Project Review

NoSQL Database

Neo4j

DUMBLEDORE'S ARMY

Team Members:

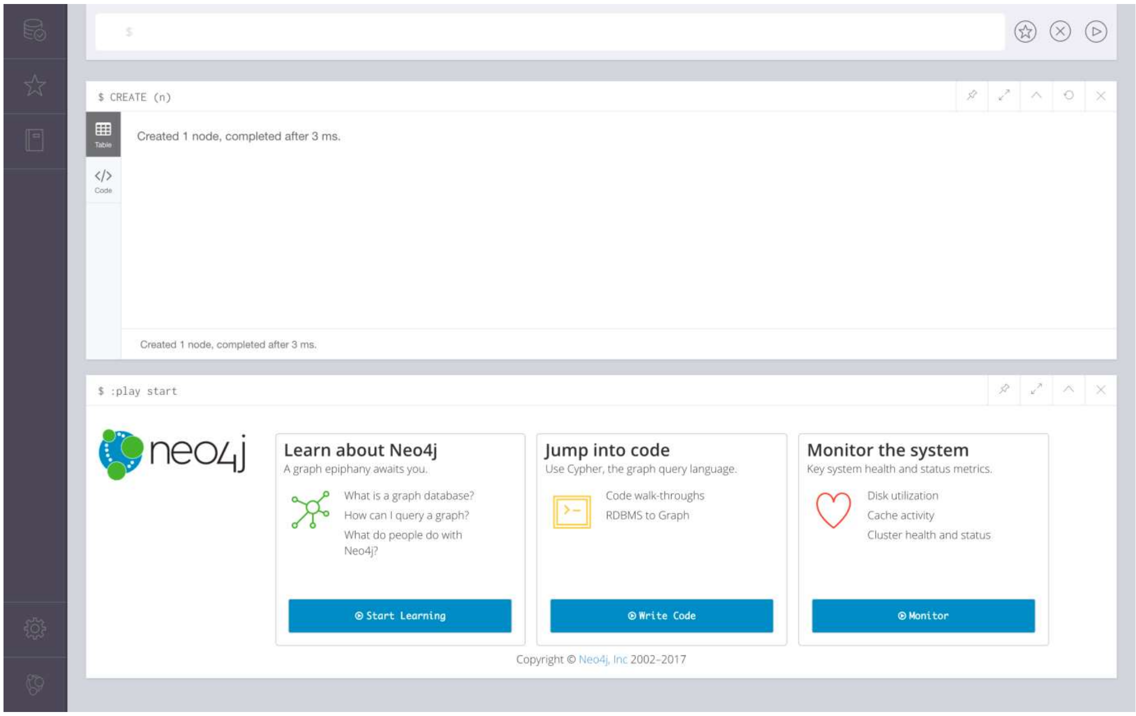
LAKSHAY NANDA (16BCE0805)

KESHIKA TANK (16BCE0796)

SHIVAM KHANNA (16BCI0106)

SNIPPETS OF THE CQL QUERY AND THE OUTPUT

1) CREATE(n)



```
$ CREATE (n)
```

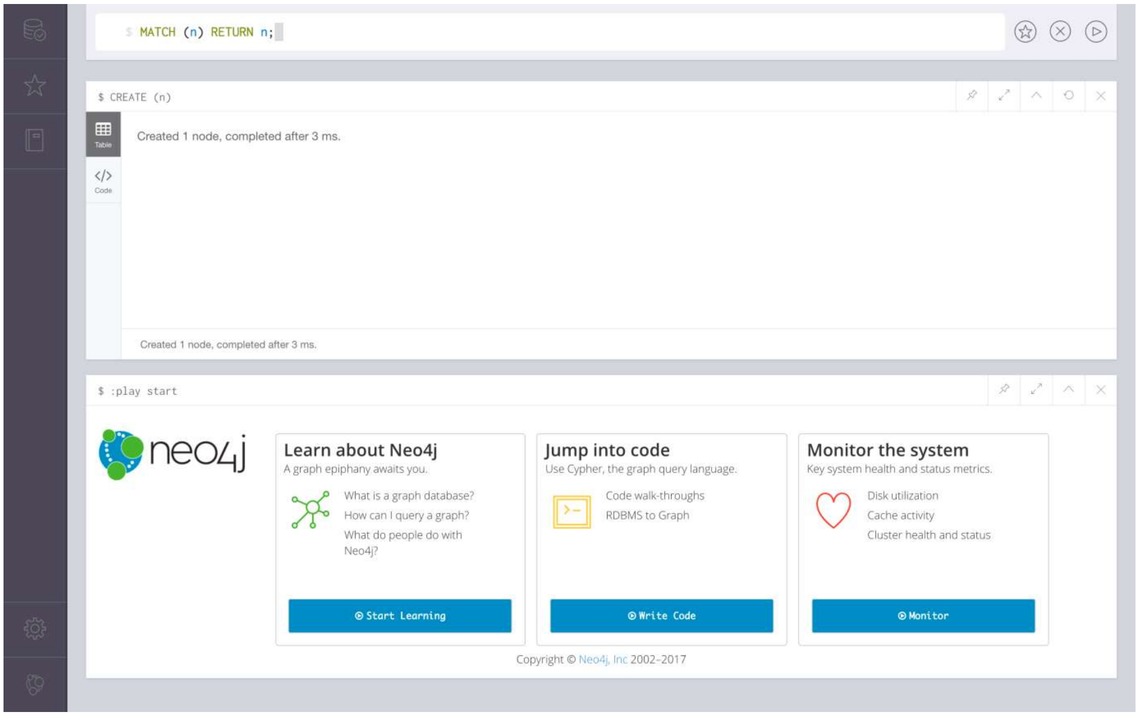
Created 1 node, completed after 3 ms.

