Tutorial No.1

Content:

Requirements:

- Internet Connection
- Setup

Note: I am using Neo4j Desktop v 1.2.7

Introduction

Neo4j is the world's leading graph database. Its architecture is designed for optimal management, storage, and traversal of nodes and relationships. The database takes a property graph approach, which is beneficial for both traversal performance and operations runtime. Neo4j offers dedicated memory management and memory efficient operations.

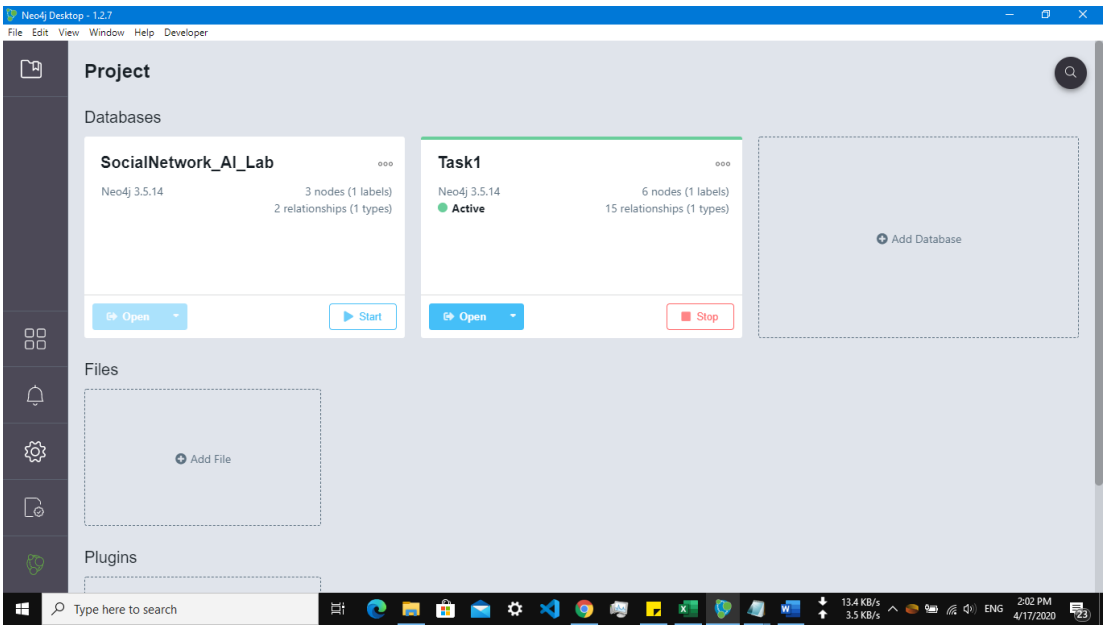
Setup Link:

You can download Neo4j Desktop from https://neo4j.com/download/

Note: If anyone facing issues with Neo4j latest version then please install old version.

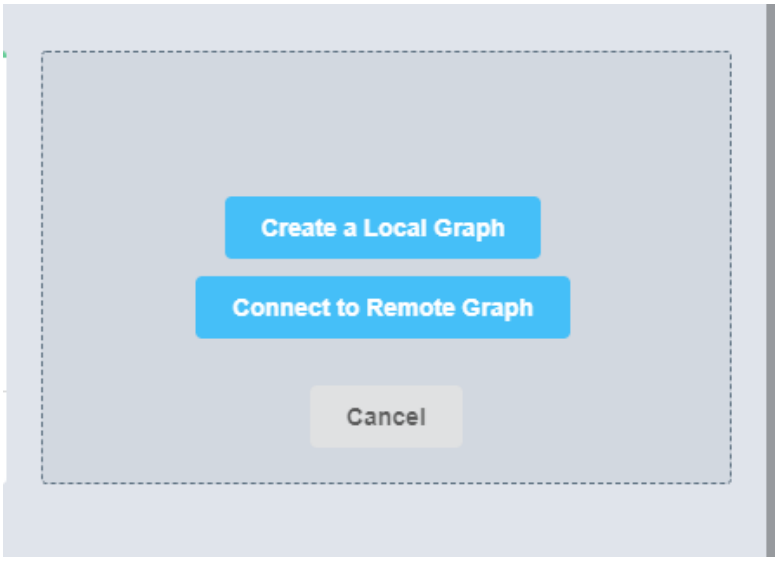
How to create Database and Start the server

