Course code : **CSE3009**
Course title : **No SQL Data Bases**
Module : **6**
Topic : **3**

# Neo4j Query Execution

# Objectives

This session will give the knowledge about

- Neo4j Building Blocks
- Cypher Query Language (CQL) Introduction

# Neo4j - Building Blocks

Neo4j Graph Database has the following building blocks −

- Nodes
- Properties
- Relationships
- Labels
- Data Browser

# Node

**Node** is a fundamental unit of a Graph. It contains properties with key-value pairs as shown in the following image.

Here, Node Name = "Employee" and it contains a set of properties as key-value pairs.

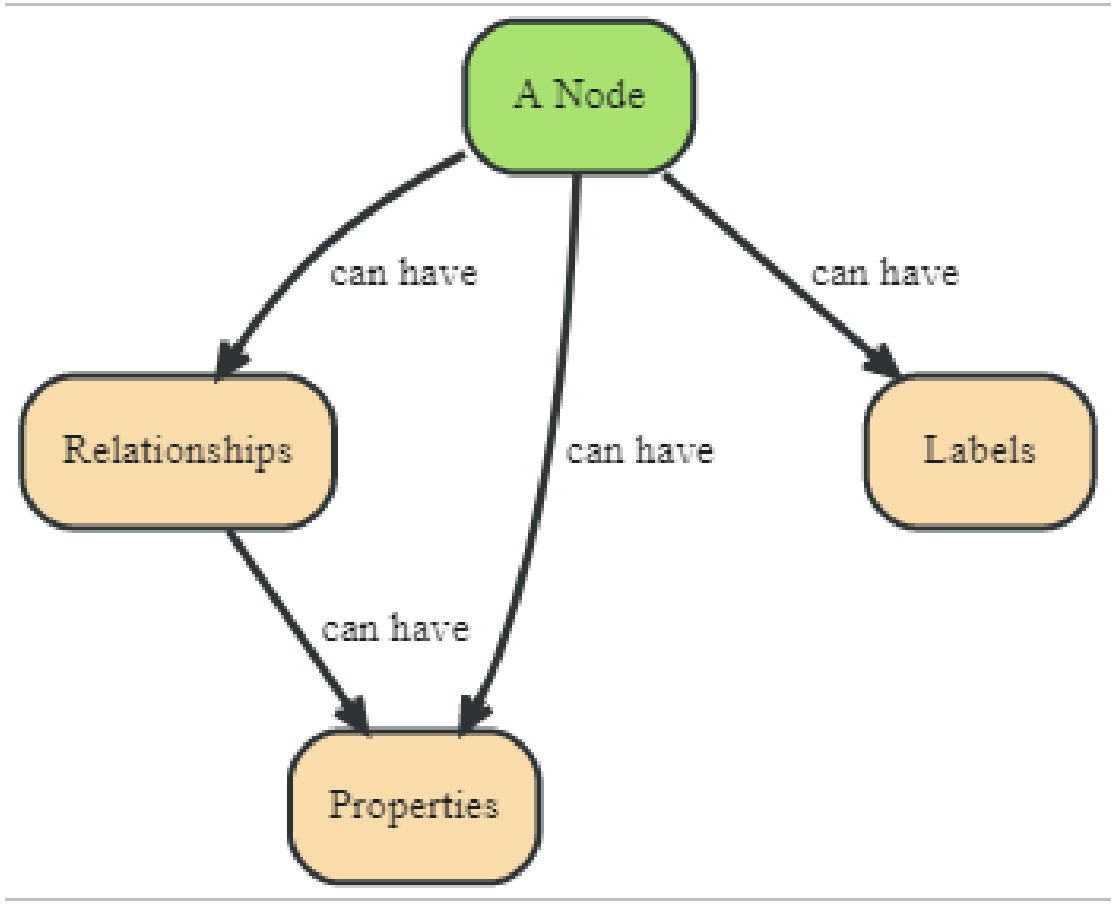

**Properties**

Property is a key-value pair to describe Graph Nodes and Relationships.

Key = Value

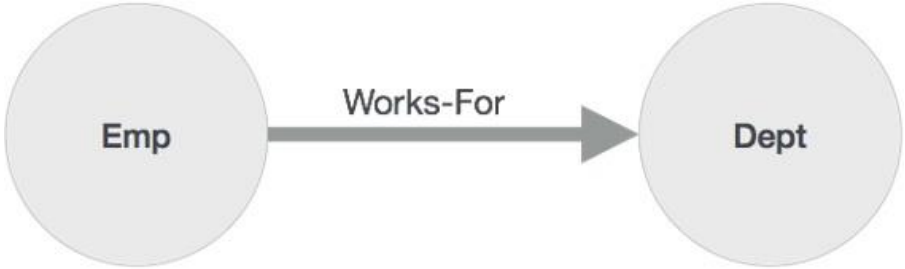
Where Key is a String and Value may be represented using any Neo4j Data types.

# Node


A Node
can have
can have
Relationships
can have
Labels
can have
Properties

# Relationships

Relationships are another major building block of a Graph Database. It connects two nodes as depicted in the following figure.