Created 1 node, completed after 3 ms.

```
$ :play start
```

Table
Code
neo4j
Learn about Neo4j
A graph epiphany awaits you.
What is a graph database?
How can I query a graph?
What do people do with Neo4j?
Start Learning
Jump into code
Use Cypher, the graph query language.
Code walk-throughs
RDBMS to Graph
Write Code
Monitor the system
Key system health and status metrics.
Disk utilization
Cache activity
Cluster health and status
Monitor
Copyright © Neo4j, Inc 2002-2017

2) MATCH(n) RETURN n;



```
$ MATCH (n) RETURN n;
```

```
$ CREATE (n)
```

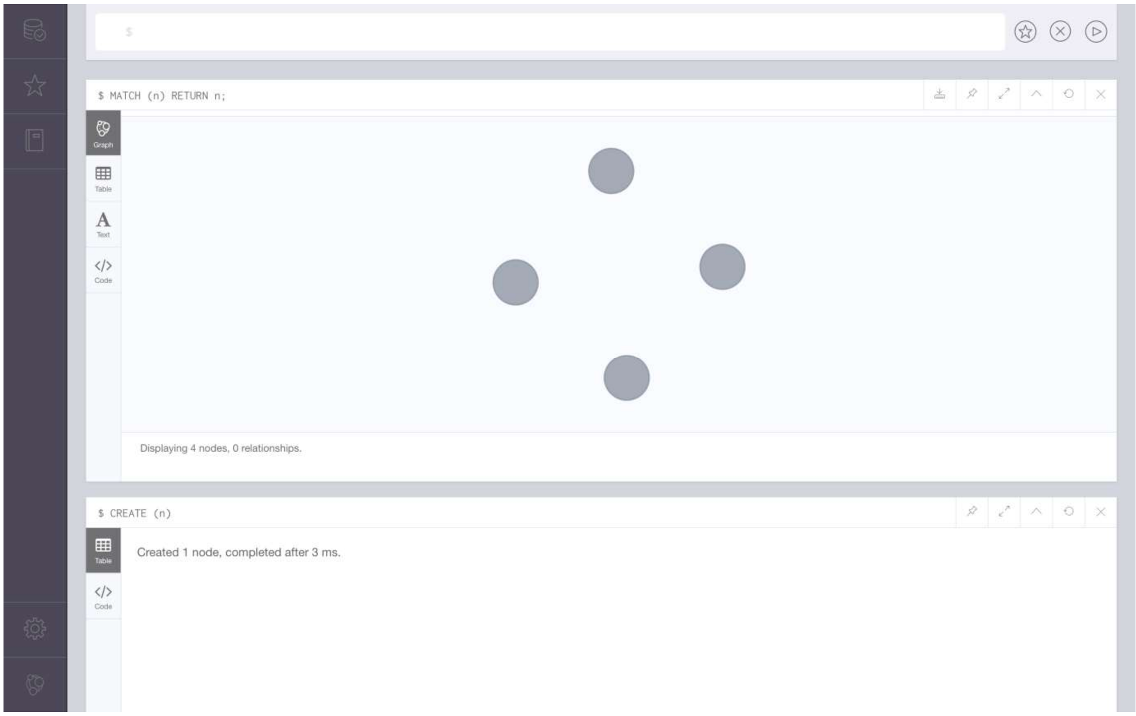
Created 1 node, completed after 3 ms.

Created 1 node, completed after 3 ms.

```
$ :play start
```

Table
Code
neo4j
Learn about Neo4j
A graph epiphany awaits you.
What is a graph database?
How can I query a graph?
What do people do with Neo4j?
Start Learning
Jump into code
Use Cypher, the graph query language.
Code walk-throughs
RDBMS to Graph
Write Code
Monitor the system
Key system health and status metrics.
Disk utilization
Cache activity
Cluster health and status
Monitor
Copyright © Neo4j, Inc 2002-2017

3) MATCH(n) RETURN n;



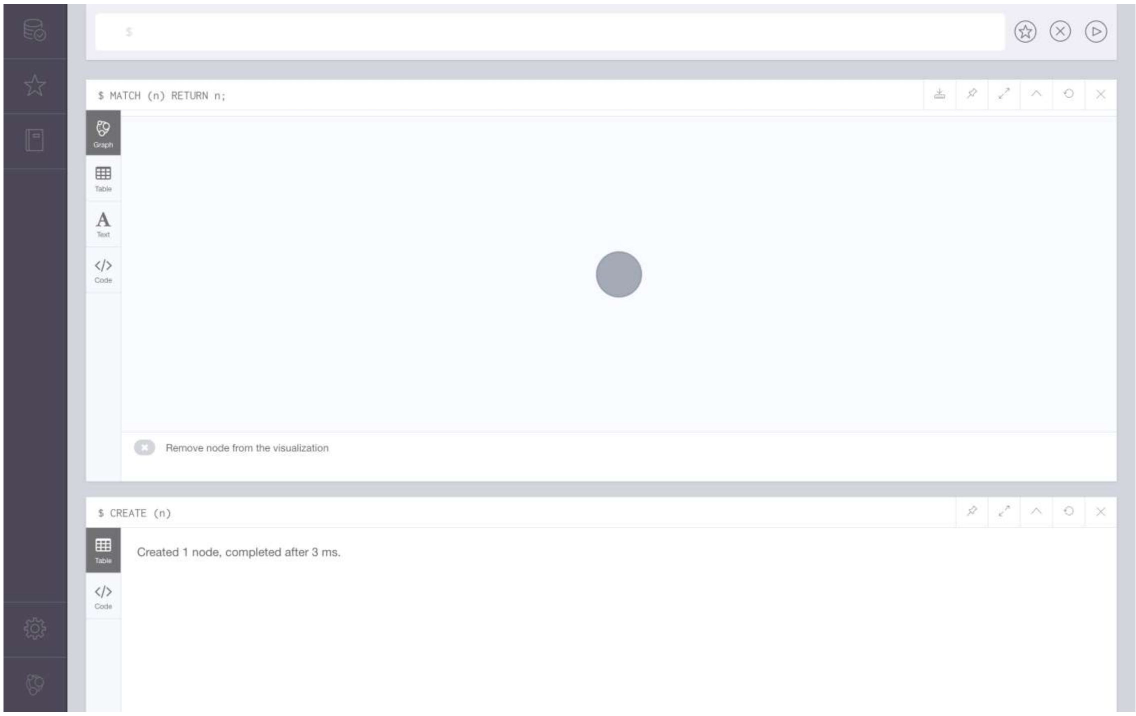
```
$ MATCH (n) RETURN n;
```

Displaying 4 nodes, 0 relationships.

```
$ CREATE (n)
```

Created 1 node, completed after 3 ms.

