

WEEK2: INTRODUCTION TO GRAPH DATABASE


NAME: TUSHAR Y S

SRN : PES1UG19CS545 (Section: I)

1. Create node and relationships:


- Create a single node:

```
$ create(n)
```



Table


Created 1 node, completed after 271 ms.



Code


- Create multiple nodes:

```
$ create(n),(m)
```



Table


Created 2 nodes, completed after 15 ms.



Code


- Create a node with a label:

```
$ create(n:Person)
```



Table

Added 1 label, created 1 node, completed after 89 ms.



Code

- View nodes:

The screenshot shows the Neo4j Browser interface. The top bar indicates the user is signed in as 'PES'. The browser address bar shows 'localhost:7474/browser/'. The main query editor contains the Cypher query: `$ match(n) return(n)`. Below the query, the 'Graph' view is selected, showing a graph with 4 nodes and 0 relationships. The nodes are represented by circles: three grey and one pink. The status bar at the bottom indicates 'Displaying 4 nodes, 0 relationships.' The left sidebar contains icons for Graph, Table, Text, and Code views.

- Create a node with multiple labels:

The screenshot shows the Neo4j Browser interface with the Cypher query `$ create(n:tush:yst)` entered in the query editor. The 'Table' view is selected, displaying the result of the query: 'Added 2 labels, created 1 node, completed after 114 ms.' The left sidebar contains icons for Graph, Table, Text, and Code views.

The screenshot shows the Neo4j Browser interface. The top bar indicates the user is signed in as 'PES'. The browser address bar shows 'localhost:7474/browser/'. The main query editor contains the Cypher query: `$ match(n) return(n)`. Below the query, the 'Graph' view is selected, showing a graph with 5 nodes and 0 relationships. The nodes are represented by circles: three grey, one pink, and one blue. The status bar at the bottom indicates 'Displaying 5 nodes, 0 relationships.' The left sidebar contains icons for Graph, Table, Text, and Code views.

- Create node and add labels and properties:

```
$ create(n:Person{name:'Andy',title:'Developer'})
```

Table

Code

Added 1 label, created 1 node, set 2 properties, completed after 191 ms.

You are signed in as PES X neo4j@bolt://localhost: X Inbox (2,383) - ystushar X

localhost:7474/browser/

Graph

Table

Text

Code

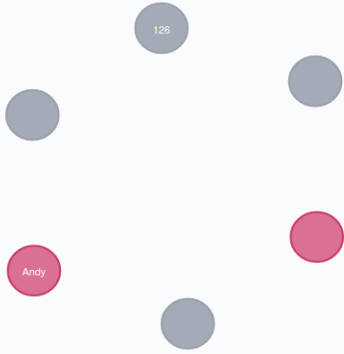
\$ match(n) return(n)

*(4)

Person(2)

tush(1)

yst(1)

