4j@bolt://localhost:7687 - Neo4j Browser
$ LOAD CSV WITH HEADERS FROM 'file:///orders.csv' AS row WITH row.orderDate AS orderdate, row.shipC…
```

*(300) orderdate(150) country(150)
*(150) MADE_IN(150)

Not all return nodes are being displayed due to Initial Node Display setting. Only 300 of 300 nodes are being displayed

```
$ LOAD CSV WITH HEADERS FROM 'file:///orders.csv' AS row WITH toInteger(row.orderID) AS orderId, row.sh…
```

Now for Creating Unique field:
Command:
create constraint on (c:country) assert c.name is unique;


neo4j@bolt://localhost:7687 - Neo4j Browser
File Edit View Window Help Developer
Table
Code

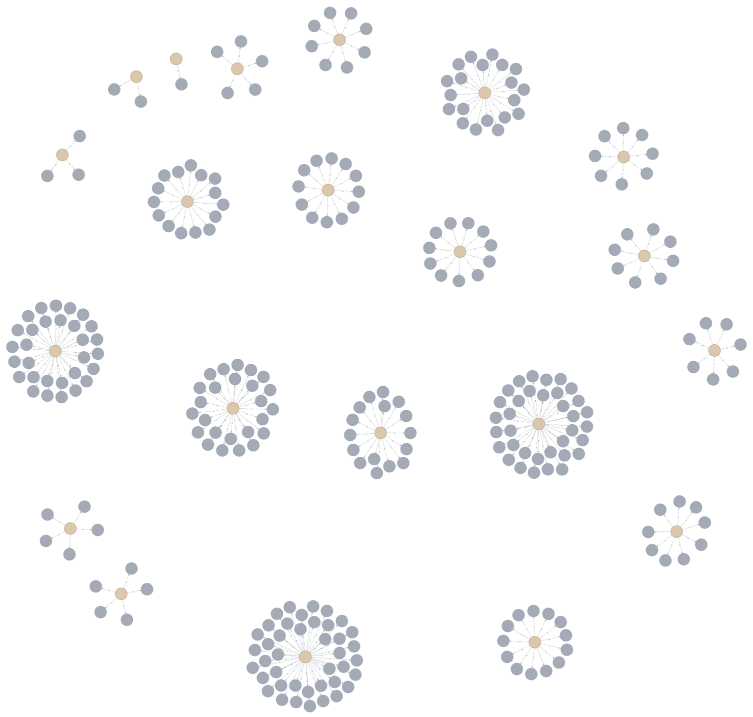
```
$ create constraint on (c:country) assert c.name is unique;
```

```
Added 1 constraint, completed after 407 ms.
```

```
Added 1 constraint, completed after 407 ms.
```

Command:

```
LOAD CSV WITH HEADERS FROM 'file:///orders.csv' AS row
//TH toInteger(row.orderID) AS orderId, row.shipCountry AS country
create (orderId:orderId {id:toInteger(row.orderID)})
MERGE (country: country{name:row.shipCountry})
create (country)-[r:try]->(orderID)
return country ,orderID
//create (a:orderID {id:orderId})
```



Conclusion : Learned , understood and implemented on the Neo4j and imported the files and thus created the nodes & relationships in it.