1. Launch Neo4j Desktop
2. Click on Add Database:

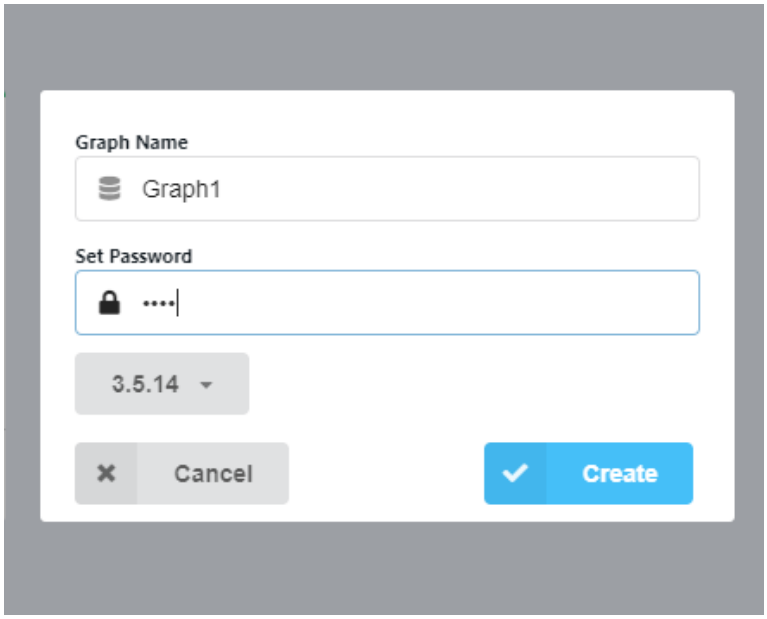


Note: If you have not created any database earlier then please press **Add Graph** from Dashboard and then press **Create a Local Graph** as you can show in Step3.

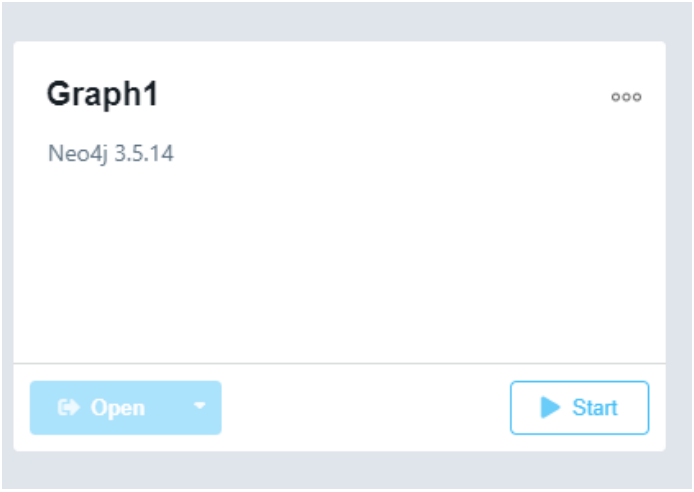
3. Press **Create a Local Graph.**



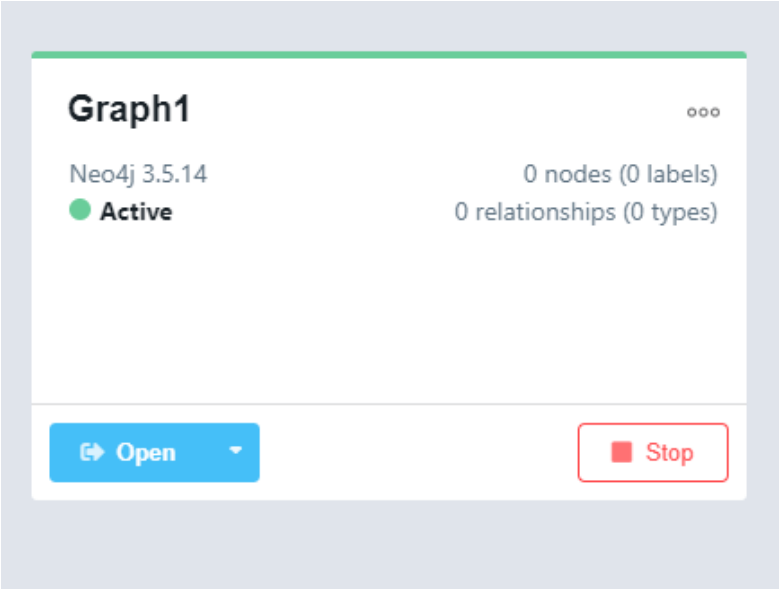
4. Enter **Graph Name** and **Password** for DB.(Please Remember the credentials).



