

# **Project Review**

## **NoSQL Database**

### **Neo4j**

#### **DUMBLEDORE'S ARMY**

##### **Team Members:**

**LAKSHAY NANDA (16BCE0805)**

**KESHIKA TANK (16BCE0796)**

**SHIVAM KHANNA (16BCI0106)**

## SNIPPETS OF THE CQL QUERY AND THE OUTPUT

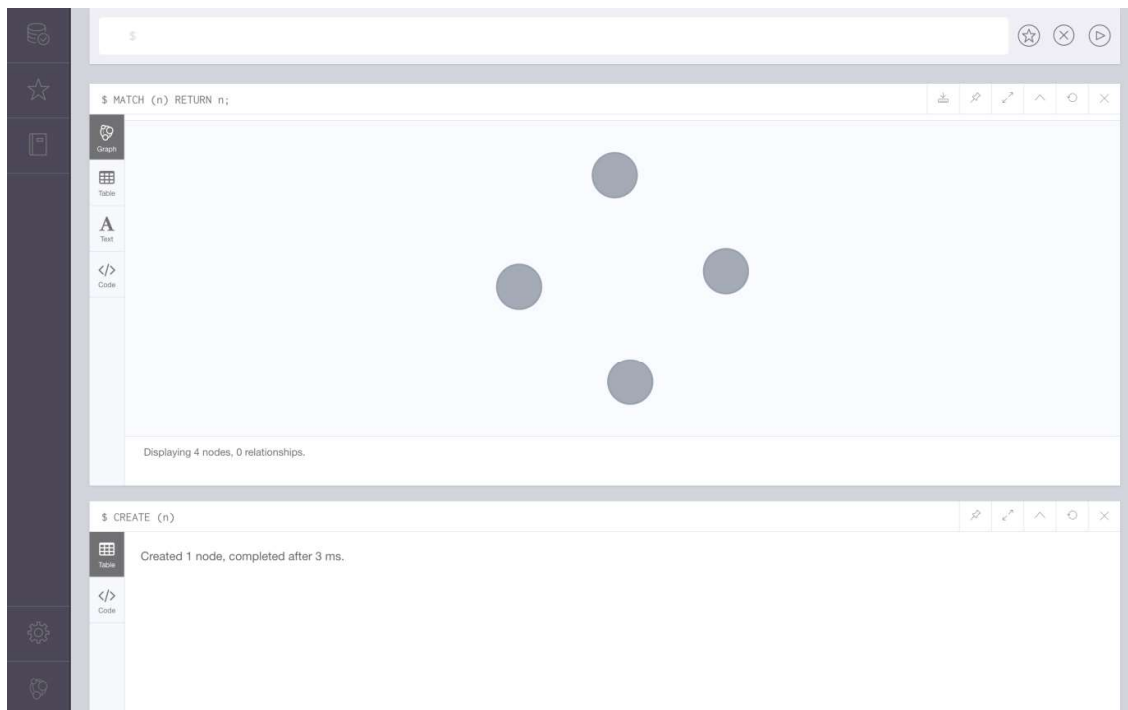
### 1) CREATE(n)

The screenshot shows the Neo4j web interface. At the top, a query editor contains the command `$ CREATE (n)`. Below the editor, a message states "Created 1 node, completed after 3 ms." The interface also features a sidebar with icons for database, star, and code. At the bottom, there is a section titled "neo4j" with three main cards: "Learn about Neo4j" (with a "Start Learning" button), "Jump into code" (with a "Write Code" button), and "Monitor the system" (with a "Monitor" button). The footer includes the text "Copyright © Neo4j, Inc 2002-2017".

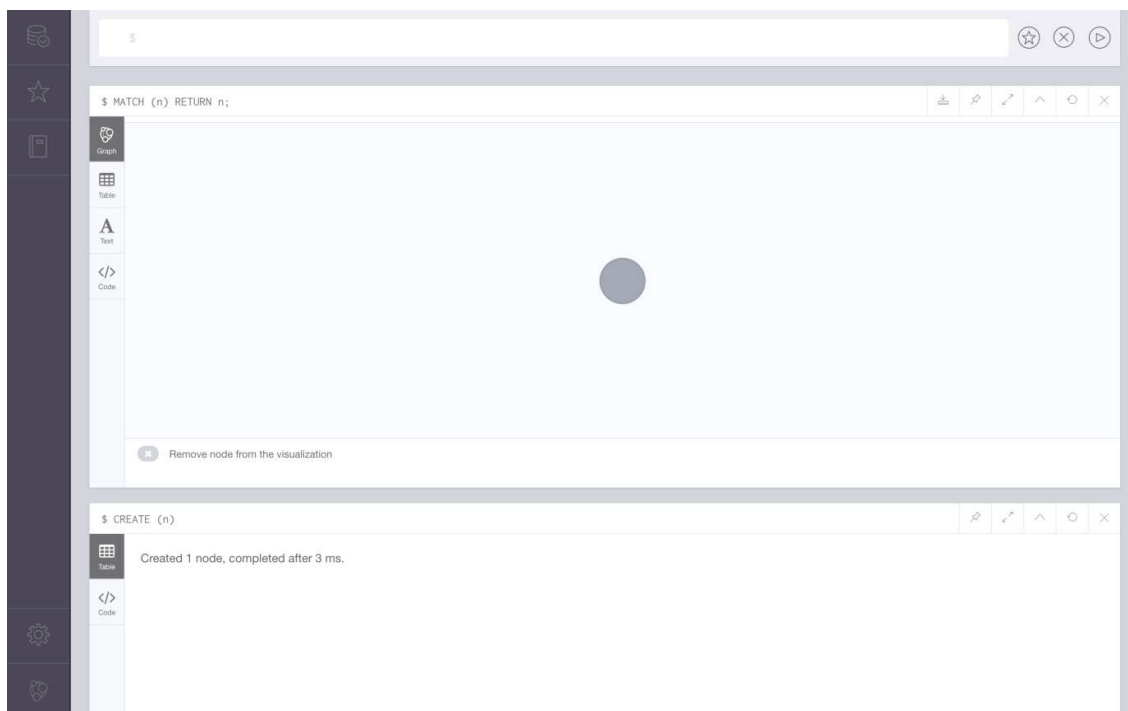
### 2) MATCH(n) RETURN n;

This screenshot is identical to the one above, showing the Neo4j web interface. The query editor at the top now contains the command `$ MATCH (n) RETURN n;`. The rest of the interface, including the sidebar, the "Created 1 node, completed after 3 ms." message, and the bottom section with "neo4j" cards and the footer, remains the same.

## 3) MATCH(n) RETURN n;



## 4) Match(n) REMOVE(n);



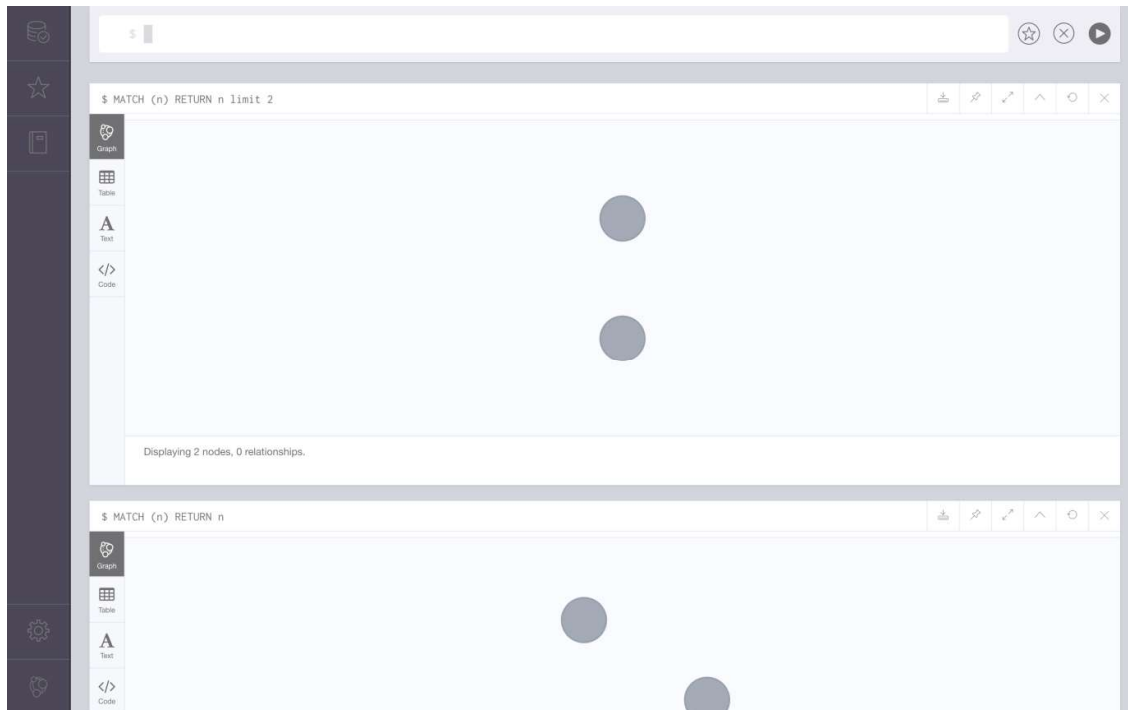
## 5) CREATE (n),(m);

The screenshot shows a Neo4j Cypher console interface. The top panel displays the Cypher query `$ CREATE (n),(m);`. Below the query, a message states "Created 2 nodes, completed after 1 ms." The bottom panel shows the same query `$ MATCH (n) RETURN n;` and a graph visualization with a single dark grey circular node.