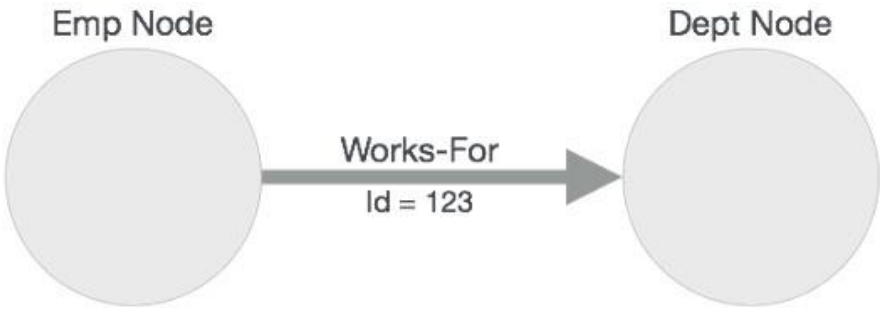

Here, Emp and Dept are two different nodes. "WORKS_FOR" is a relationship between Emp and Dept nodes. As it denotes, the arrow mark from Emp to Dept, this relationship describes −

Emp WORKS_FOR Dept

Each relationship contains one start node and one end node. Here, "Emp" is a start node, and "Dept" is an end node.

# Relationships

As this relationship arrow mark represents a relationship from "Emp" node to "Dept" node, this relationship is known as an "Incoming Relationship" to "Dept" Node and "Outgoing Relationship" to "Emp" node.



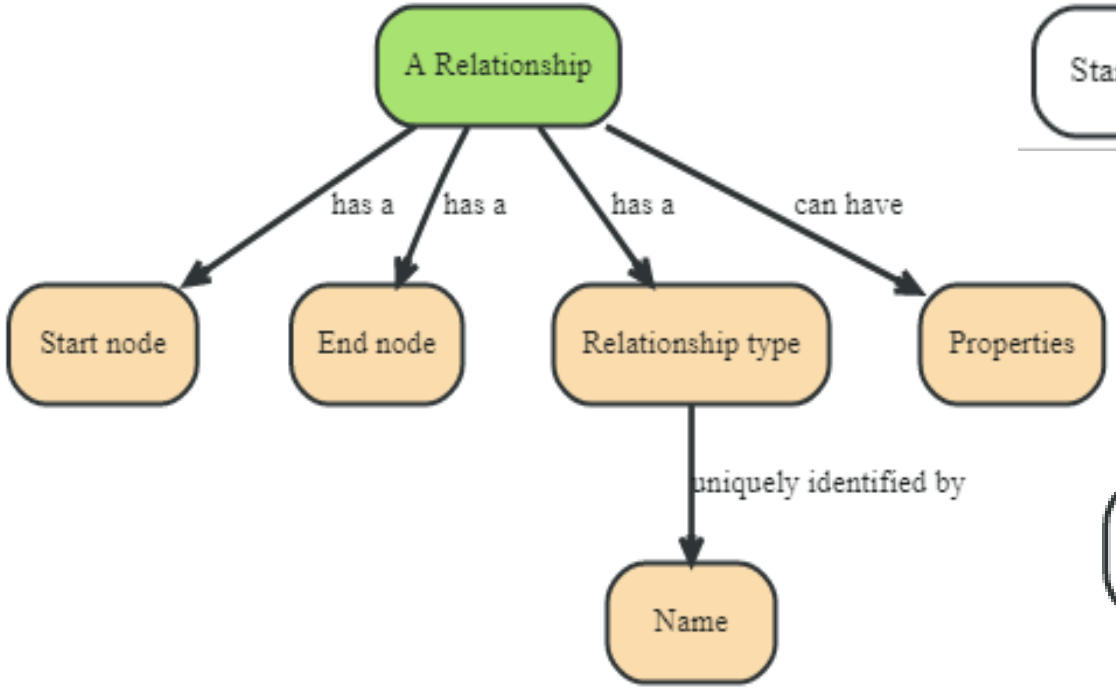

Like nodes, relationships also can contain properties as key-value pairs.

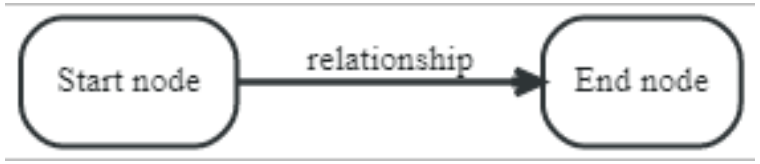
Here, "WORKS_FOR" relationship has one property as key-value pair.