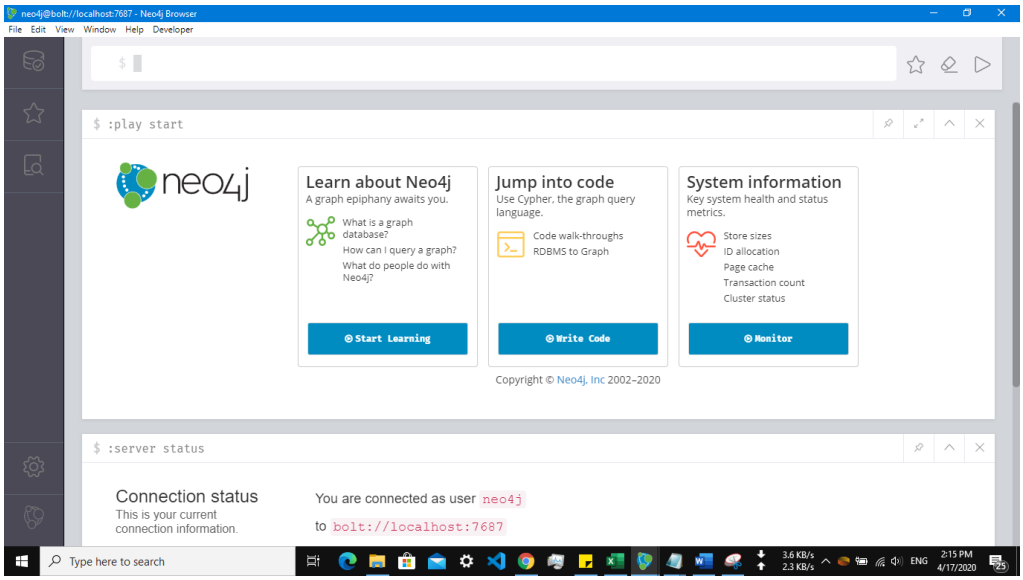
5. Wait until Graph is being creating.
6. Now DB has been created. And to launch it press **Start** Button.