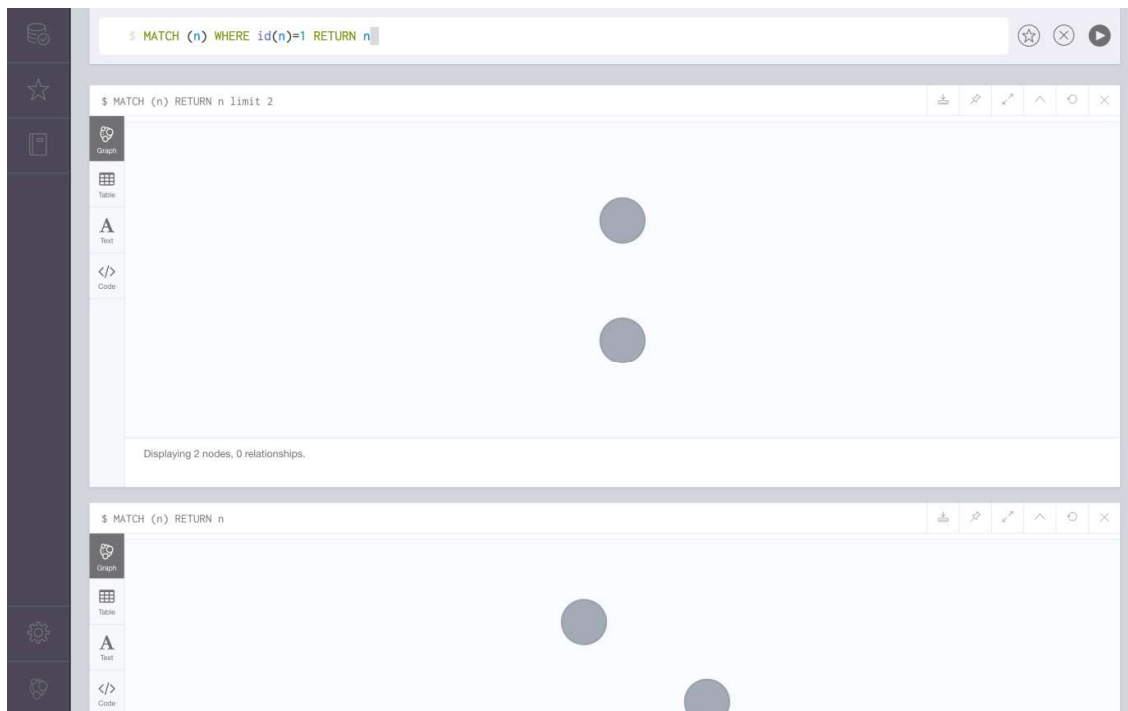
## 6) MATCH(n) RETURN n;

The screenshot shows a Neo4j Cypher console interface. The top panel displays the Cypher query `$ MATCH (n) RETURN n;`. Below the query, a message states "Created 2 nodes, completed after 1 ms." The bottom panel shows the same query `$ MATCH (n) RETURN n;` and a graph visualization with three dark grey circular nodes arranged in a triangle. A button labeled "Remove node from the visualization" is visible at the bottom of the graph area.

## 7) MATCH(n) RETURN n limit 2;



## 8) MATCH (n) WHERE id(n)=1 RETURN n;



### 9) MATCH (n) WHERE id(n)=21 RETURN n;

The screenshot shows a Cypher query interface with a sidebar on the left containing icons for Graph, Table, Text, and Code. The main area displays the query: `$ MATCH (n) WHERE id(n)=21 RETURN n`. Below the query, the result is shown in a table view, indicating "Displaying 1 nodes, 0 relationships." The interface includes a search bar at the top and a status bar at the bottom.

### 10) MATCH (n) WHERE id(n)<=6 RETURN n;

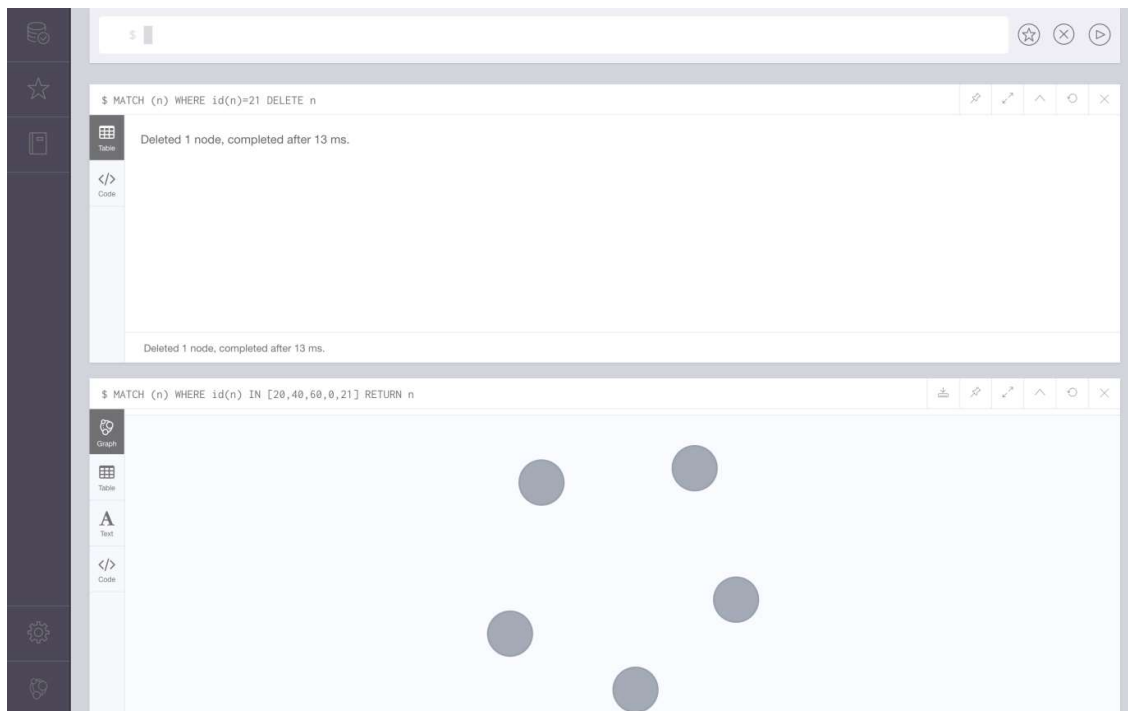
The screenshot shows a Cypher query interface with a sidebar on the left containing icons for Graph, Table, Text, and Code. The main area displays the query: `$ MATCH (n) WHERE id(n)<=6 RETURN n`. Below the query, the result is shown in a table view, indicating "Displaying 1 nodes, 0 relationships." The interface includes a search bar at the top and a status bar at the bottom.

## Dumbledore's Army

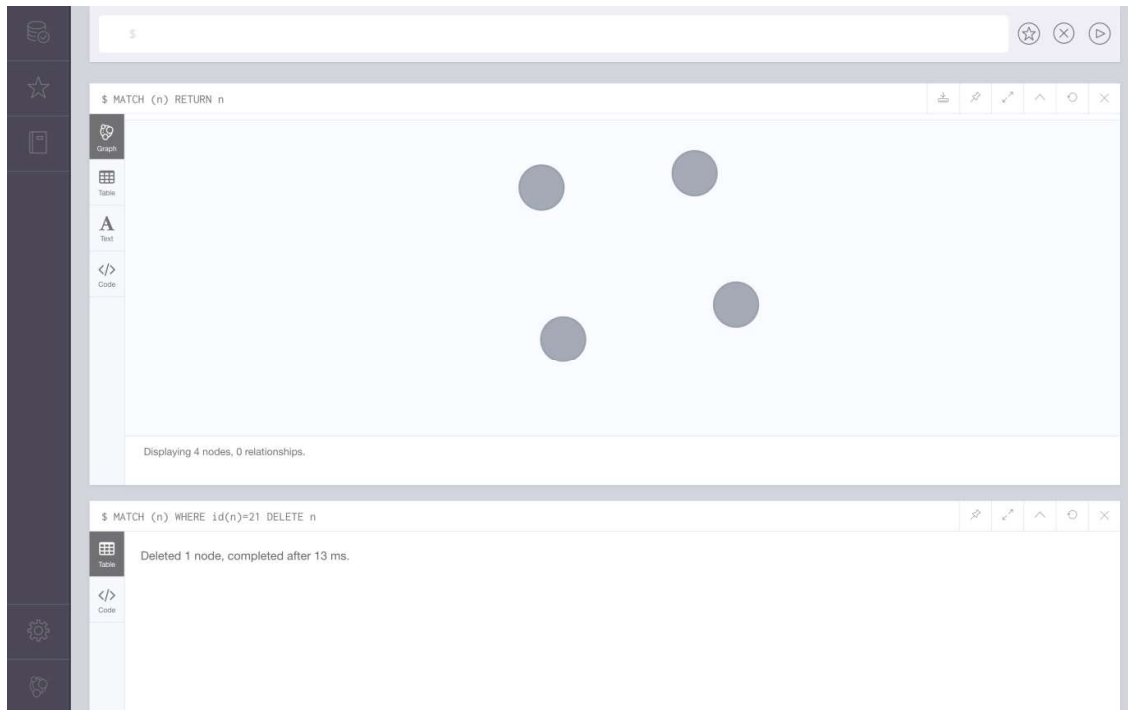
11) **MATCH (n) WHERE id(n) IN [20,40,60,0,21] RETURN n;**



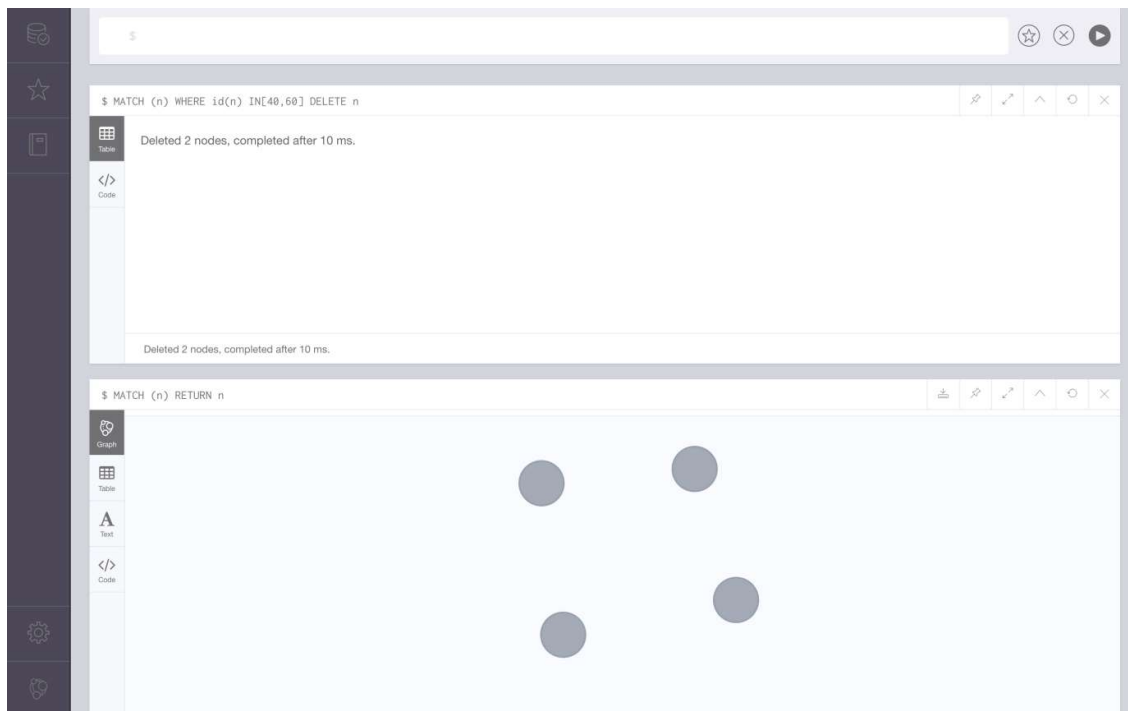
12) **MATCH (n) WHERE id(n)=21 DELETE n;**