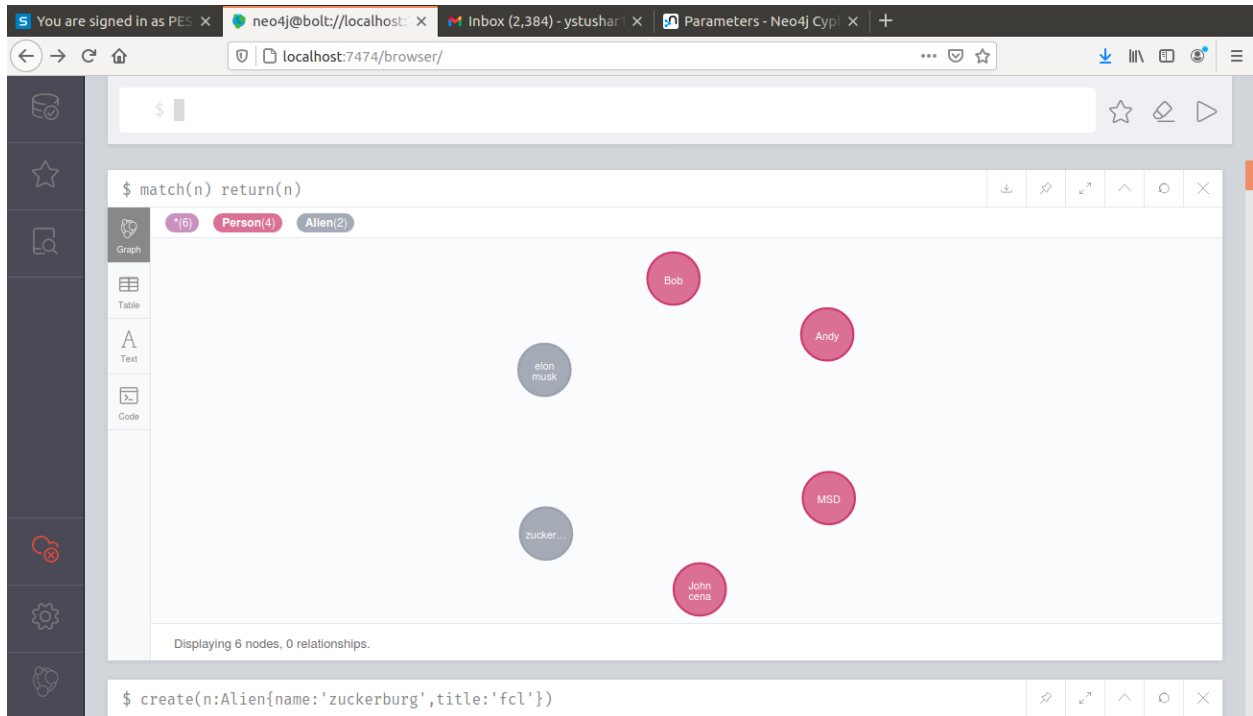


Displaying 6 nodes, 0 relationships.

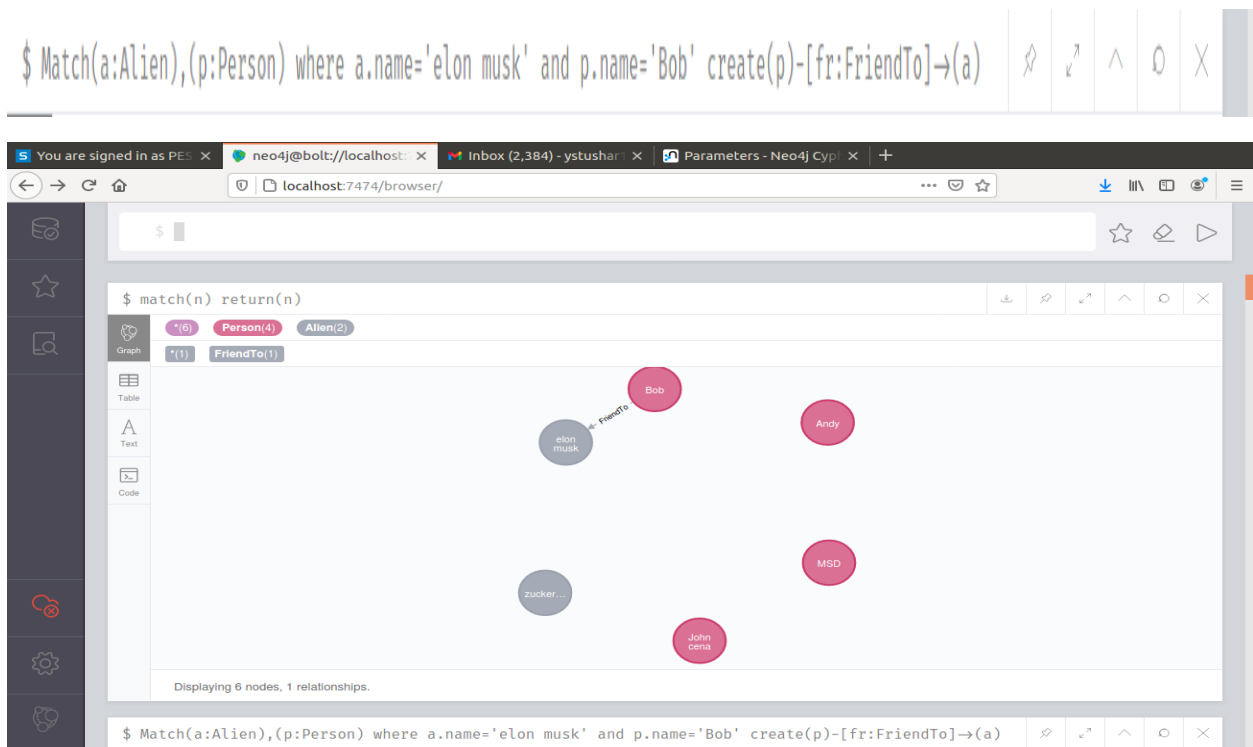
\$ create(n:Person{name:'Andy',title:'Developer'})