7. Now DB has been started. Click Open to Launch Neo4j Browser and run queries.

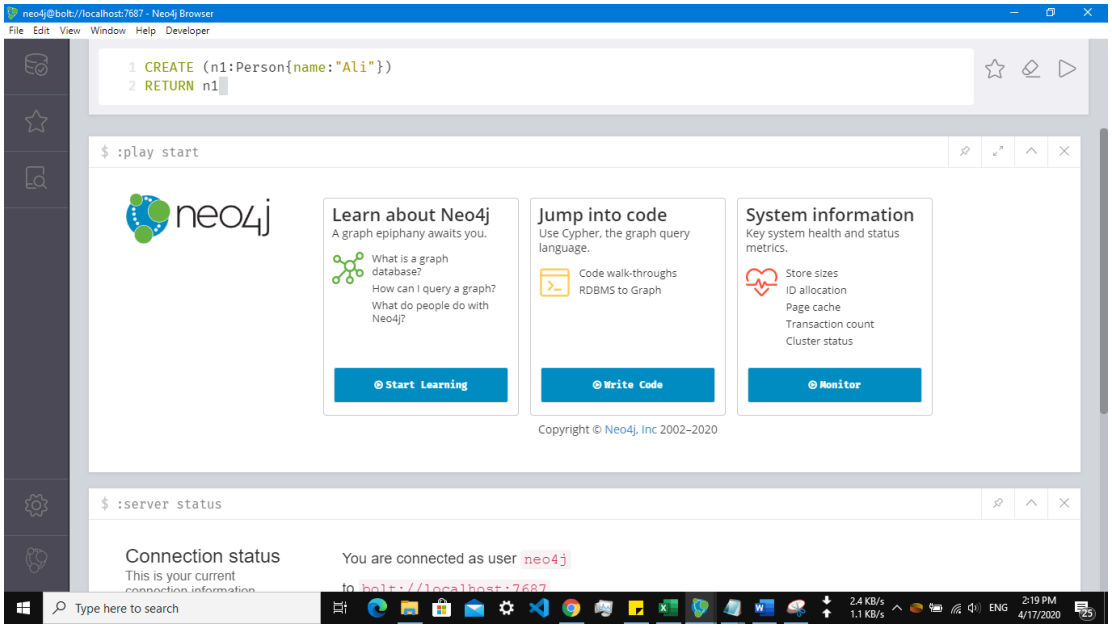


8. Now this screen will be visible to you. Here we will write all queries.

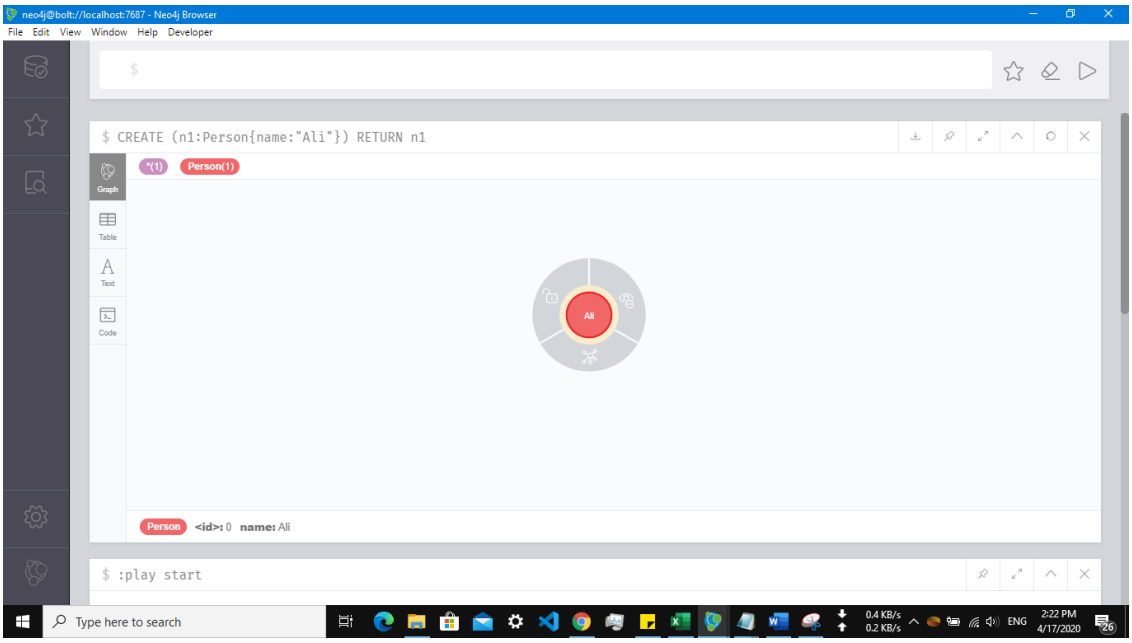


Create Single Node

1. We will create a Node of Type Person and with attributes name
2. **Syntax** of Creating Node is:
   CREATE(temporary-variable: Node-Type{attribute: value})
3. After writing CREATE statement and then RETURN temporary-variable. After clicking on **Run Button**, our single Node will be created and shown to you with return statement.