## 13) MATCH (n) RETURN n;



## 14) MATCH (n) WHERE id(n) IN[40,60] DELETE n;





## 15) MATCH(n) DELETE n;

The screenshot shows a database interface with a sidebar on the left containing icons for a database, star, and document. The main area displays two query execution results. The top query is `$ MATCH (n) DELETE n`, which has been executed and shows a message: "Deleted 3 nodes, completed after 1 ms." The bottom query is `$ MATCH (n) RETURN n`, which has also been executed and shows a message: "Deleted 3 nodes, completed after 1 ms." The interface includes a search bar at the top and a sidebar with icons for a database, star, and document.

## 16) MATCH(n) RETURN (n)→The Database is Empty Here

The screenshot shows a database interface with a sidebar on the left containing icons for a database, star, and document. The main area displays two query execution results. The top query is `$ MATCH (n) RETURN n`, which has been executed and shows a message: "(no changes, no records)". The bottom query is `$ MATCH (n) DELETE n`, which has been executed and shows a message: "Deleted 3 nodes, completed after 1 ms." The interface includes a search bar at the top and a sidebar with icons for a database, star, and document.

## 17) CREATE (n:Keshika)

The screenshot shows a database interface with a sidebar on the left containing icons for a database, star, and document. The main area displays the execution of a Cypher query. The query bar at the top contains the statement: `$ CREATE (n:Keshika)`. Below the query bar, the results are shown in a table view. The first table has a single row with the text: "Added 1 label, created 1 node, completed after 26 ms." The second table is empty and has the text: "(no changes, no records)". The third table has a single row with the text: "Completed in less than 1 ms." The query bar at the bottom contains the statement: `$ MATCH (n) DELETE n`.

## 18) MATCH(n) WHERE n:Keshika RETURN n;

The screenshot shows a database interface with a sidebar on the left containing icons for a database, star, and document. The main area displays the execution of a Cypher query. The query bar at the top contains the statement: `$ MATCH (n) WHERE n:Keshika RETURN n`. Below the query bar, the results are shown in a graph view. A single node is displayed, labeled "Keshika(1)". The node is a blue circle with the number "1" inside. The text "Displaying 1 nodes, 0 relationships." is shown below the graph. The query bar at the bottom contains the statement: `$ CREATE (n:Keshika)`.

## 19) MATCH(n) WHERE n:Keshika RETURN n;

The screenshot shows a Cypher query interface with the following components:

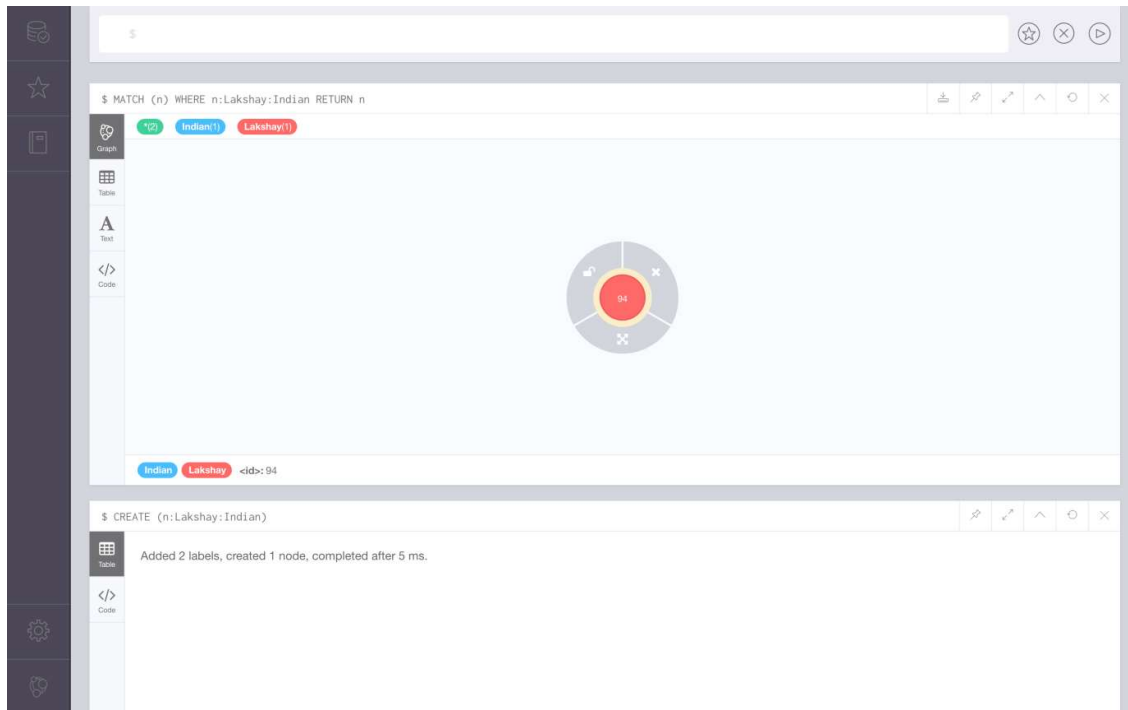
- Query Bar:** Contains the query `$ MATCH (n) WHERE n:Keshika RETURN n`.
- Execution Panel:** Shows the query result as a single node labeled `Keshika` with ID `21`. The node is represented by a blue circle with the number 21 inside.
- Left Sidebar:** Contains icons for Graph, Table, Text, and Code.
- Bottom Panel:** Shows the query execution status: `$ CREATE (n:Keshika)` and the message "Added 1 label, created 1 node, completed after 26 ms."

## 20) CREATE(n:Lakshay:Indian)

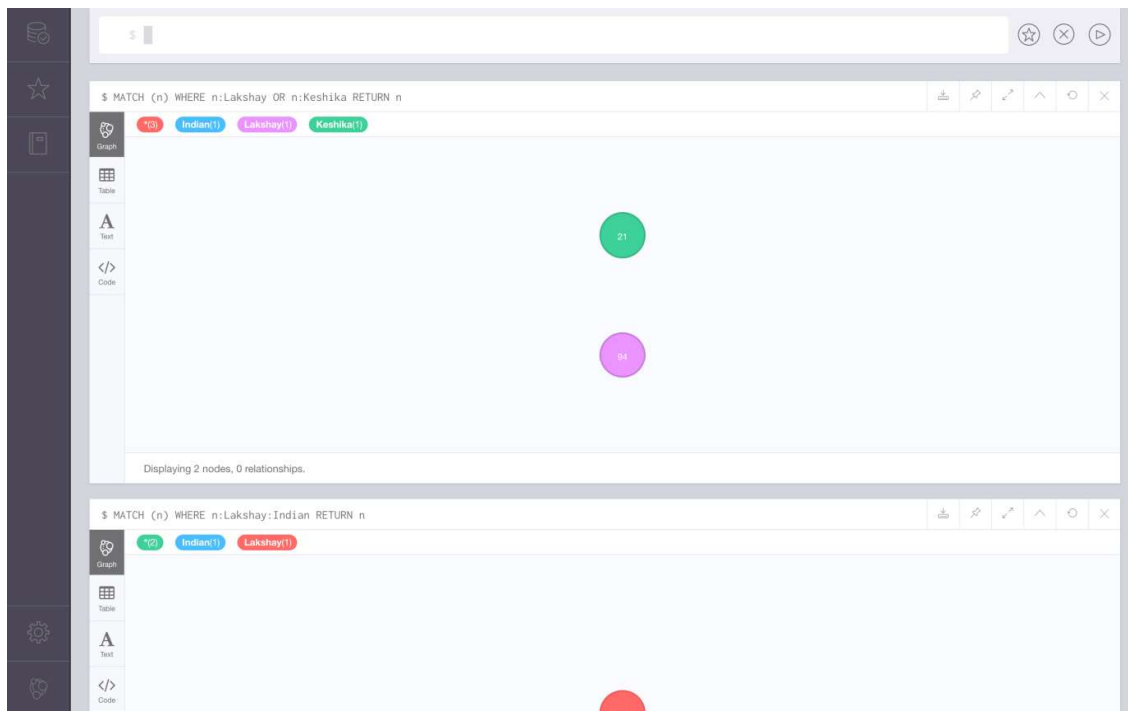
The screenshot shows a Cypher query interface with the following components:

- Query Bar:** Contains the query `$ CREATE (n:Lakshay:Indian)`.
- Execution Panel:** Shows the query result as a single node labeled `Lakshay` with ID `21`. The node is represented by a blue circle with the number 21 inside.
- Left Sidebar:** Contains icons for Graph, Table, Text, and Code.
- Bottom Panel:** Shows the query execution status: `$ CREATE (n:Lakshay:Indian)` and the message "Added 2 labels, created 1 node, completed after 5 ms."