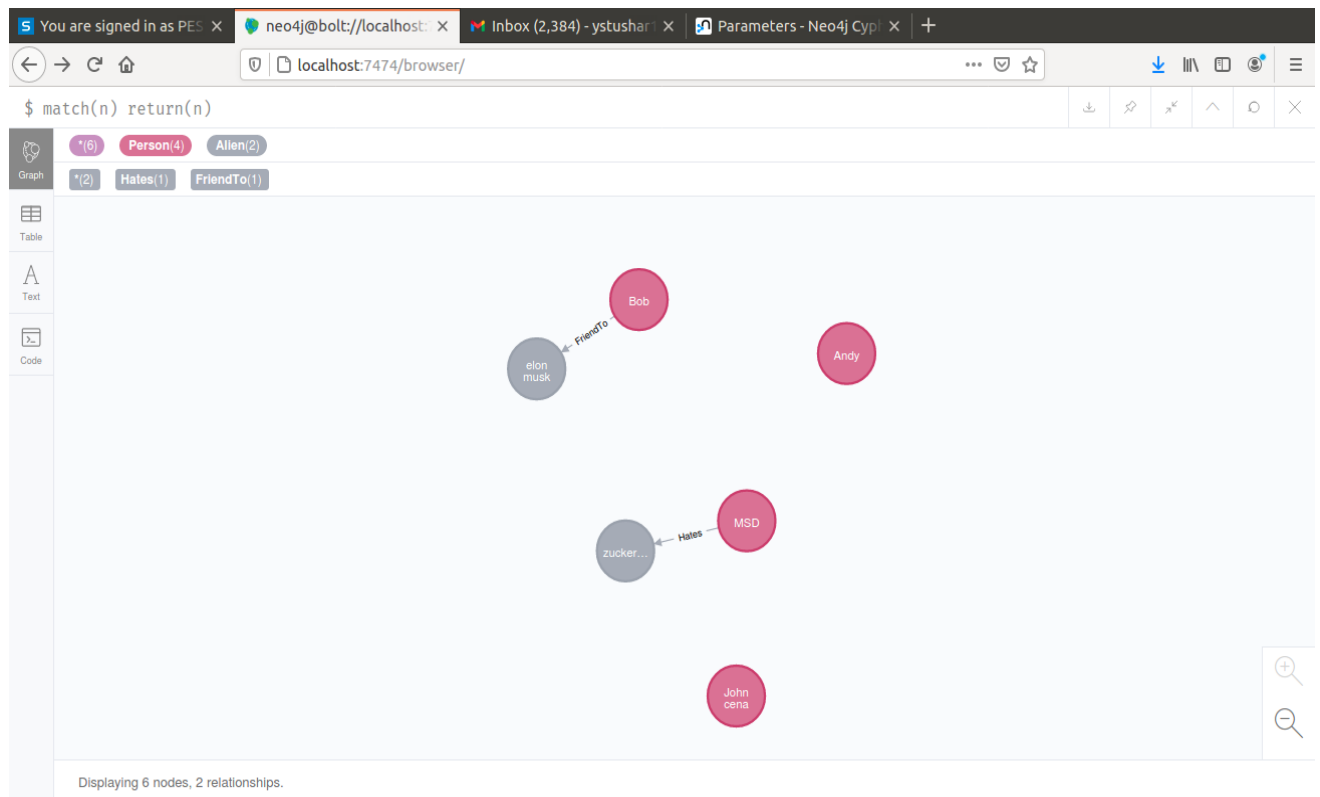
2. Create Relationships between the nodes:

- Before:



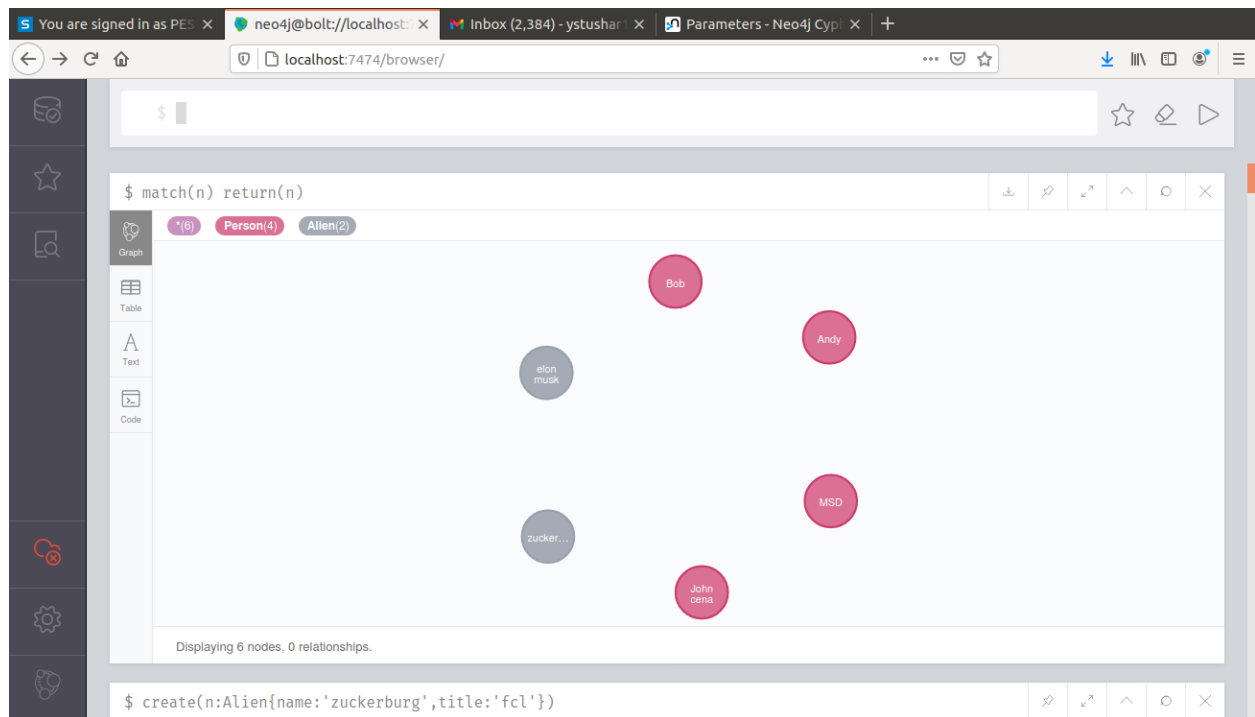
- After establishing relationships:





3. Read nodes and attributes:

- Get all nodes:



- Get all nodes with a label:

The screenshot shows the Neo4j Browser interface. The browser's address bar displays `localhost:7474/browser/`. The left sidebar contains navigation icons for Graph, Table, Text, and Code. The main workspace shows a Cypher query `$ match(per:Person) return(per)` in the top editor. Below the query, a status bar indicates `*(4) Person(4)`. The graph view displays four nodes: Andy, Bob, MSD, and John cena, each represented by a pink circle. A status bar at the bottom of the graph view states "Displaying 4 nodes, 0 relationships." The bottom editor shows a second query `$ match(n) return(n)`.

- Related nodes:

The screenshot shows the Neo4j Browser interface. The browser's address bar displays `localhost:7474/browser/`. The left sidebar contains navigation icons for Graph, Table, Text, and Code. The main workspace shows a Cypher query `$ match(Person{name:"Bob"})--(aaa) return aaa` in the top editor. Below the query, a status bar indicates `*(1) Allen(1)`. The graph view displays a single node labeled "elon musk" in a grey circle. A status bar at the bottom of the graph view states "Displaying 1 nodes, 0 relationships." The bottom editor shows the same query `$ match(Person{name:"Bob"})--aaa return aaa`.

4. Update or set a value:

```
$ match(n:Person) Set n.id='newvalue'
```

Table

Set 4 properties, completed after 73 ms.

Code

You are signed in as PES X neo4j@bolt://localhost: X Inbox (2,385) - ystushar1 X Parameters - Neo4j Cyp X +

localhost:7474/browser/

\$

\$ match(n) return(n)

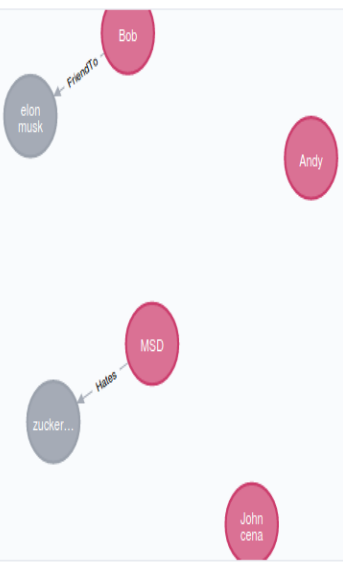
(6) Person(4) Allen(2)
(2) Hates(1) FriendTo(1)

Graph

Table

Text

Code



```
graph LR; elon_musk((elon musk)) -- FriendTo --> Bob((Bob)); zucker_...((zucker...)) -- Hates --> MSD((MSD)); Andy((Andy)); John_cena((John cena));
```

Displaying 6 nodes, 2 relationships.

\$ match(n:Person) Set n.id='newvalue'

5. Delete operation:

- Delete single node:

```
$ match(n:Person{name:'Andy'}) Delete(n)
```

Deleted 1 node, completed after 4 ms.

Table

Code

After deleting one node:

You are signed in as PES X neo4j@bolt://localhost: X Inbox (2,385) - ystushar X Parameters - Neo4j Cyp X +

localhost:7474/browser/

\$

* (5) Person(3) Alien(2)

* (2) Hates(1) FriendTo(1)