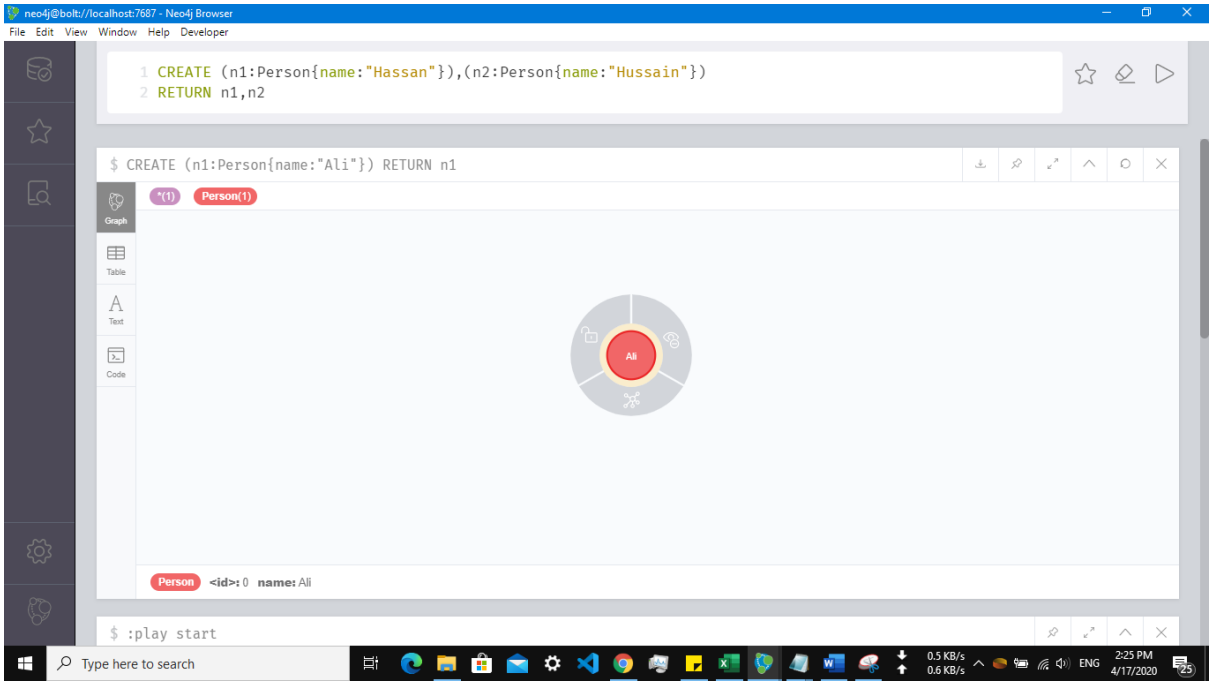


4. As you can see in Figure Node is created and at bottom you can see Node type is Person and name of Person is Ali.

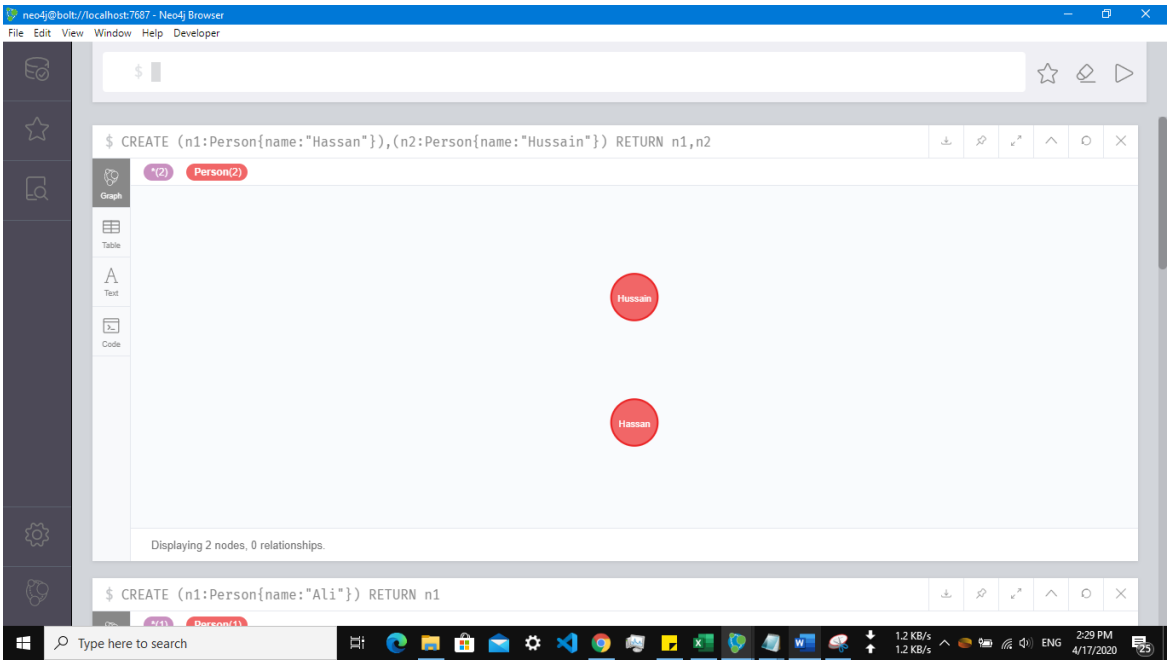


Create Multiple Nodes:

1. Now if we want to create multiple nodes in single line then we will use **syntax**:
   CREATE (temp-var1: Node-Type {attribute: value}),(temp-var2: Node-Type {attribute: value})

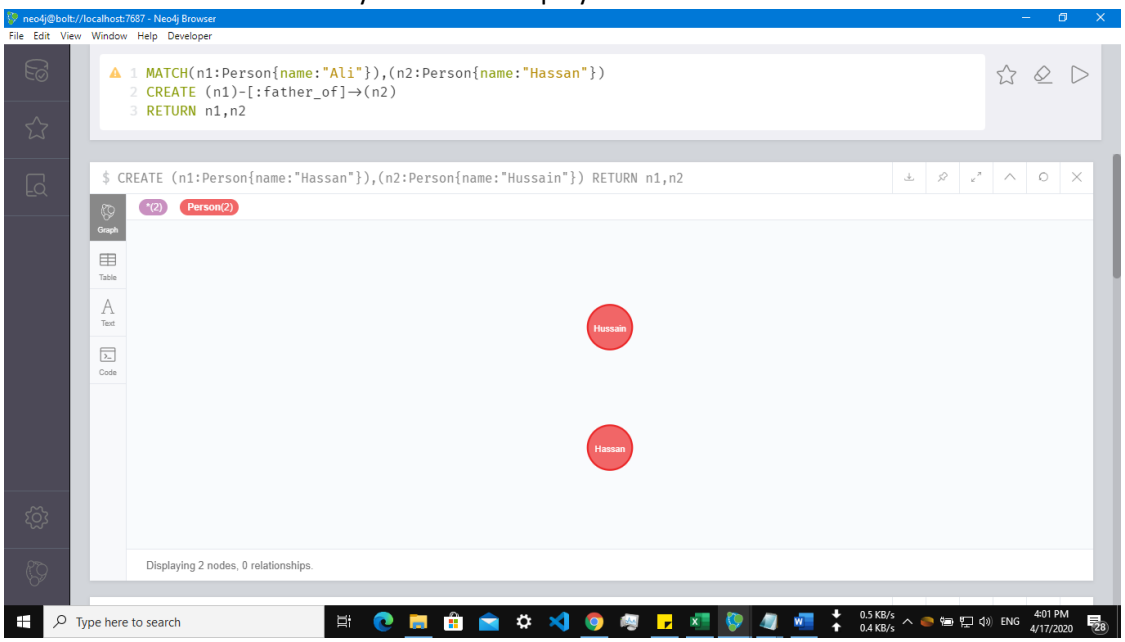


2. Now we have only RETURN n1,n2 i.e, Hassan and Hussain so only Hassan and Hussain Node will be visible to us. Rest of nodes are stored in database but not visible.