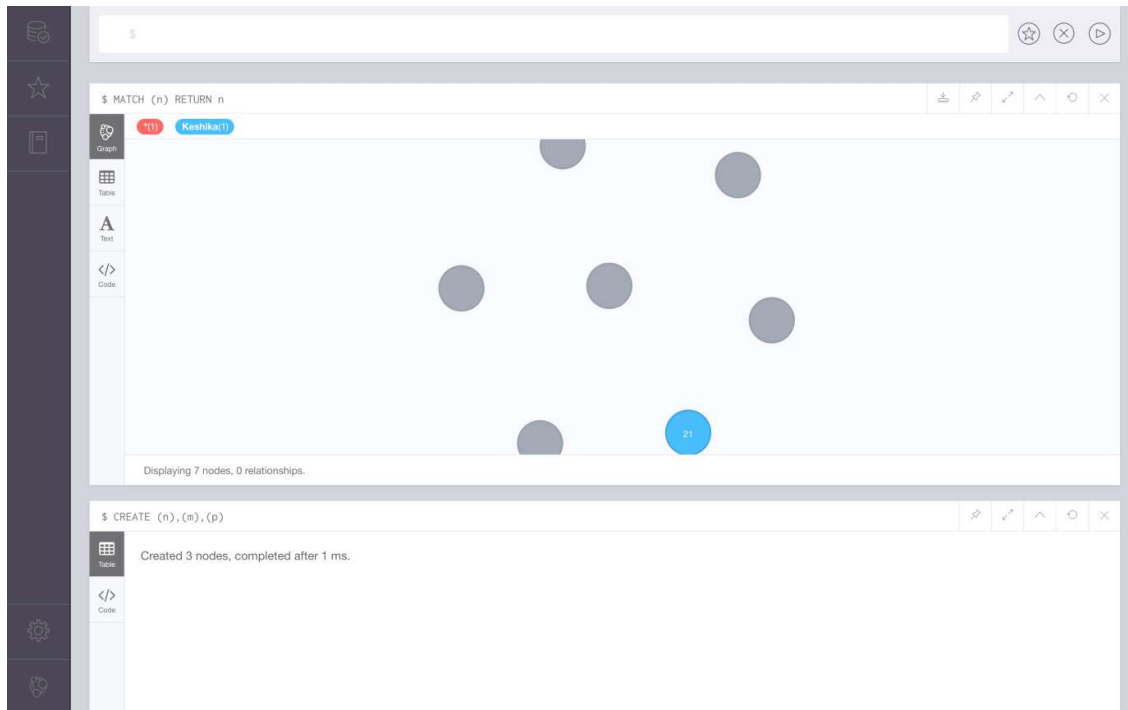
## 21) MATCH (n) WHERE n:Lakshay:Indian RETURN n;



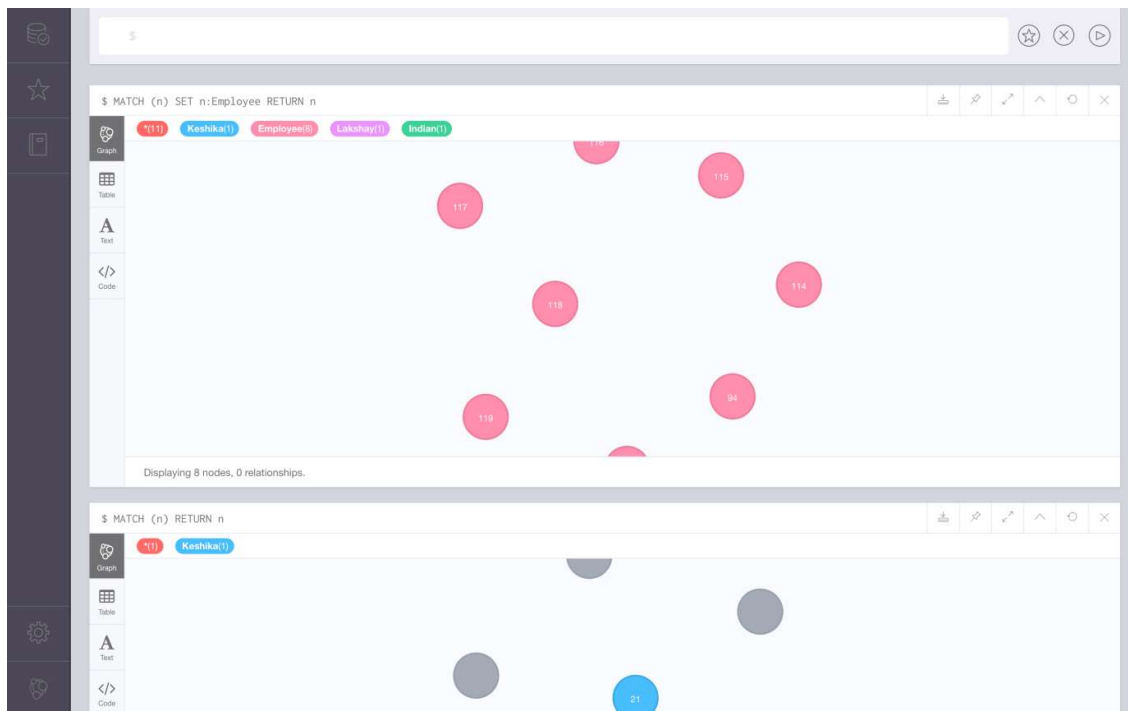
## 22) MATCH(n) WHERE n:Lakshay OR n:Keshika RETURN n;