4) Match(n) REMOVE(n);



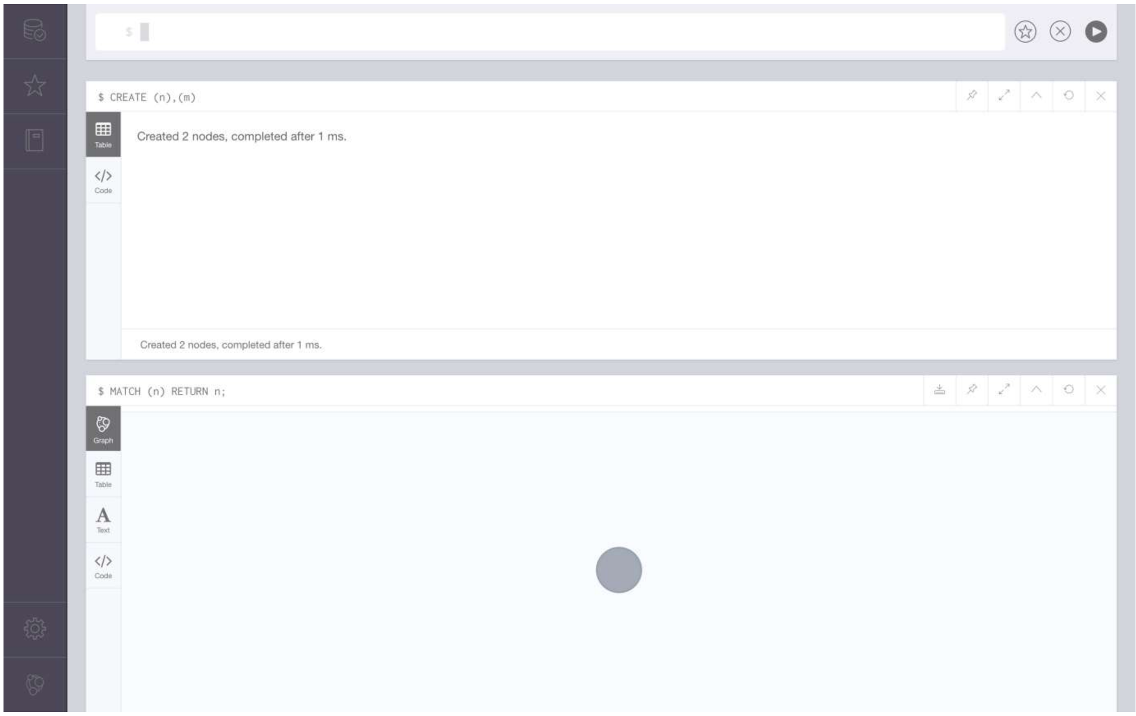
```
$ MATCH (n) RETURN n;
```

Remove node from the visualization

```
$ CREATE (n)
```

Created 1 node, completed after 3 ms.

5) CREATE (n),(m);



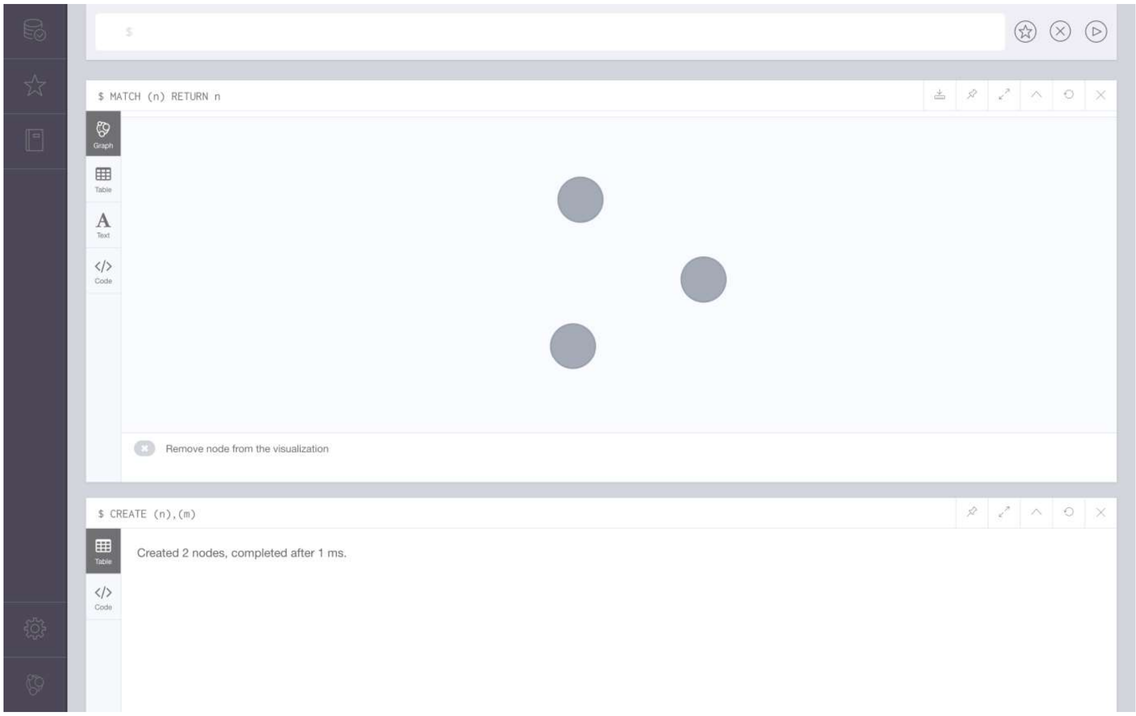
```
$ CREATE (n),(m)
```

Created 2 nodes, completed after 1 ms.