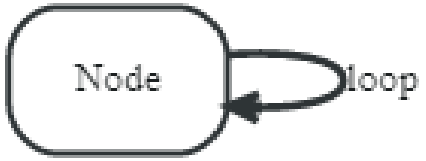
Id = 123

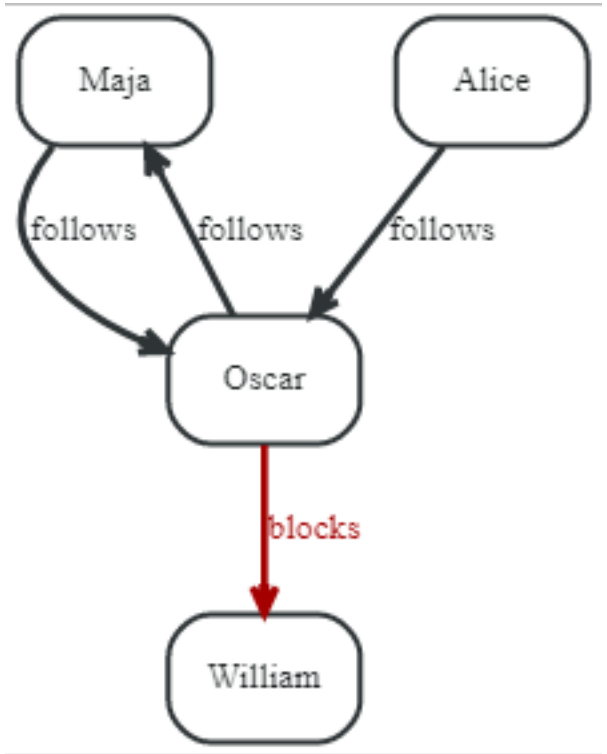
It represents an Id of this relationship.

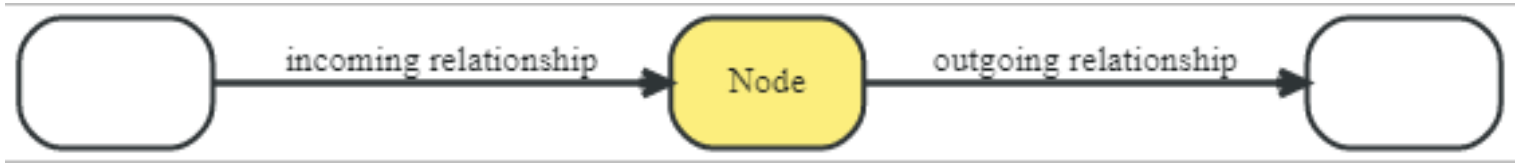
# Relationships



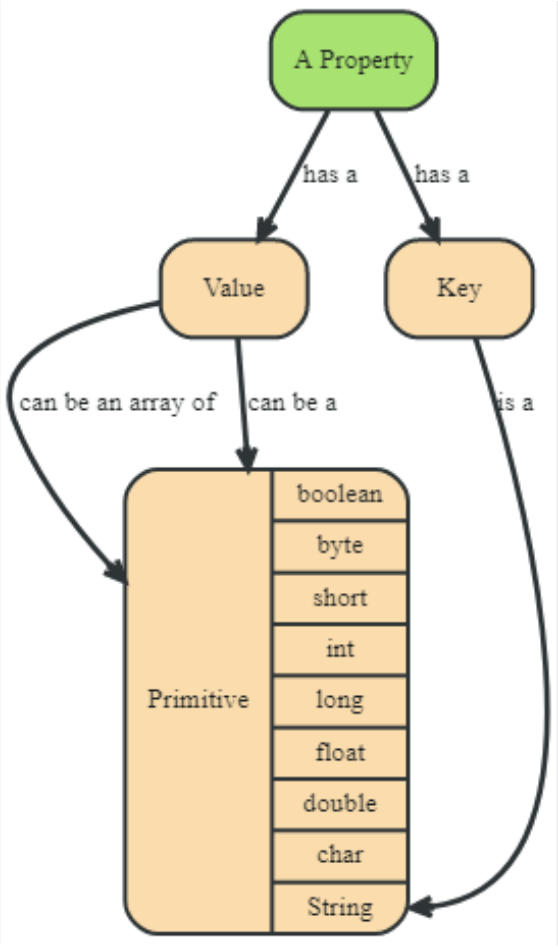



# Labels

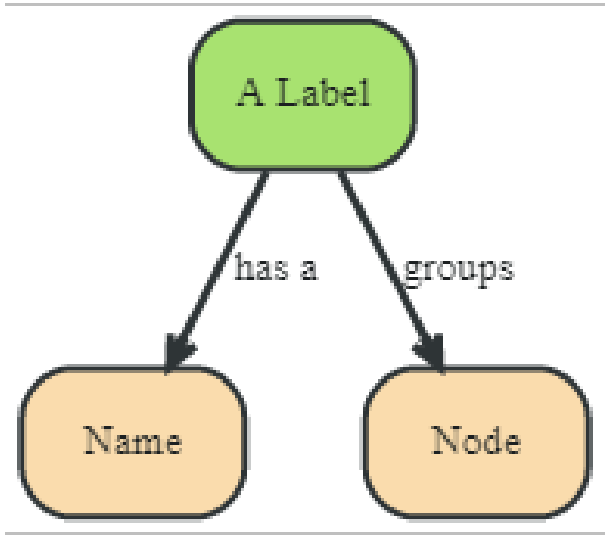
Label associates a common name to a set of nodes or relationships. A node or relationship can contain one or more labels. We can create new labels to existing nodes or relationships. We can remove the existing labels from the existing nodes or relationships.