## 23) MATCH (n) RETURN n;

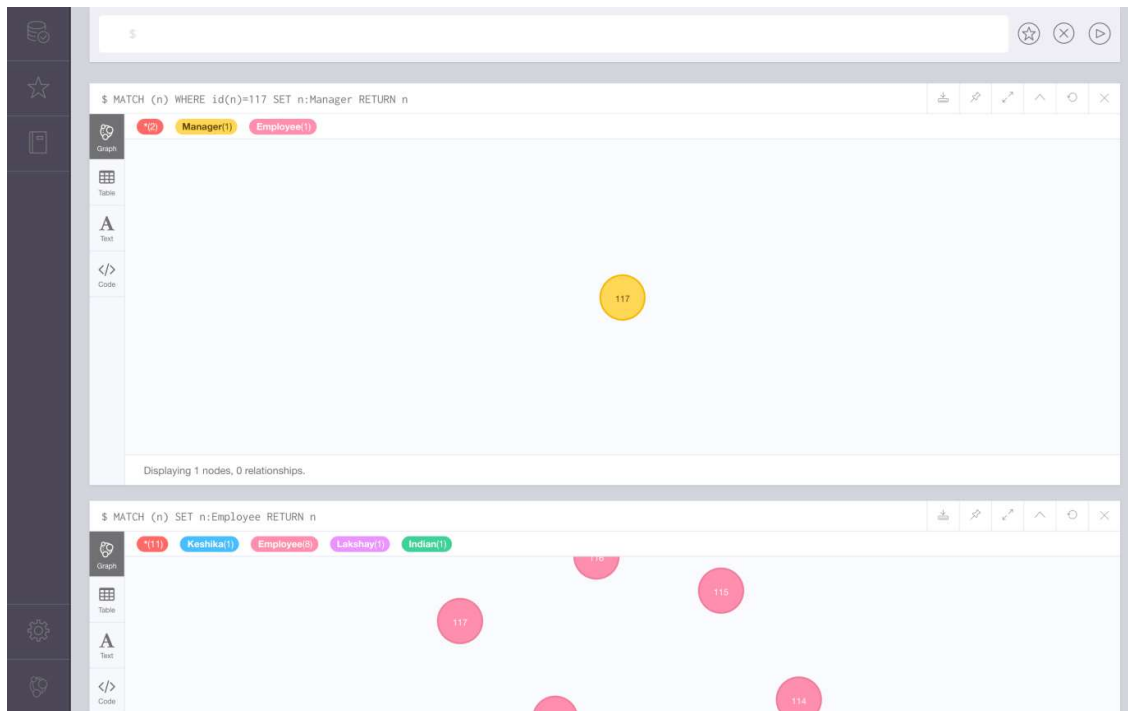


## 24) MATCH (n) SET n:Employee RETURN n;

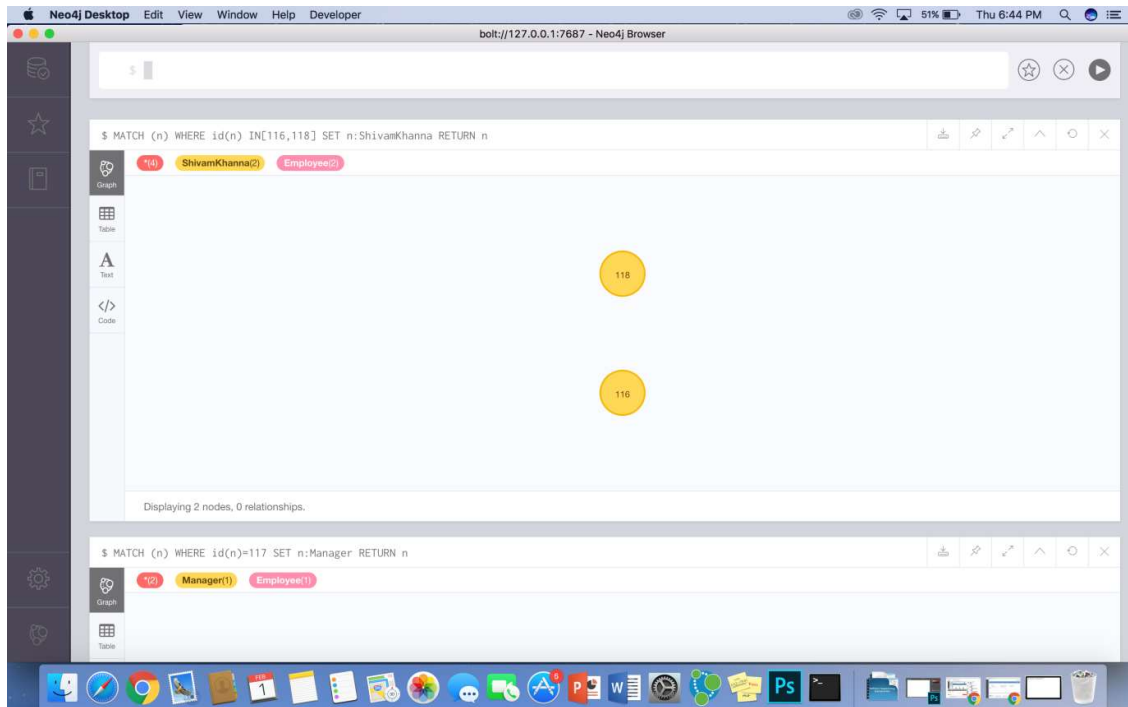


## Dumbledore's Army

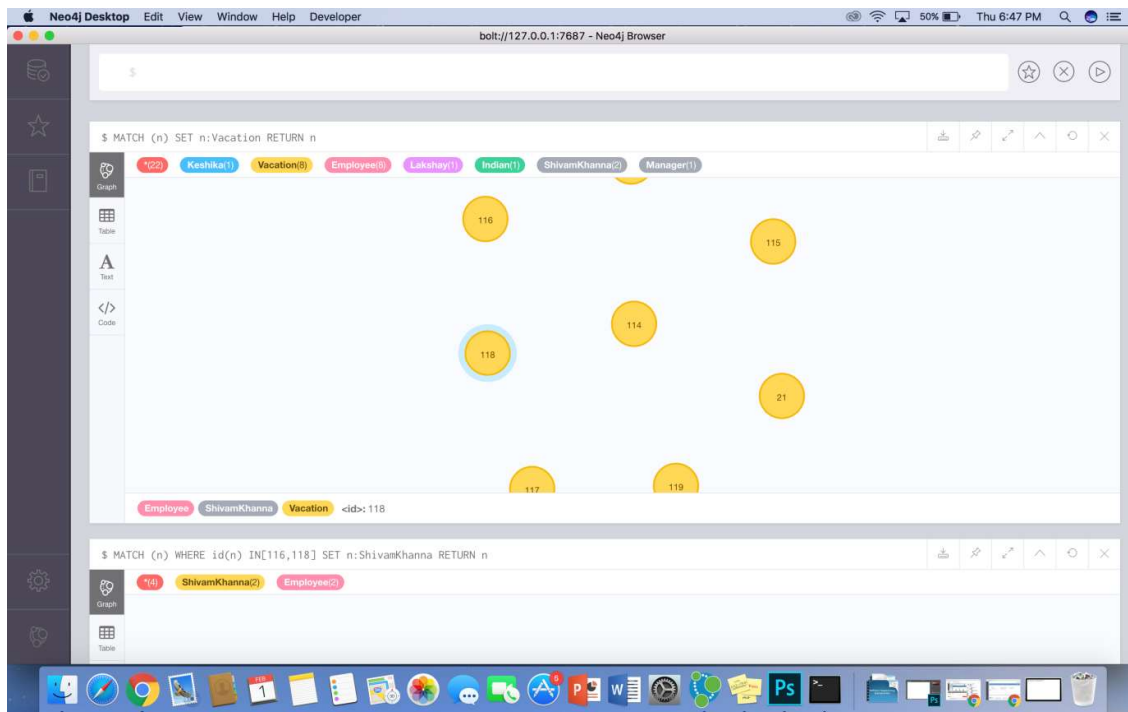
25) **MATCH(n) WHERE id(n)=117 SET n:Manager RETURN n;**



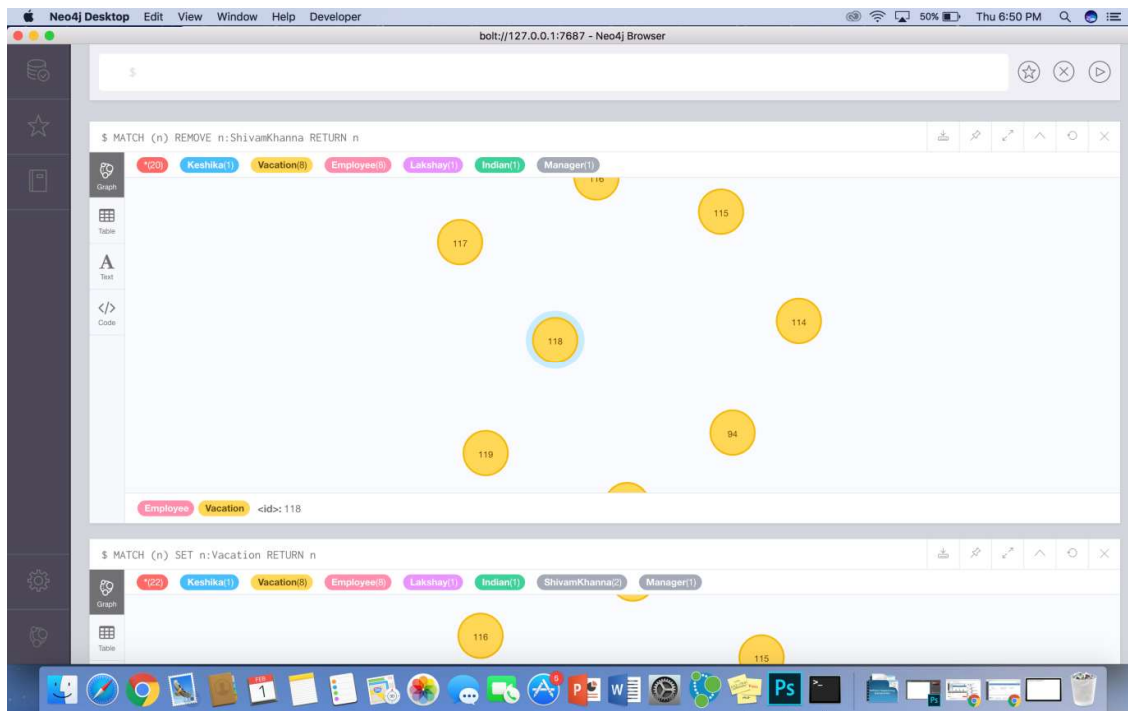
26) **MATCH (n) WHERE id(n) IN[116,118] SET n:ShivamKhanna RETURN n;**



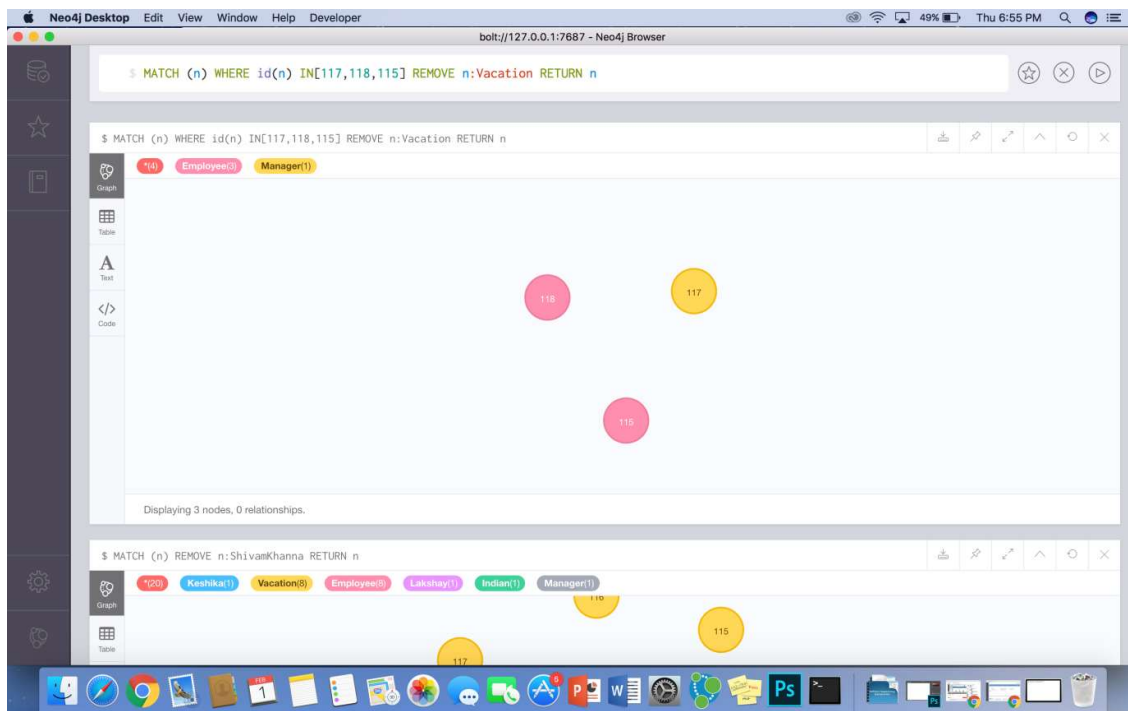
## 27) MATCH (n) SET n:Vacation RETURN n;



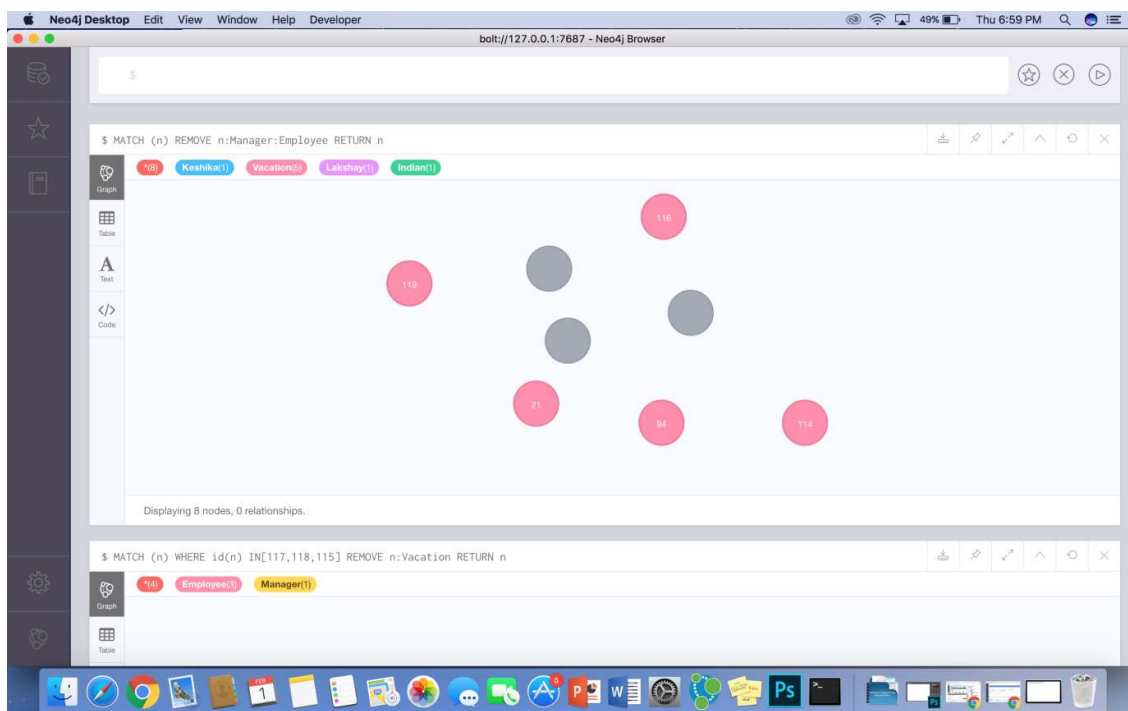
## 28) MATCH (n) REMOVE n:ShivamKhanna RETURN n;