Created 2 nodes, completed after 1 ms.

```
$ MATCH (n) RETURN n;
```

6) MATCH(n) RETURN n;



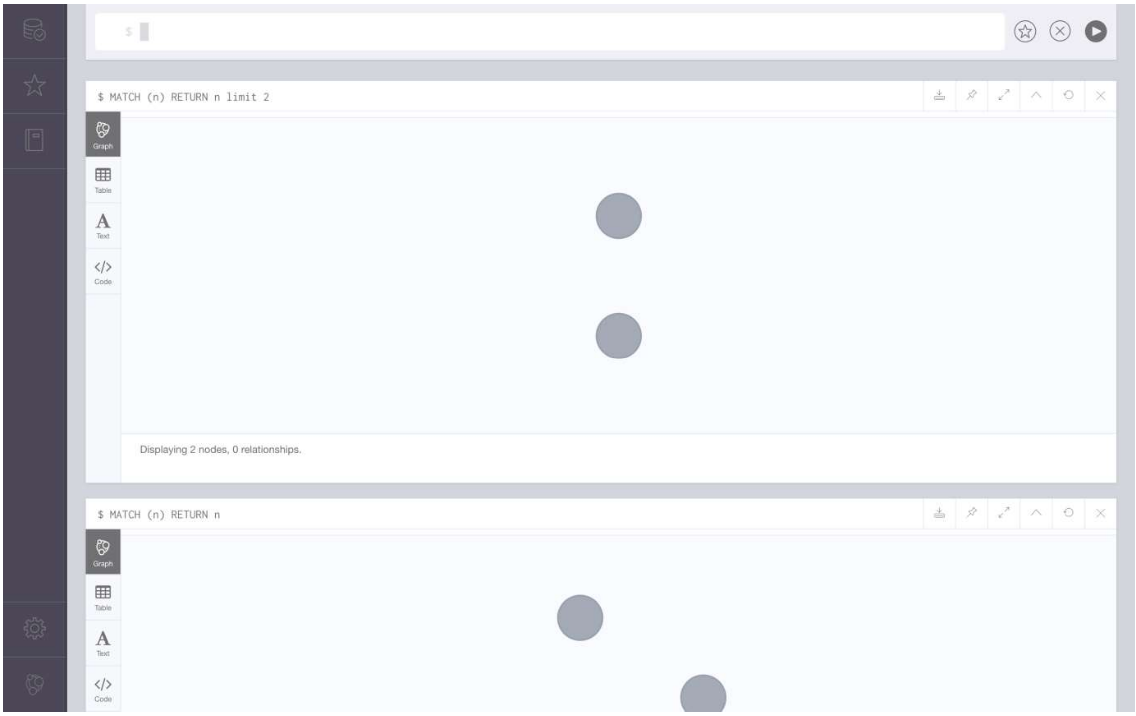
```
$ MATCH (n) RETURN n
```

Remove node from the visualization

```
$ CREATE (n),(m)
```

Created 2 nodes, completed after 1 ms.

7) MATCH(n) RETURN n limit 2;