- From the previous diagram, we can observe that there are two nodes.
- Left side node has a Label: "Emp" and the right side node has a Label: "Dept".
- Relationship between those two nodes also has a Label: "WORKS_FOR".
- Note − Neo4j stores data in Properties of Nodes or Relationships.

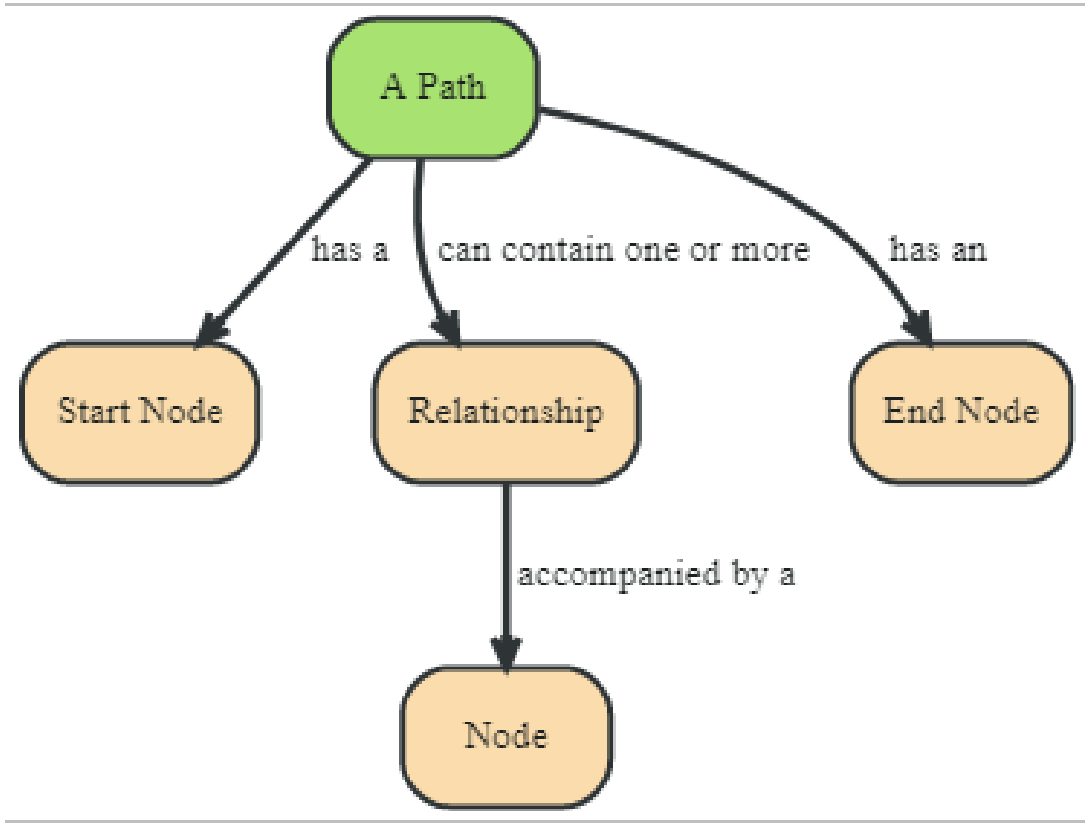
# Properties

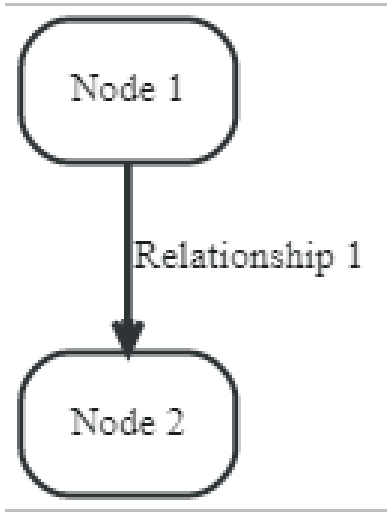



# Label



# Paths


A Path
has a
can contain one or more
has an
Start Node
Relationship
End Node
accompanied by a
Node