### 29) MATCH(n) WHERE id(n) In[117,118,115] REMOVE n:Vacation RETURN n;

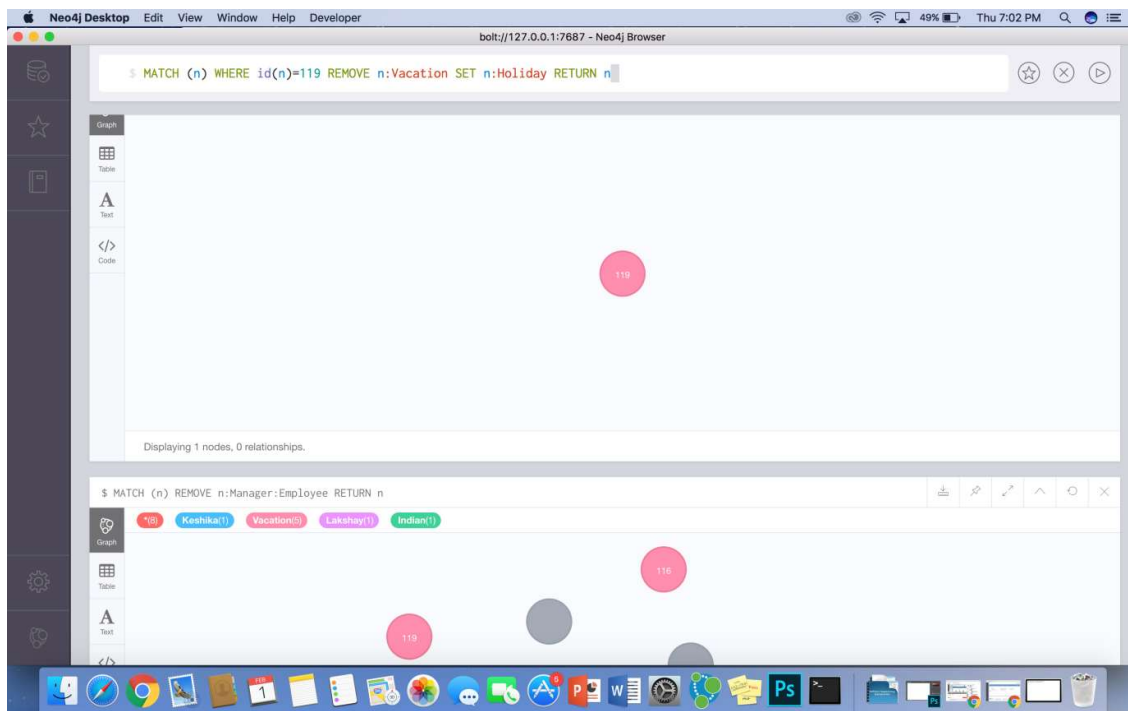


### 30) MATCH(n) REMOVE n:Manager:Employee RETURN n;

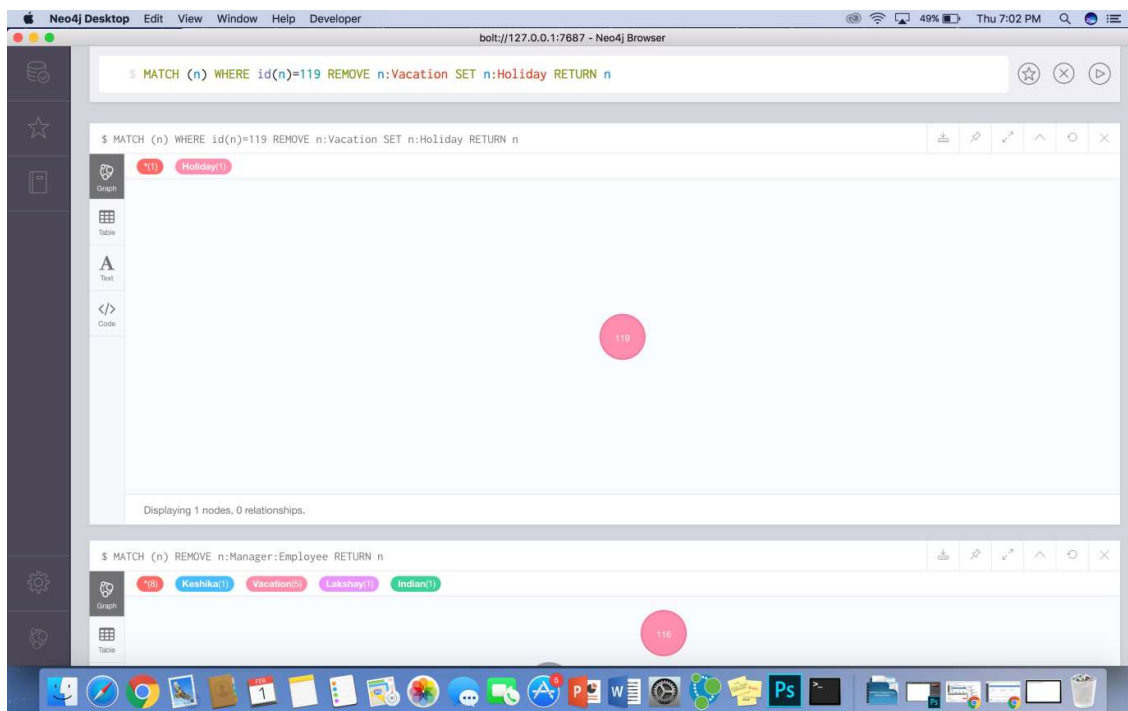




**31) MATCH(n) WHERE id(n)=119 REMOVE n:Vacation SET n:Holiday RETURN n;  
SET n:Holiday RETURN n;**



**32) MATCH(n) WHERE id(n)=119 REMOVE n:Vacation SET n:Holiday RETURN n;  
n;**



## 33) MATCH(n) RETURN DISTINCT labels(n);

The screenshot shows the Neo4j Desktop interface with the following details:

- Query:** `$ MATCH (n) RETURN DISTINCT labels(n);`
- Results Table:**

| labels(n)                         |
|-----------------------------------|
| ["Keshika", "Vacation"]           |
| ["Lakshay", "Indian", "Vacation"] |
| ["Vacation"]                      |
| []                                |
| ["Holiday"]                       |
- Execution Info:** Started streaming 5 records after 4 ms and completed after 17 ms.
- Graph View:** A graph visualization showing a single node with ID 119, highlighted in pink.

## 34) MATCH(n) WHERE id(n)=94 RETURN labels(n);

The screenshot shows the Neo4j Desktop interface with the following details:

- Query:** `$ MATCH (n) WHERE id(n)=94 RETURN labels(n);`
- Results Table:**

| labels(n)                         |
|-----------------------------------|
| ["Lakshay", "Indian", "Vacation"] |
- Execution Info:** Started streaming 1 records after 1 ms and completed after 1 ms.
- Graph View:** A graph visualization showing a single node with ID 94, highlighted in pink.

## 35) MATCH (n) RETURN DISTINCT count(labels(n));

The screenshot shows the Neo4j Desktop interface with a Cypher query editor and results pane. The query is `$ MATCH (n) RETURN DISTINCT count(labels(n));`. The results pane displays a table with one column, `count(labels(n))`, and one row with the value `8`. Below the table, it states "Started streaming 1 records after 5 ms and completed after 5 ms." The bottom of the screen shows a macOS dock with various application icons.

| count(labels(n)) |
|------------------|
| 8                |

## 36) MATCH(n) RETURN DISTINCT count(labels(n)),labels(n);

The screenshot shows the Neo4j Desktop interface with two Cypher queries and their results. The first query is `$ MATCH (n) RETURN DISTINCT count(labels(n)),labels(n);`. The results pane displays a table with two columns, `count(labels(n))` and `labels(n)`, and five rows of data. Below the table, it states "Started streaming 5 records after 6 ms and completed after 6 ms." The second query is `$ MATCH (n) RETURN DISTINCT count(labels(n));`. The results pane displays a table with one column, `count(labels(n))`, and one row with the value `8`. The bottom of the screen shows a macOS dock with various application icons.

| count(labels(n)) | labels(n)                         |
|------------------|-----------------------------------|
| 2                | ["Vacation"]                      |
| 3                | []                                |
| 1                | ["Lakshay", "Indian", "Vacation"] |
| 1                | ["Holiday"]                       |
| 1                | ["Keshika", "Vacation"]           |

| count(labels(n)) |
|------------------|
| 8                |

## 37) MATCH (n) WHERE n:Holiday DELETE n;

The screenshot shows the Neo4j Desktop interface with the following details:

- Query:** `$ MATCH (n) WHERE n:Holiday DELETE n;`
- Execution Status:** Deleted 1 node, completed after 3 ms.
- Table View:** A table with two columns: `count(labels(n))` and `labels(n)`.

| count(labels(n)) | labels(n)                         |
|------------------|-----------------------------------|
| 2                | ["Vacation"]                      |
| 3                | []                                |
| 1                | ["Lakshay", "Indian", "Vacation"] |
| 1                | ["Holiday"]                       |
| 1                | ["Keshika", "Vacation"]           |