Graph

Table

Text

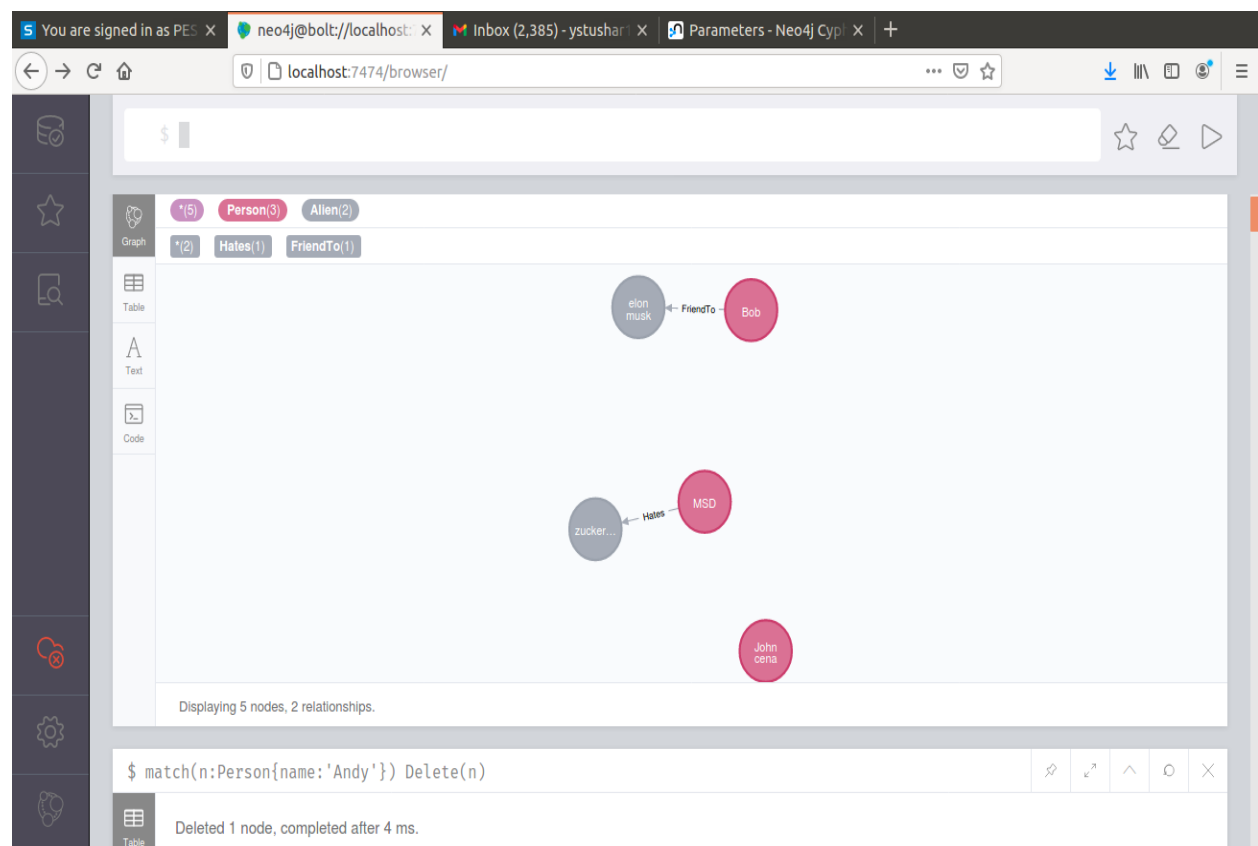
Code

Displaying 5 nodes, 2 relationships.

\$ match(n:Person{name:'Andy'}) Delete(n)

Deleted 1 node, completed after 4 ms.

Table



- Delete all nodes:

```
$ match(n) detach delete n
```

The screenshot shows the Neo4j Cypher console interface. On the left, there is a sidebar with icons for 'Table' (a grid) and 'Code' (a document with a cursor). The main area displays the result of a Cypher query: 'Deleted 5 nodes, deleted 2 relationships, completed after 67 ms.'

After deleting all nodes:

This screenshot shows the Neo4j Cypher console with two queries executed. The top query is `$ match(n) return(n)`, which returned '(no changes, no records)' and completed after 1 ms. The bottom query is `$ match(n) detach delete n`, which returned 'Deleted 5 nodes, deleted 2 relationships, completed after 67 ms.' The interface includes a browser window at the top with multiple tabs, a sidebar with navigation icons, and a main area for query execution with 'Table' and 'Code' views.