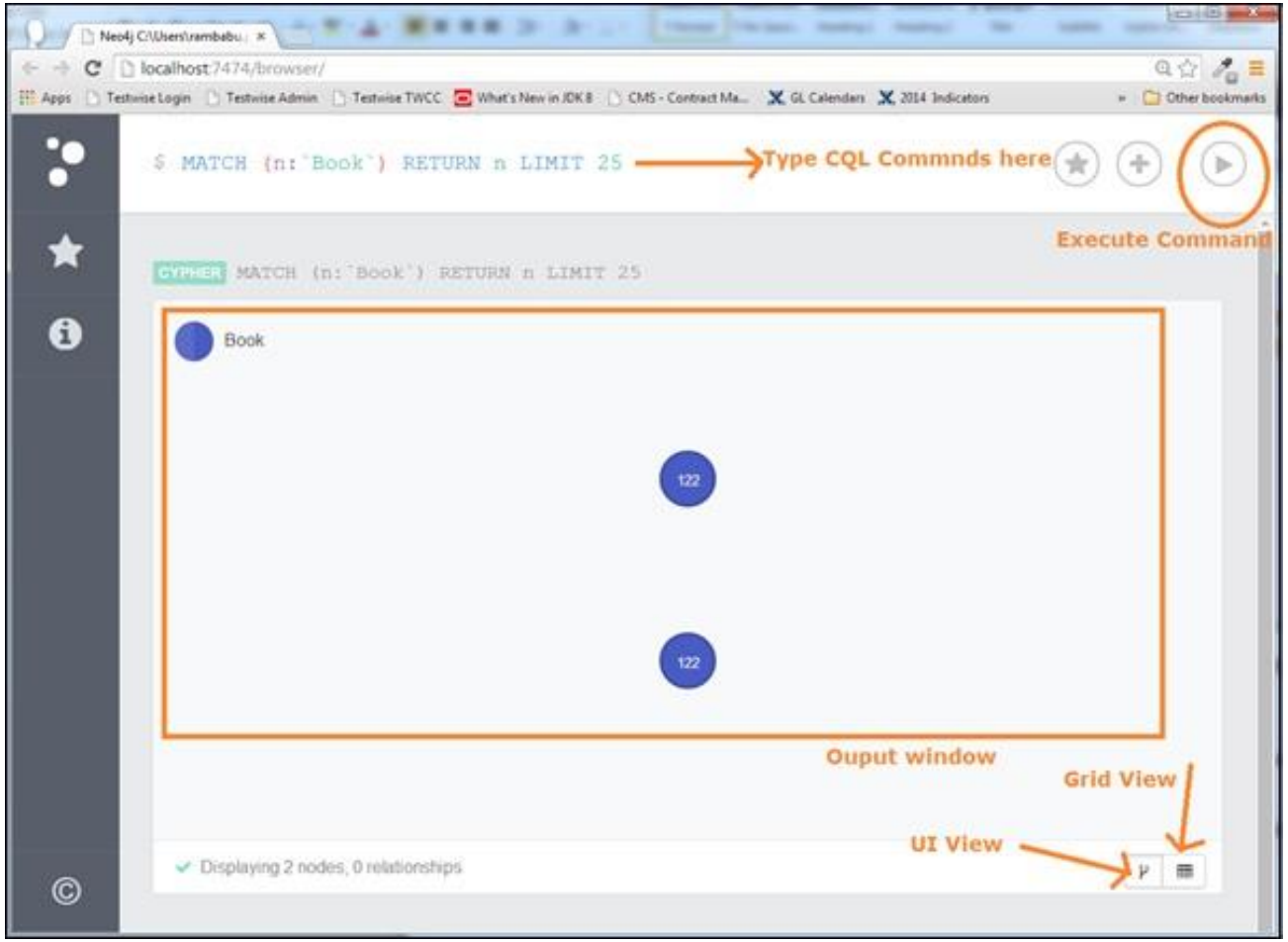

Node 1
Relationship 1
Node 2


Node 1
Relationship 1

# Neo4j Data Browser

Once we install Neo4j, we can access Neo4j Data Browser using the following URL:

http://localhost:7474/browser/

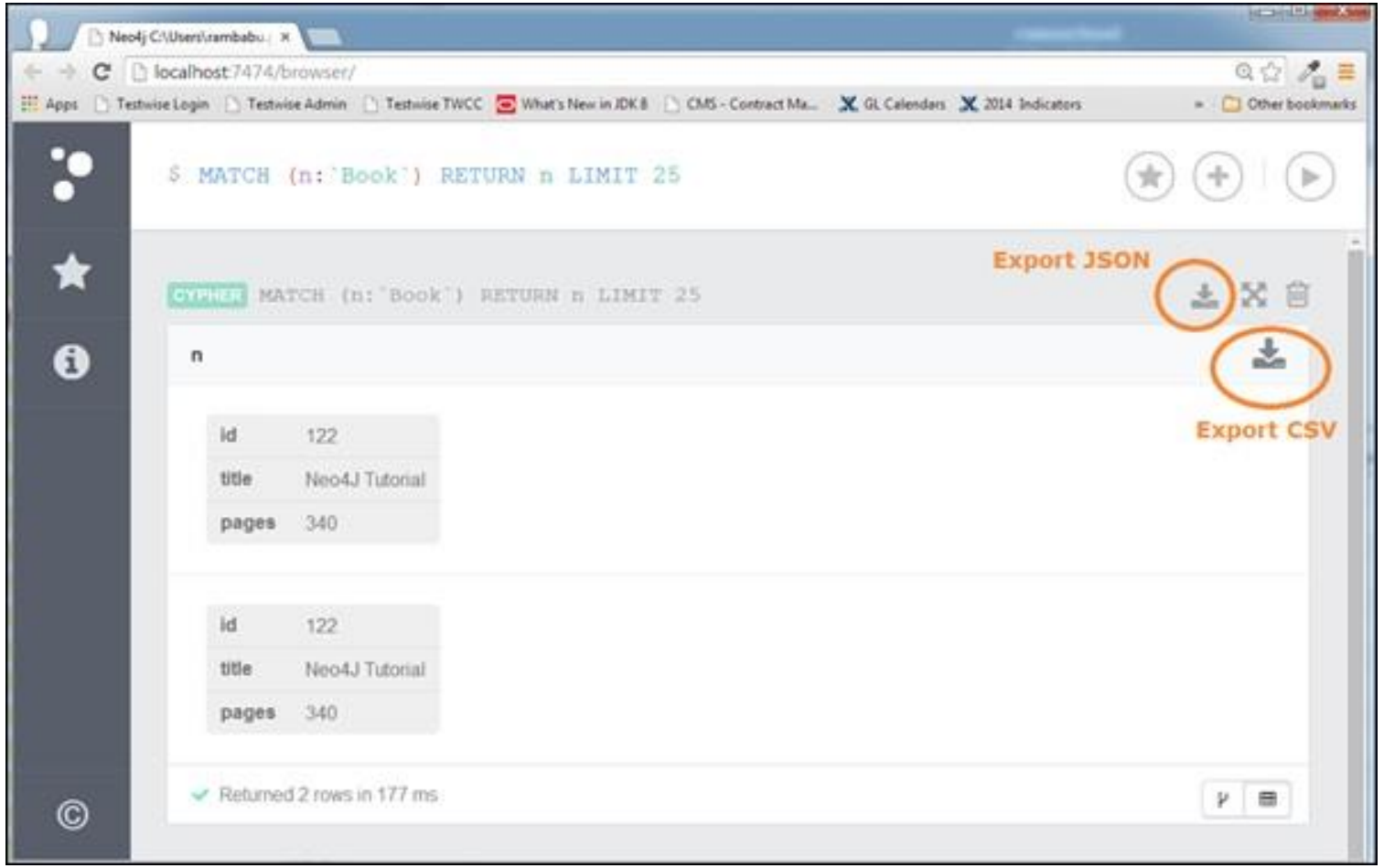


# Neo4j Data Browser

- Neo4j Data Browser is used to execute CQL commands and view the output.
- Here, we need to execute all CQL commands at dollar prompt: "$"
- Type commands after the dollar symbol and click the "Execute" button to run your commands.
- It interacts with Neo4j Database Server, retrieves and displays the results just below the dollar prompt.
- Use "VI View" button to view the results in diagrams format. The above diagram shows results in "UI View" format.
- Use "Grid View" button to view the results in Grid View. The following diagram shows the same results in "Grid View" format.

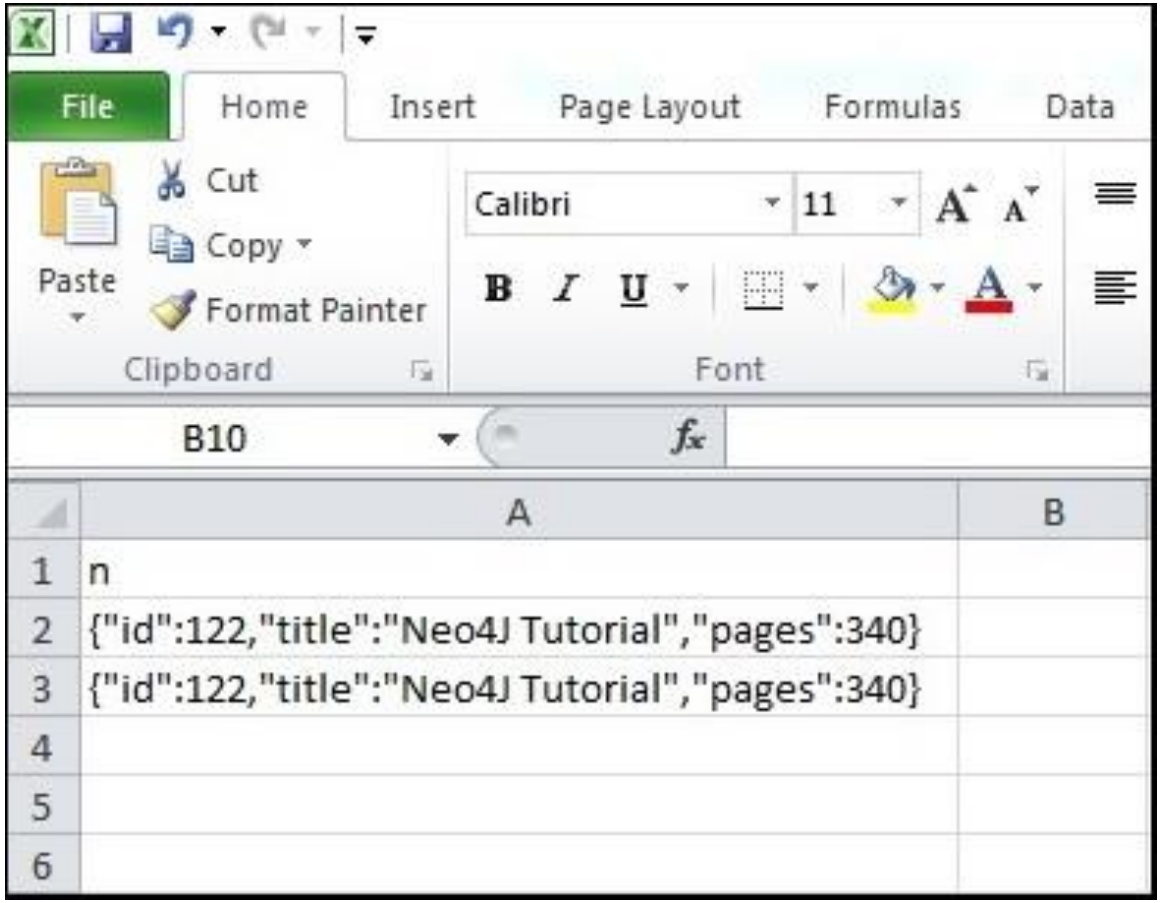
# Neo4j Data Browser

When we use "Grid View" to view our Query results, we can export them into a file in two different formats.