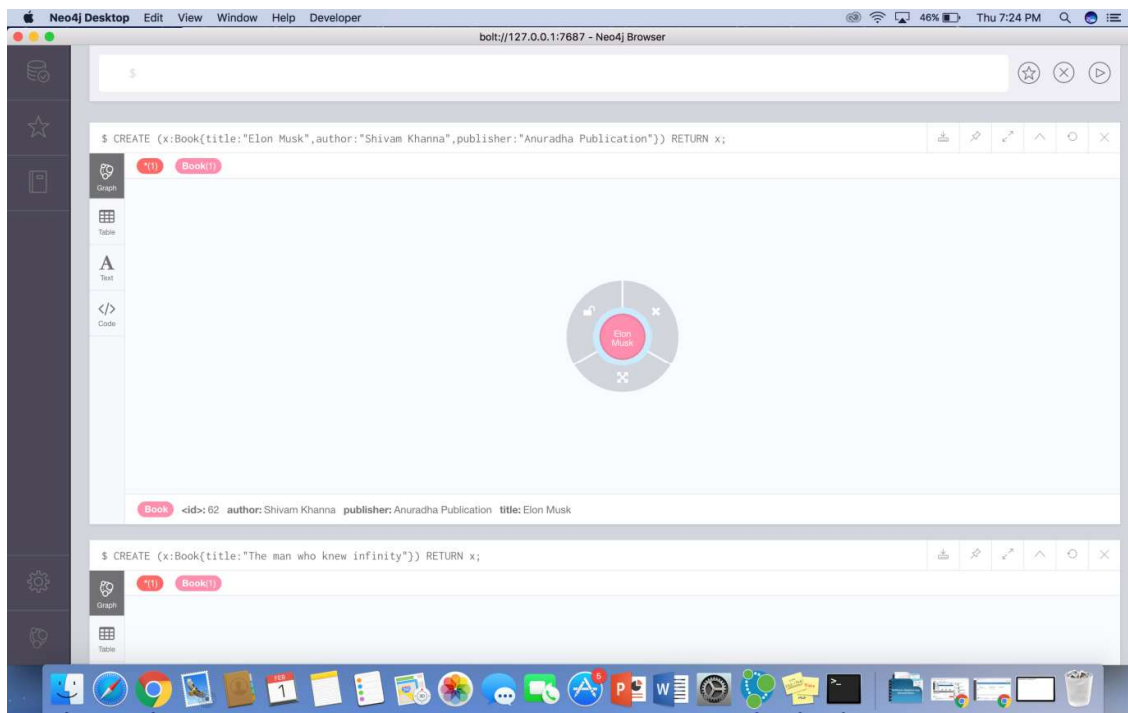
## 38) CREATE (x:Book{title:"The man who knew infinity"}) RETURN x;

The screenshot shows the Neo4j Desktop interface with the following details:

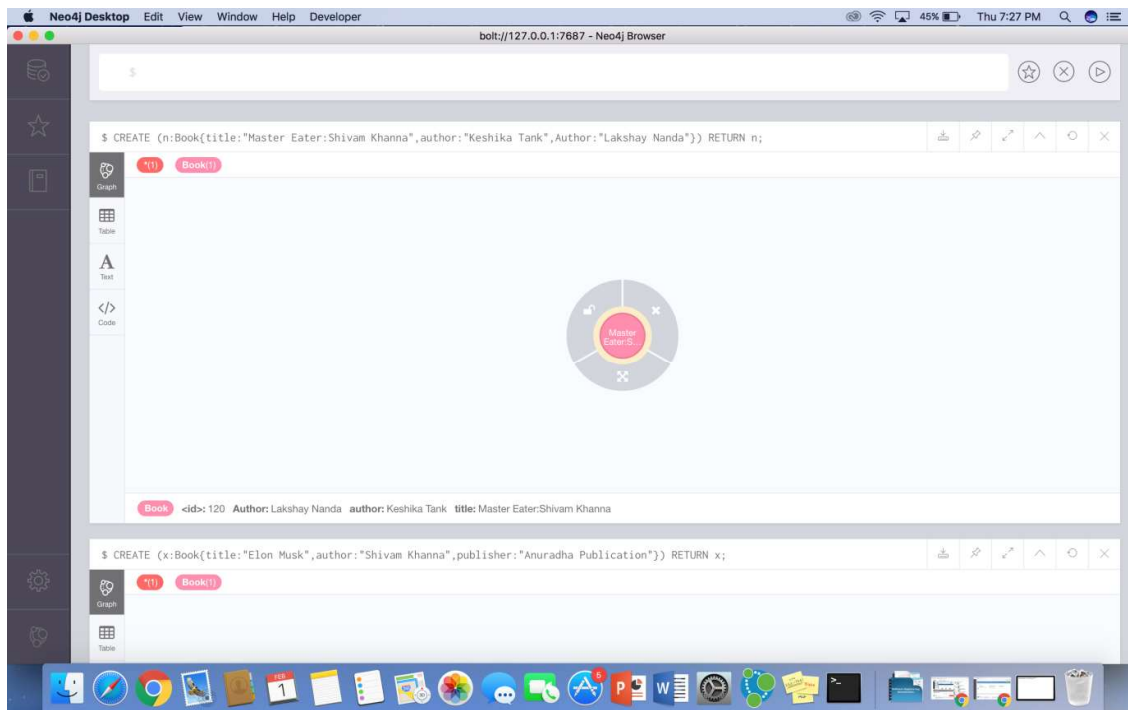
- Query:** `$ CREATE (x:Book{title:"The man who knew infinity"}) RETURN x;`
- Execution Status:** Displaying 1 nodes, 0 relationships.
- Graph View:** A single pink node labeled "The man who knew infinity".
- Error Message:** `ERROR Neo.ClientError.Statement.SyntaxError`  
Variable 'n' not defined (Line 1, column 59 (offset: 58))  
"CREATE (x:Book{title:"The man who knew infinity"}) RETURN n;"

## Dumbledore's Army

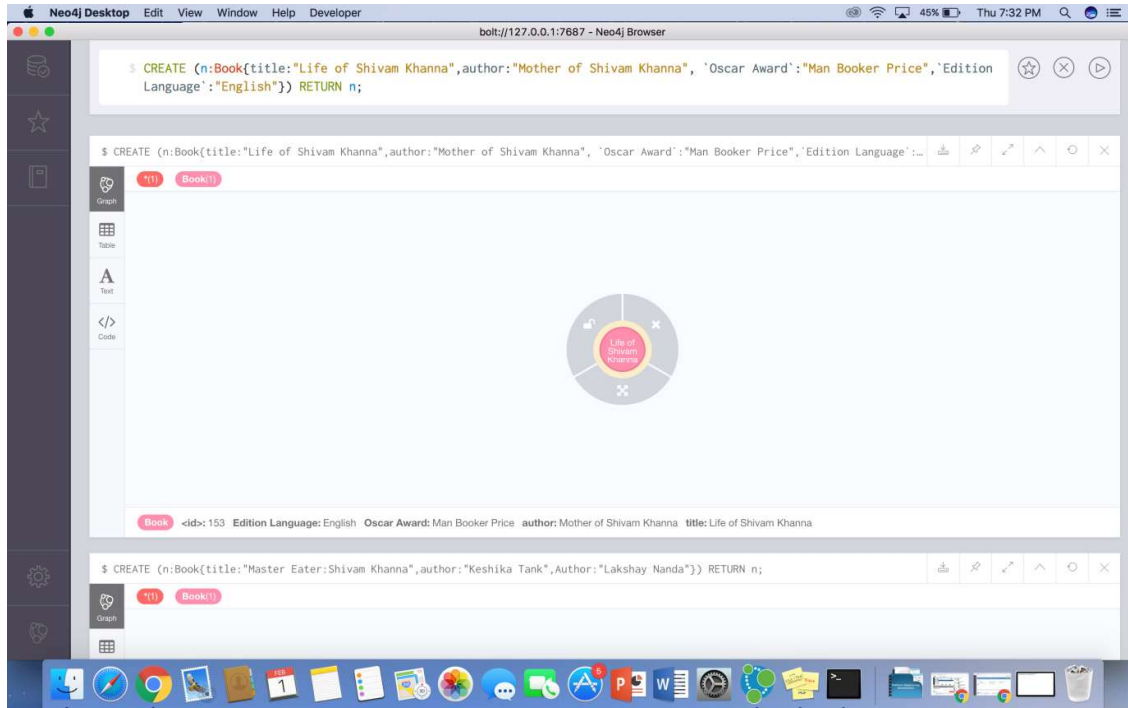
39) **CREATE (x:Book(title:"Elon Musk", author:"Shivam Khanna",publisher:"Anuradha Publications")) RETURN x;**



40) **CREATE (n:Book(title:"Master Eater: Shivam Khanna", author:"Keshika Tank", Author:"Lakshay Nanda")) RETURN n**