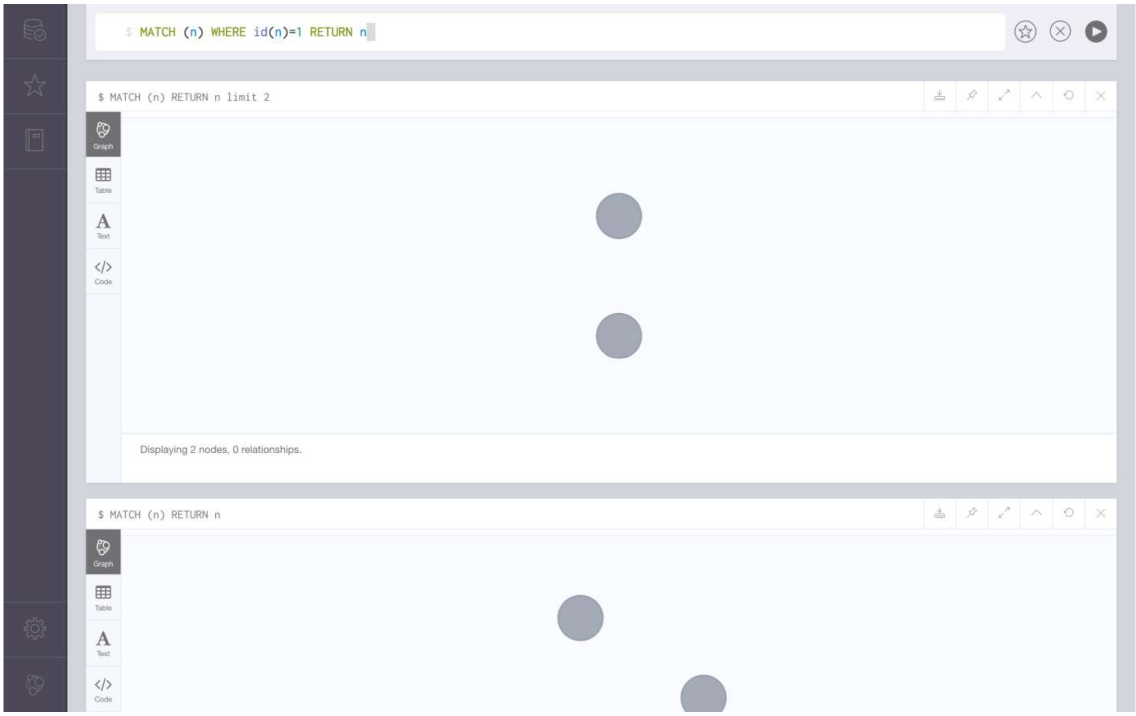


```
$ MATCH (n) RETURN n limit 2
```

Displaying 2 nodes, 0 relationships.

```
$ MATCH (n) RETURN n
```

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



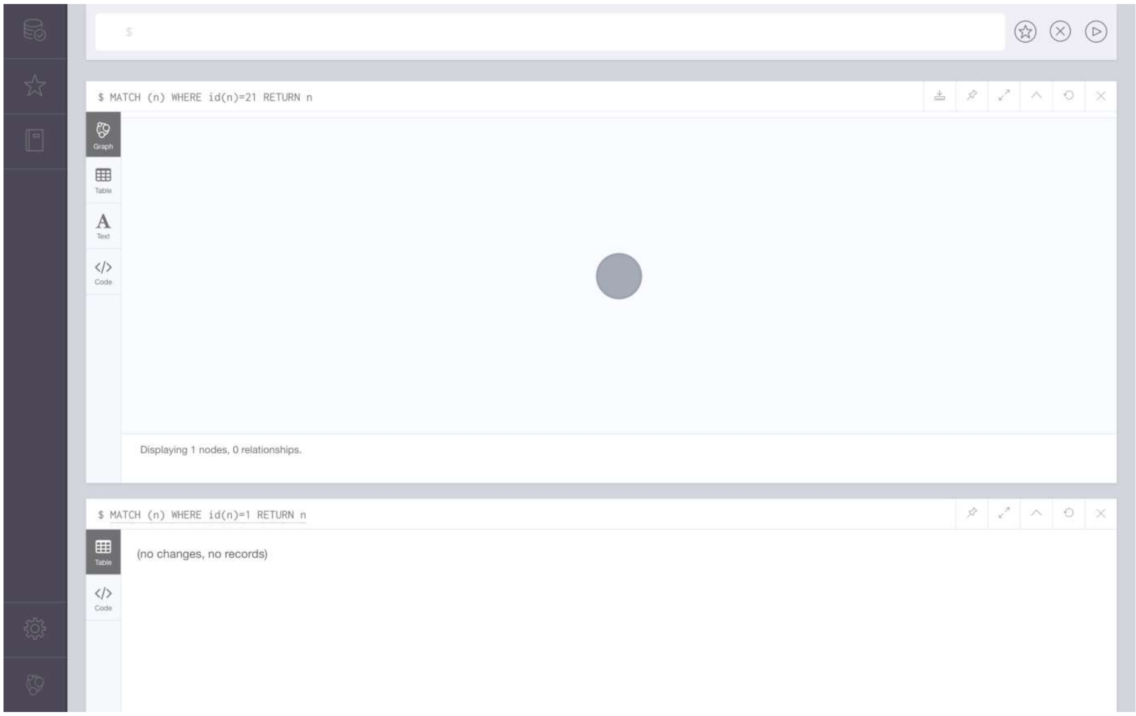
```
$ MATCH (n) WHERE id(n)=1 RETURN n
```

```
$ MATCH (n) RETURN n limit 2
```

Displaying 2 nodes, 0 relationships.

```
$ MATCH (n) RETURN n
```

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



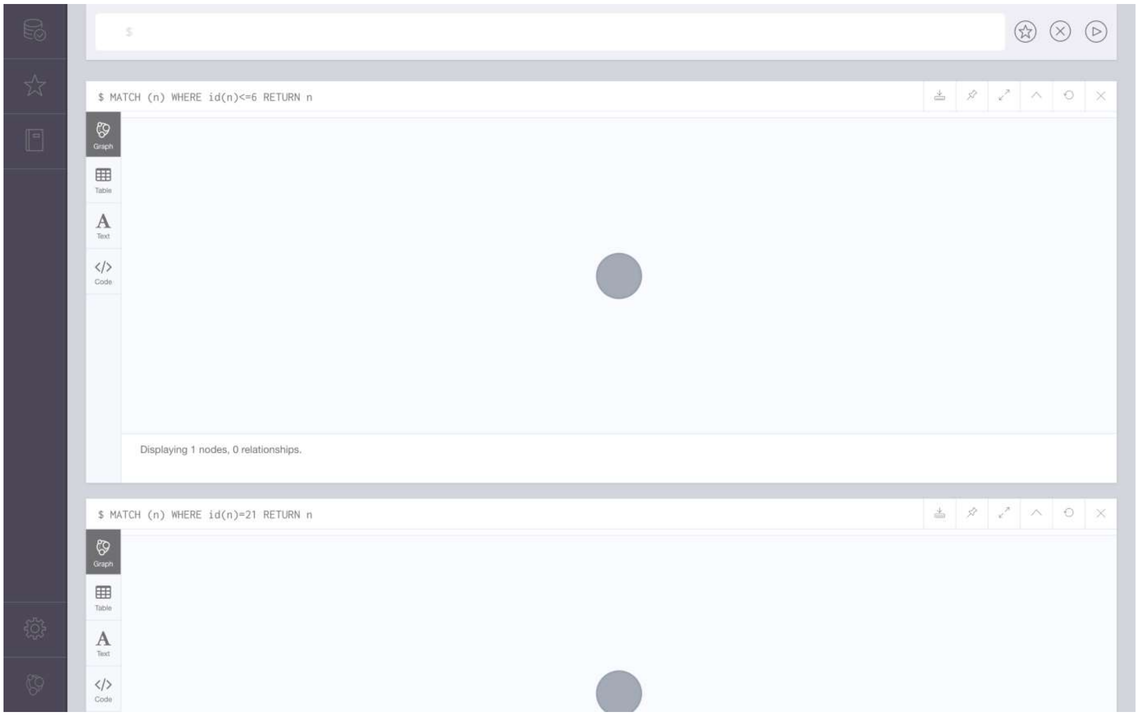
```
$ MATCH (n) WHERE id(n)=21 RETURN n
```

Displaying 1 nodes, 0 relationships.

```
$ MATCH (n) WHERE id(n)=1 RETURN n
```

(no changes, no records)

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

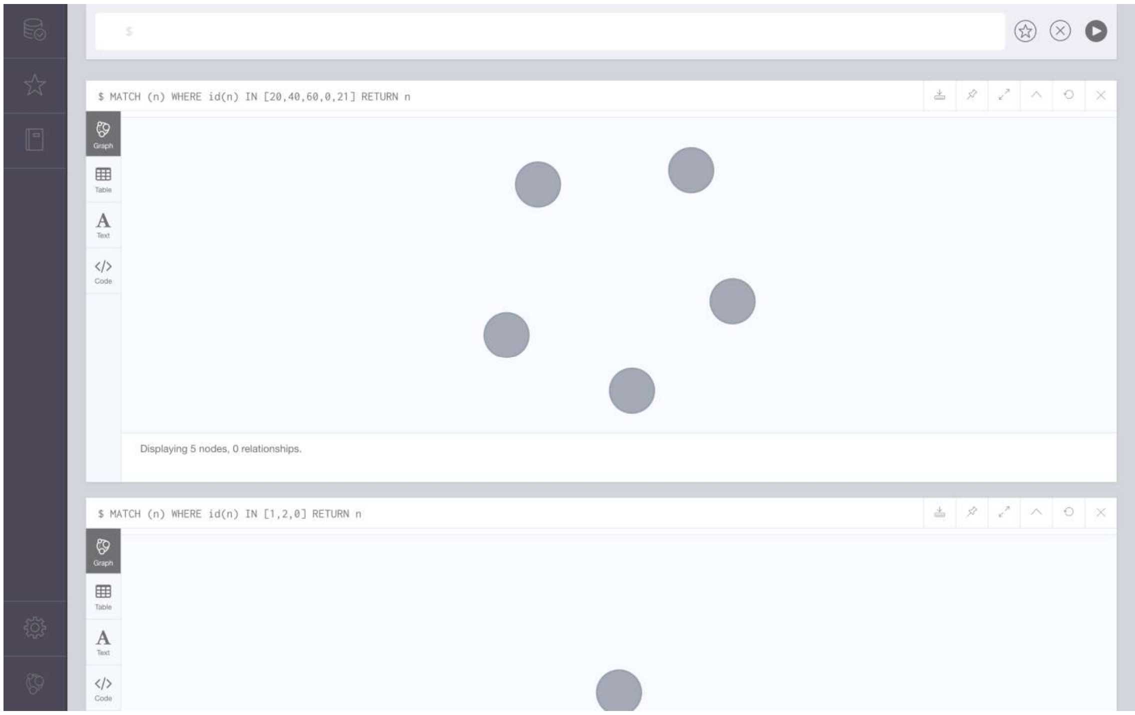


```
$ MATCH (n) WHERE id(n)<=6 RETURN n
```