# Neo4j export CSV

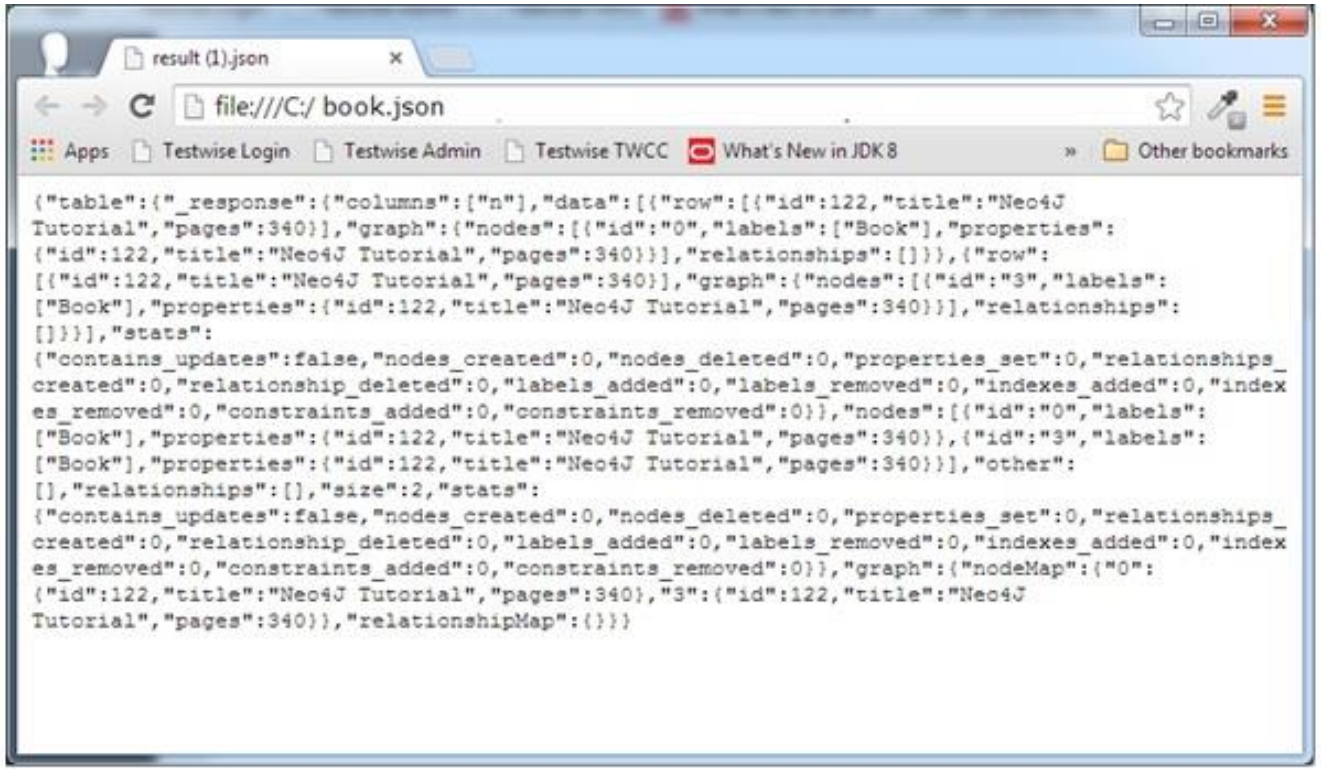
Click the "Export CSV" button to export the results in csv file format.



# Neo4j export JSON

Click the "Export JSON" button to export the results in JSON file format.

However, if we use "UI View" to see our Query results, we can export them into a file in only one format: JSON



# Neo4j - CQL

CQL stands for Cypher Query Language. Like Oracle Database has query language SQL, Neo4j has CQL as query language.

Neo4j CQL

- Is a query language for Neo4j Graph Database.
- Is a declarative pattern-matching language.
- Follows SQL like syntax.
- Syntax is very simple and in human readable format.

# Oracle SQL VS CQL

- Neo4j CQL has commands to perform Database operations.
- Neo4j CQL supports many clauses such as WHERE, ORDER BY, etc., to write very complex queries in an easy manner.
- Neo4j CQL supports some functions such as String, Aggregation. In addition to them, it also supports some Relationship Functions.