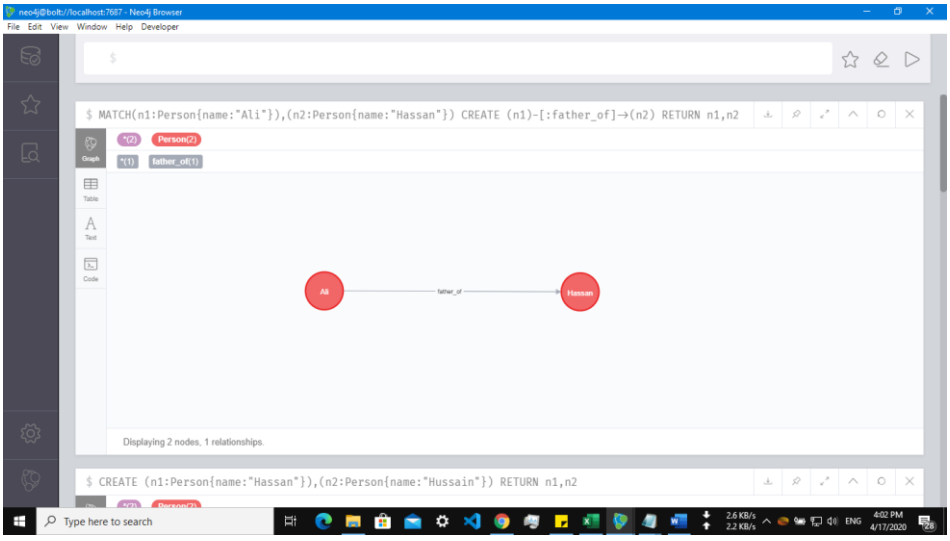


Create Relationships

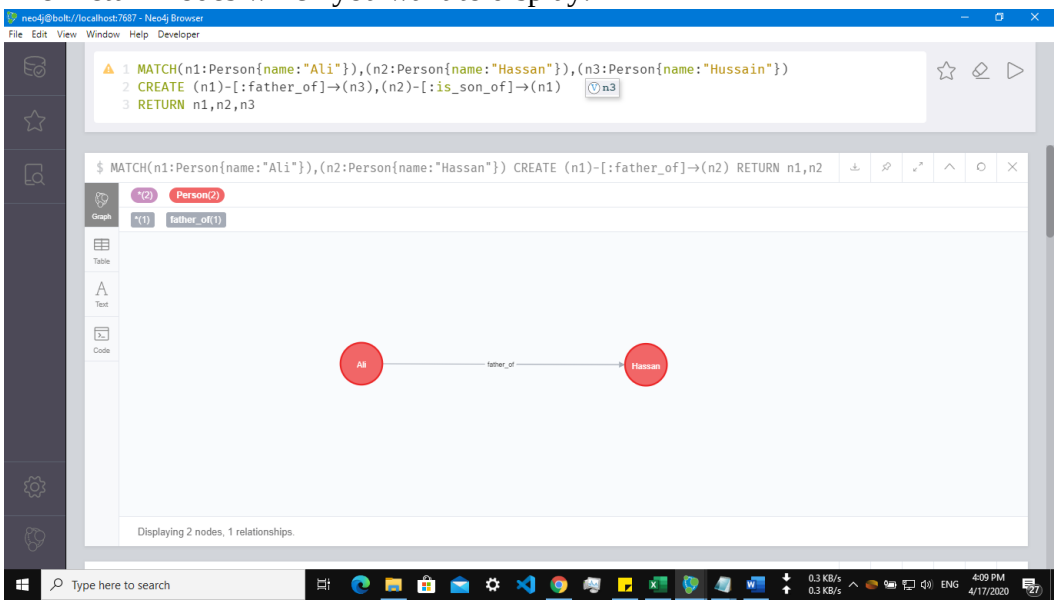
- **Single Relation At a time:**
  1. First we have to Match nodes from Database i.e., (Find them from DB).
  2. Then we have to create Relationship with Nodes by following **syntax**:
     CREATE (Source-Node-temp-var-used-in-match)-[:relationtype]->(Destination-node-temp-var)
  3. Then return nodes which you want to display.



4. After launching Command,



- **More than one Relations at a time:**
  1. First we have to Match nodes from Database i.e., (Find them from DB).
  2. Then we have to create Relationship with Nodes by following **syntax**: CREATE (Source-Node-temp-var-used-in-match)-[:relationtype]->(Destination-node-temp-var), (Source-Node2-temp-var-used-in-match)-[:relationtype]->(Destination-node2-temp-var)
  3. Then return nodes which you want to display.



Delete Single Node:

Note: If your node have relationships with other nodes, then you cant delete that node. You have to remove relationships first.

1. First match it from DB and then use Delete node-var.

```
MATCH(n2:Person{name:"Hussain"})
DELETE n2
```

Delete Multiple Nodes:

1. First match it from DB and then use delete node-variables.

```
MATCH(n2:Person{name:"Hussain"}),(n3:Person{name:"Ali"})
DELETE n2,n3
```

Delete All Nodes With Relationships

1. First match it from DB. And then use Delete variable.

```
MATCH(Person)
DETACH DELETE (Person)
```

Delete Single Relationship:

1. First Match it from DB. And then use delete relation-var.

```
MATCH(n1:Person{name:"Ali"})-[r1:father_of]→(n2:Person{name:"Hussain"})
DELETE r1
```