Displaying 1 nodes, 0 relationships.

```
$ MATCH (n) WHERE id(n)=21 RETURN n
```

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

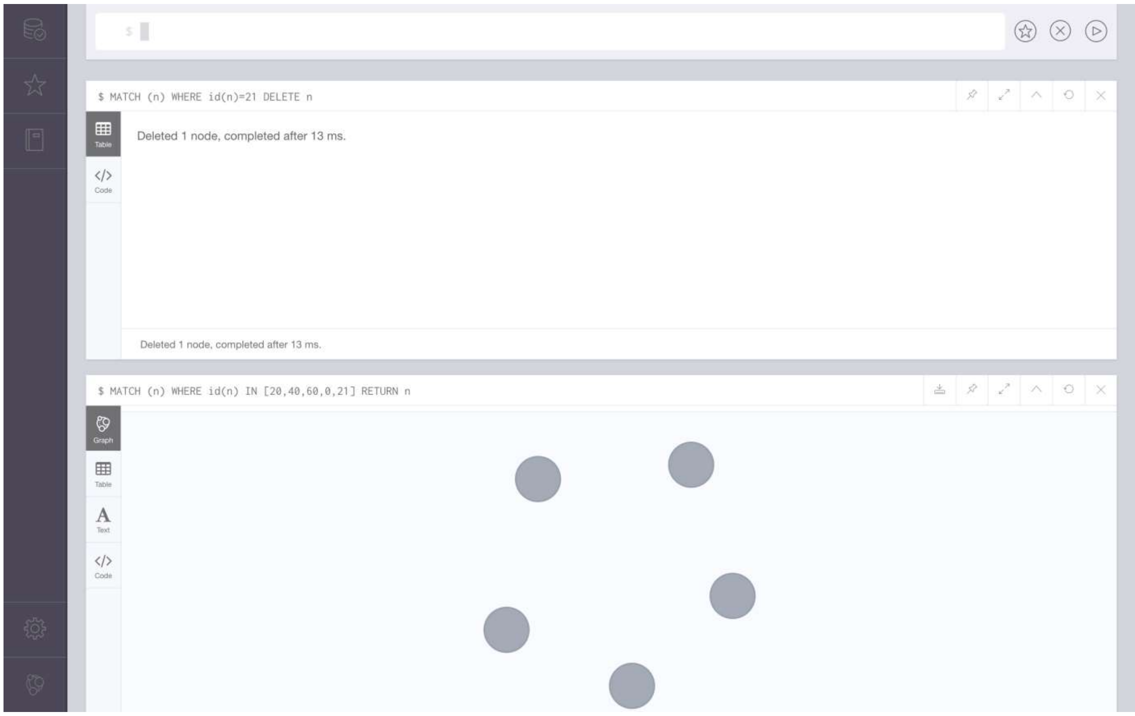


```
$ MATCH (n) WHERE id(n) IN [20,40,60,0,21] RETURN n
```

Displaying 5 nodes, 0 relationships.

```
$ MATCH (n) WHERE id(n) IN [1,2,0] RETURN n
```

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



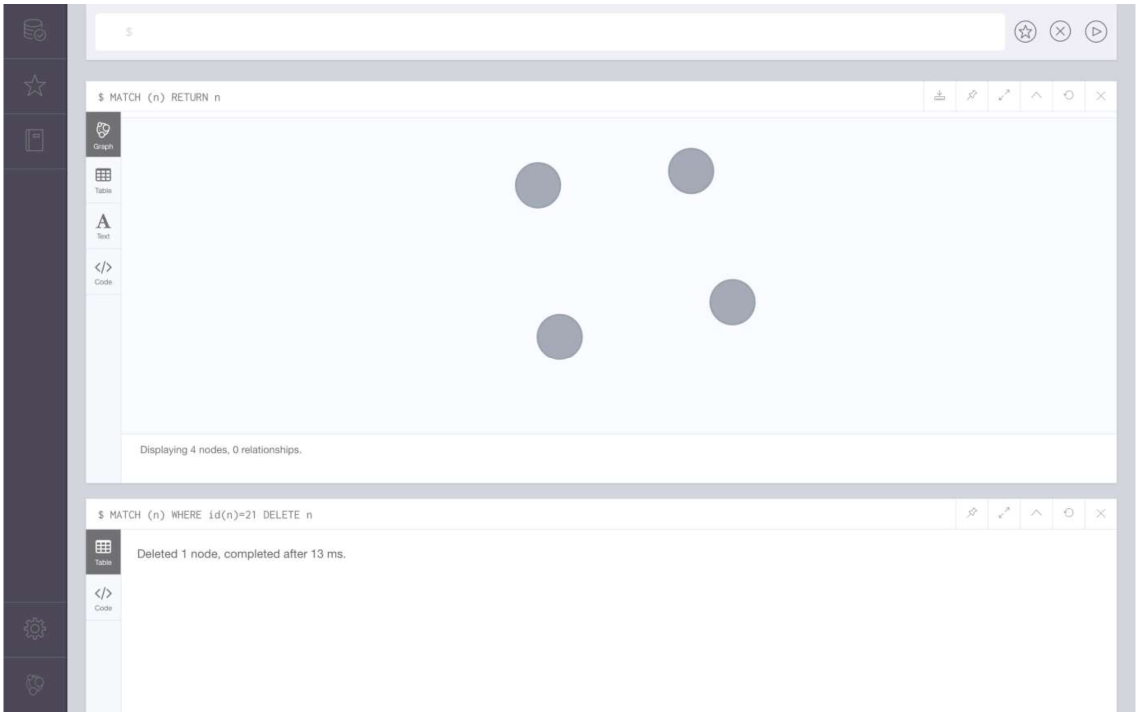
```
$ MATCH (n) WHERE id(n)=21 DELETE n
```

Deleted 1 node, completed after 13 ms.