# Neo4j CQL: Write Clauses

- Following are the write clauses of Neo4j Cypher Query Language

| Sr.No | Write Clause | Usage |
|---|---|---|
| 1 | CREATE | This clause is used to create nodes, relationships, and properties. |
| 2 | MERGE | This clause verifies whether the specified pattern exists in the graph. If not, it creates the pattern. |
| 3 | SET | This clause is used to update labels on nodes, properties on nodes and relationships. |
| 4 | DELETE | This clause is used to delete nodes and relationships or paths etc. from the graph. |

# Neo4j CQL : Write Clauses

| Sr.No | Write Clause | Usage |
|---|---|---|
| 5 | REMOVE | This clause is used to remove properties and elements from nodes and relationships. |
| 6 | FOREACH | This class is used to update the data within a list. |
| 7 | CREATE UNIQUE | Using the clauses CREATE and MATCH, you can get a unique pattern by matching the existing pattern and creating the missing one. |
| 8 | Importing CSV files with Cypher | Using Load CSV you can import data from .csv files. |

# Neo4j CQL : Read Clauses

- Following are the read clauses of Neo4j Cypher Query Language

| Sr.No | Read Clauses | Usage |
|---|---|---|
| 1 | MATCH | This clause is used to search the data with a specified pattern. |
| 2 | OPTIONAL MATCH | This is the same as match, the only difference being it can use nulls in case of missing parts of the pattern. |
| 3 | WHERE | This clause id is used to add contents to the CQL queries. |
| 4 | START | This clause is used to find the starting points through the legacy indexes. |
| 5 | LOAD CSV | This clause is used to import data from CSV files. |

# Neo4j CQL: General Clauses

- Following are the general clauses of Neo4j Cypher Query Language

| Sr.No | General Clauses | Usage |
|---|---|---|
| 1 | RETURN | This clause is used to define what to include in the query result set. |
| 2 | ORDER BY | This clause is used to arrange the output of a query in order. It is used along with the clauses **RETURN** or **WITH**. |
| 3 | LIMIT | This clause is used to limit the rows in the result to a specific value. |

# Neo4j CQL: General Clauses

| Sr.No | General Clauses | Usage |
|---|---|---|
| 4 | SKIP | This clause is used to define from which row to start including the rows in the output. |
| 5 | WITH | This clause is used to chain the query parts together. |
| 6 | UNWIND | This clause is used to expand a list into a sequence of rows. |
| 7 | UNION | This clause is used to combine the result of multiple queries. |
| 8 | CALL | This clause is used to invoke a procedure deployed in the database. |

# Neo4j CQL Functions

- Following are the frequently used Neo4j CQL Functions

| Sr.No | CQL Functions | Usage |
|---|---|---|
| 1 | String | They are used to work with String literals. |
| 2 | Aggregation | They are used to perform some aggregation operations on CQL Query results. |
| 3 | Relationship | They are used to get details of relationships such as startnode, endnode, etc. |

# Neo4j CQL Data Types

| Sr.No | CQL Data Type | Usage |
|---|---|---|
| 1 | Boolean | It is used to represent Boolean literals: true, false. |
| 2 | byte | It is used to represent 8-bit integers. |
| 3 | short | It is used to represent 16-bit integers. |
| 4 | int | It is used to represent 32-bit integers. |
| 5 | long | It is used to represent 64-bit integers. |
| 6 | float | It is used to represent 32-bit floating-point numbers. |
| 7 | double | It is used to represent 64-bit floating-point numbers. |
| 8 | char | It is used to represent 16-bit characters. |
| 9 | String | It is used to represent Strings. |

# Neo4j CQL Operators

| Sr.No | Type | Operators |
|---|---|---|
| 1 | Mathematical | +, -, *, /, %, ^ |
| 2 | Comparison | +, <>, <, >, <=, >= |
| 3 | Boolean | AND, OR, XOR, NOT |
| 4 | String | + |
| 5 | List | +, IN, [X], [X…..Y] |
| 6 | Regular Expression | =- |
| 7 | String matching | STARTS WITH, ENDS WITH, CONSTRAINTS |

# Neo4j CQL Boolean Operators

| Sr.No | Boolean Operators | Description |
|---|---|---|
| 1 | AND | It is a Neo4j CQL keyword to support AND operation.<br>It is like SQL AND operator. |
| 2 | OR | It is a Neo4j CQL keyword to support OR operation.<br>It is like SQL AND operator. |
| 3 | NOT | It is a Neo4j CQL keyword to support NOT operation.<br>It is like SQL AND operator. |
| 4 | XOR | It is a Neo4j CQL keyword to support XOR operation.<br>It is like SQL AND operator. |

# Neo4j CQL Comparison Operators

| Sr.No | Boolean Operators | Description |
|---|---|---|
| 1 | = | It is a Neo4j CQL "Equal To" operator. |
| 2 | < > | It is a Neo4j CQL "Not Equal To" operator. |
| 3 | < | It is a Neo4j CQL "Less Than" operator. |
| 4 | > | It is a Neo4j CQL "Greater Than" operator. |
| 5 | <= | It is a Neo4j CQL "Less Than Or Equal To" operator. |
| 6 | > = | It is a Neo4j CQL "Greater Than Or Equal To" operator. |

# Summary

This session will give the knowledge about

- Neo4j Building Blocks
- Cypher Query Language (CQL) Introduction