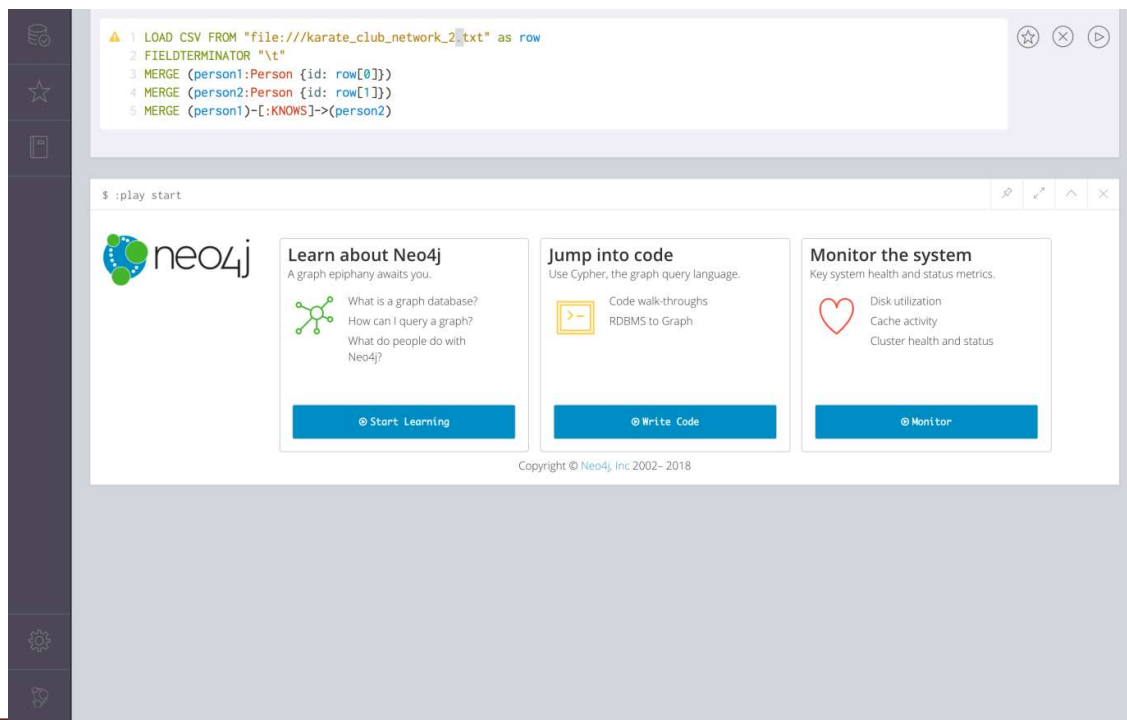
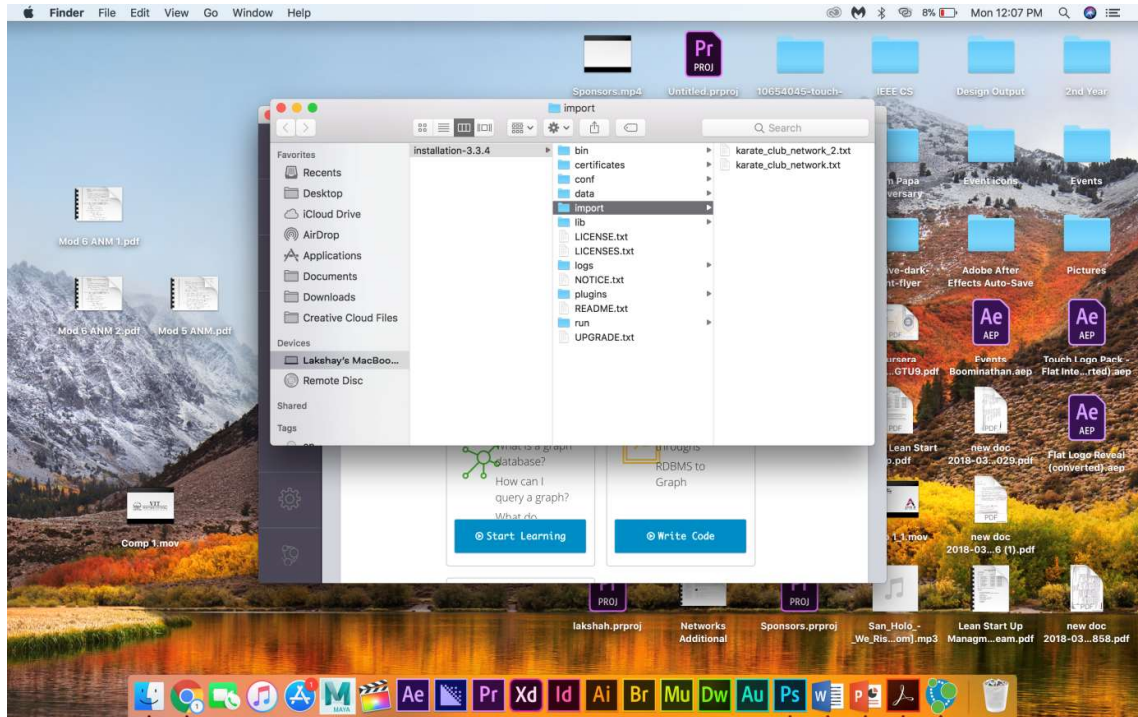
41) **CREATE(n:Book(title:"Life of Shivam Khanna",  
author:"Mother of Shivam Khanna",`Oscar Award`:"Man  
Booker Price", `Edition Language`:"English")) RETURN n;**



# Dumbledore's Army

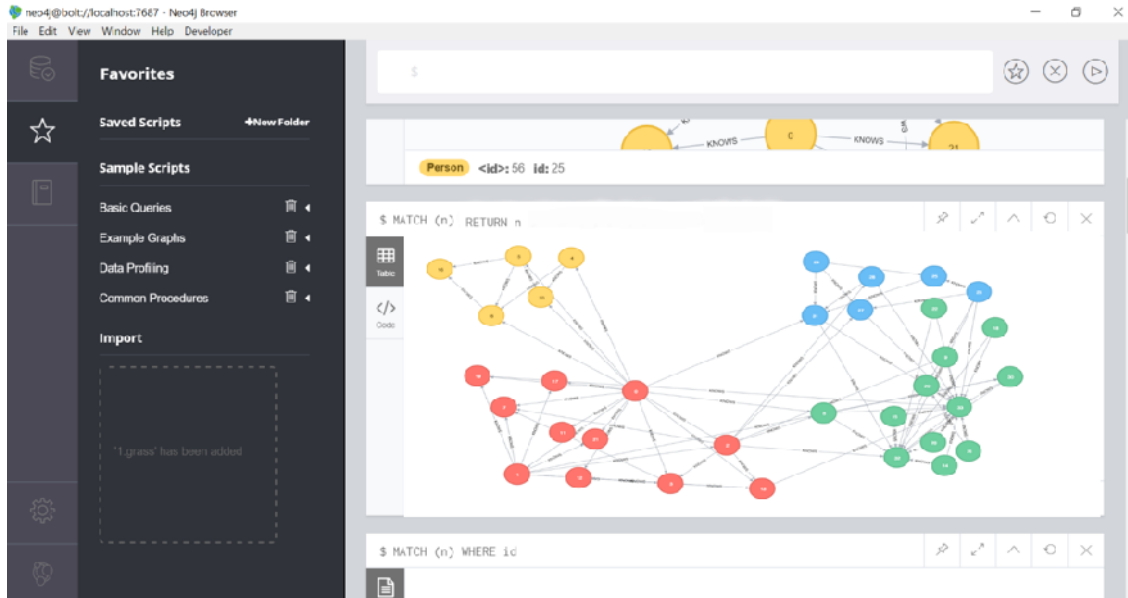
## Tasks:-

### 1. SLM clustering

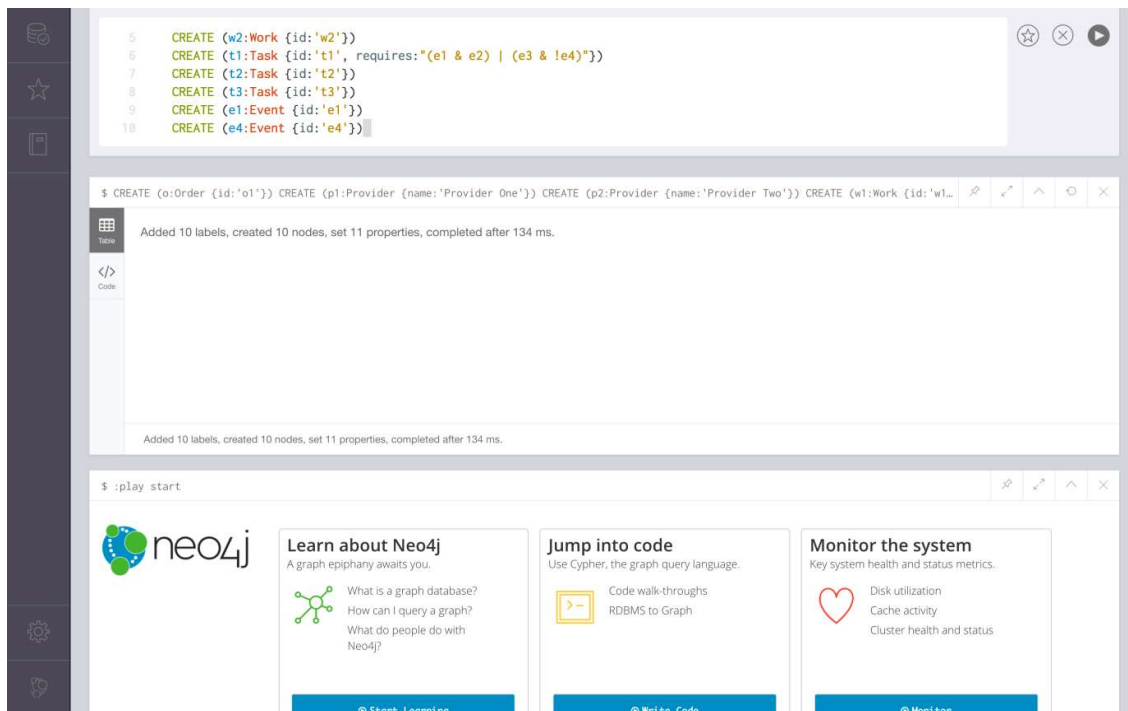




# Dumbledore's Army



## 2. Work order management





# Dumbledore's Army

The image shows the Neo4j Desktop application interface. On the left is a dark sidebar with icons for home, favorites, recent, and settings. The main area is divided into three sections:

- Query Editor:** Contains a Cypher query:

```
5 CREATE (w2:Work {id:'w2'})
6 CREATE (t1:Task {id:'t1', requires:'(e1 & e2) | (e3 & !e4)'})
7 CREATE (t2:Task {id:'t2'})
8 CREATE (t3:Task {id:'t3'})
9 CREATE (e1:Event {id:'e1'})
10 CREATE (e4:Event {id:'e4'})
```
- Table View:** Shows a table with 10 labels, 10 nodes, and 11 properties. The status bar indicates "Added 10 labels, created 10 nodes, set 11 properties, completed after 134 ms."
- Dashboard:** Features the Neo4j logo and three main sections:
  - Learn about Neo4j:** A graph epiphany awaits you. Includes links for "What is a graph database?", "How can I query a graph?", and "What do people do with Neo4j?".
  - Jump into code:** Use Cypher, the graph query language. Includes links for "Code walk-throughs" and "RDBMS to Graph".
  - Monitor the system:** Key system health and status metrics. Includes links for "Disk utilization", "Cache activity", and "Cluster health and status".

**THANK YOU!!**