Deleted 1 node, completed after 13 ms.

```
$ MATCH (n) WHERE id(n) IN [20,40,60,0,21] RETURN n
```

13) MATCH (n) RETURN n;



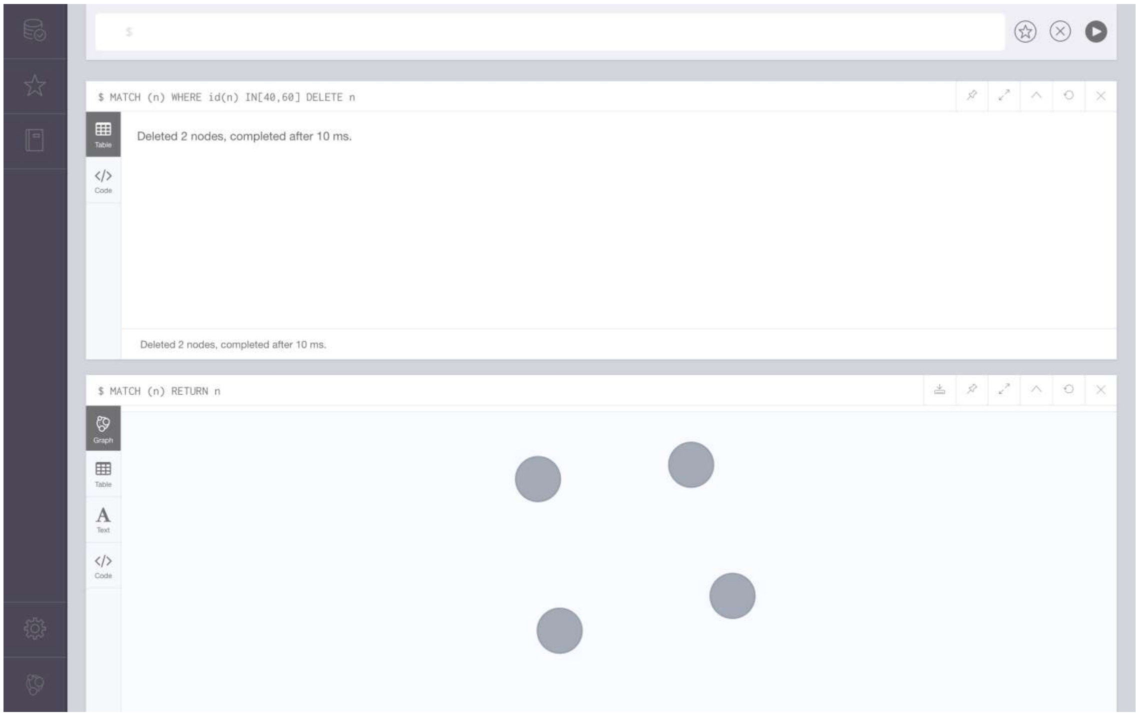
```
$ MATCH (n) RETURN n
```

Displaying 4 nodes, 0 relationships.

```
$ MATCH (n) WHERE id(n)=21 DELETE n
```

Deleted 1 node, completed after 13 ms.

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



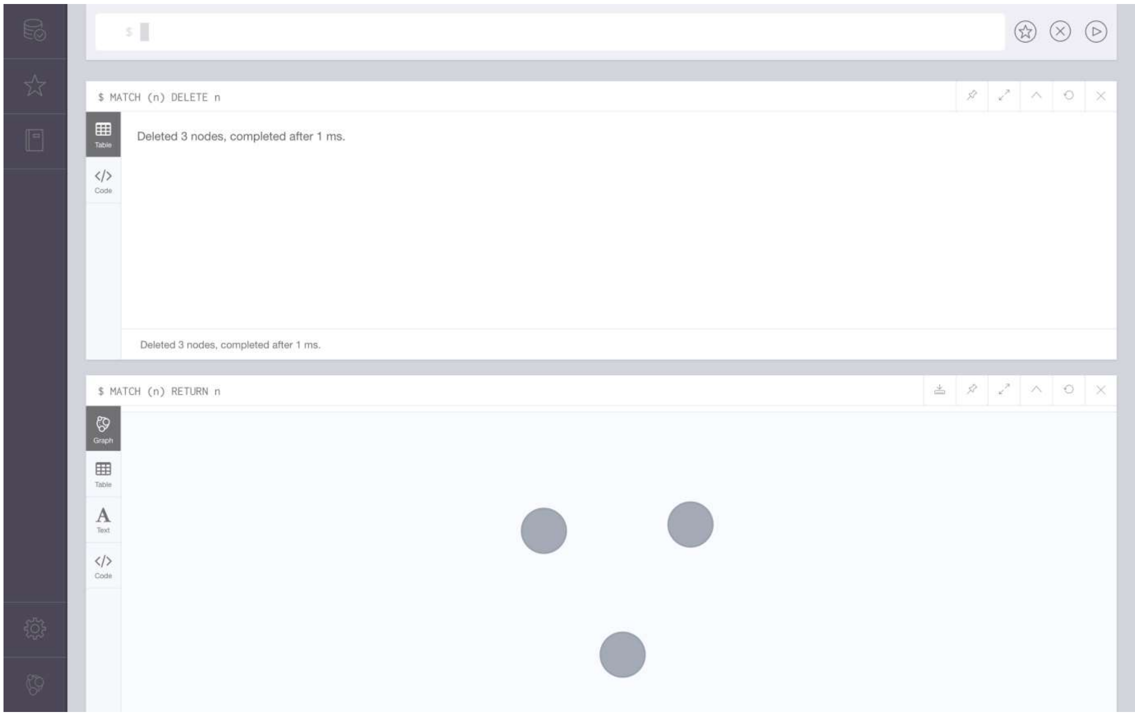
```
$ MATCH (n) WHERE id(n) IN[40,60] DELETE n
```

Deleted 2 nodes, completed after 10 ms.

Deleted 2 nodes, completed after 10 ms.

```
$ MATCH (n) RETURN n
```

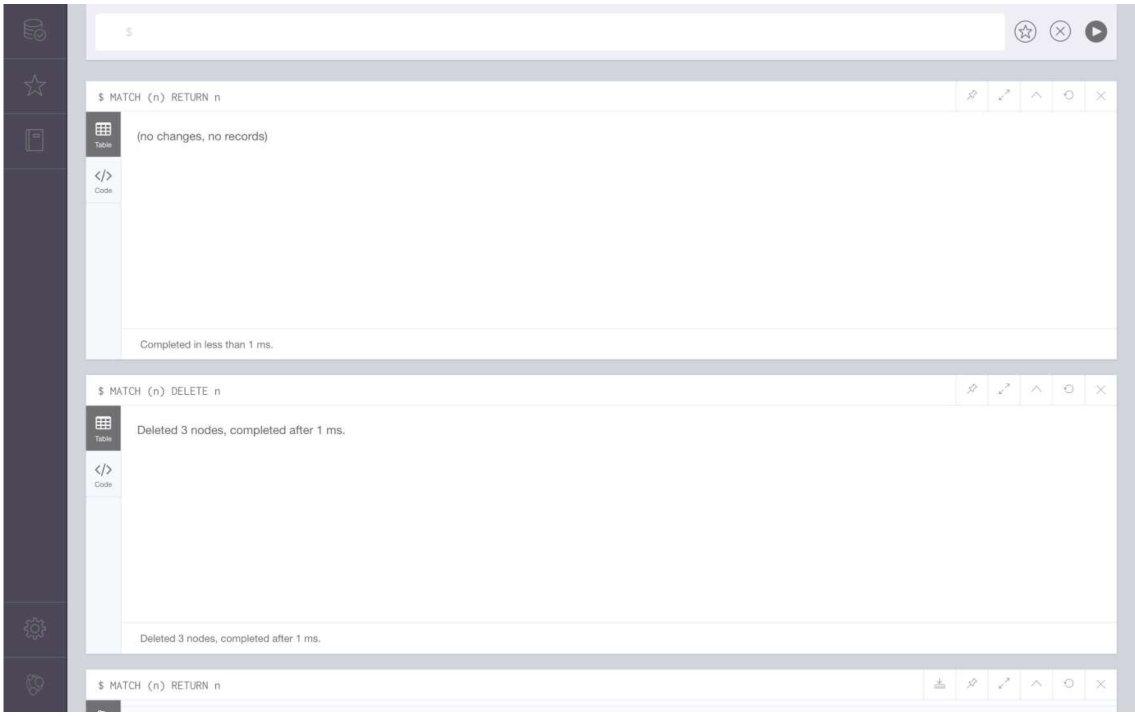
15) MATCH(n) DELETE n;



```
$ MATCH (n) DELETE n
Deleted 3 nodes, completed after 1 ms.

Deleted 3 nodes, completed after 1 ms.
```

```
$ MATCH (n) RETURN n
```

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

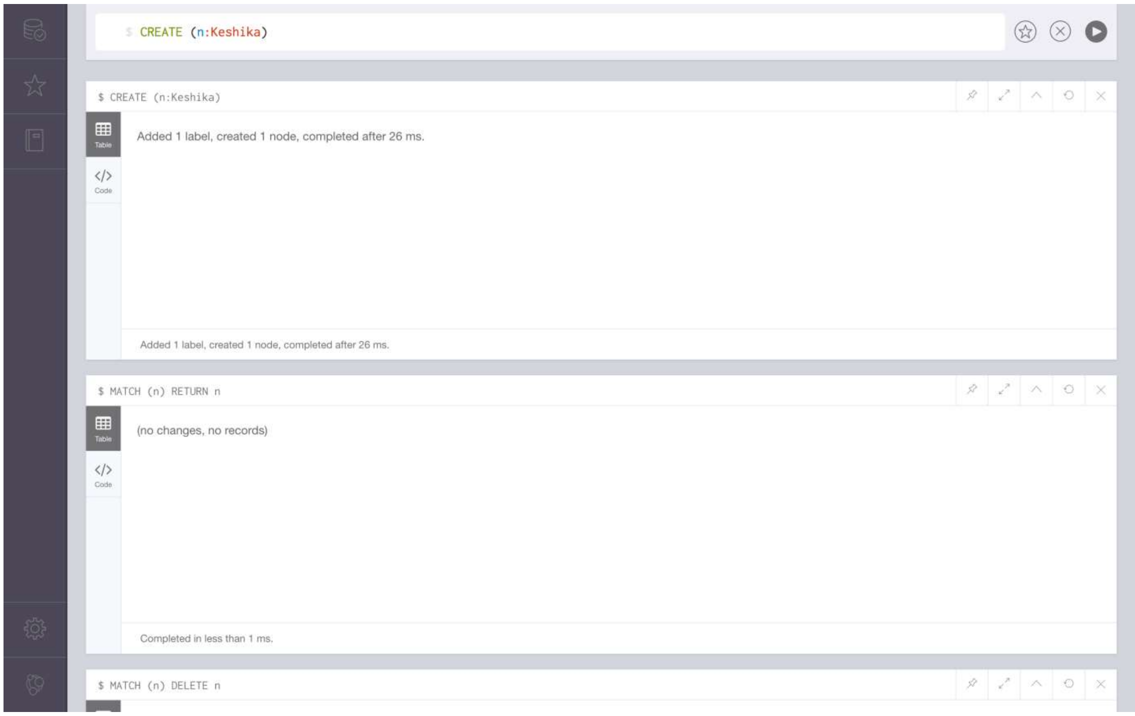


```
$ MATCH (n) RETURN n
(no changes, no records)

Completed in less than 1 ms.
```

```
$ MATCH (n) DELETE n
Deleted 3 nodes, completed after 1 ms.

Deleted 3 nodes, completed after 1 ms.
```

```
$ MATCH (n) RETURN n
```

17) CREATE (n:Keshika)



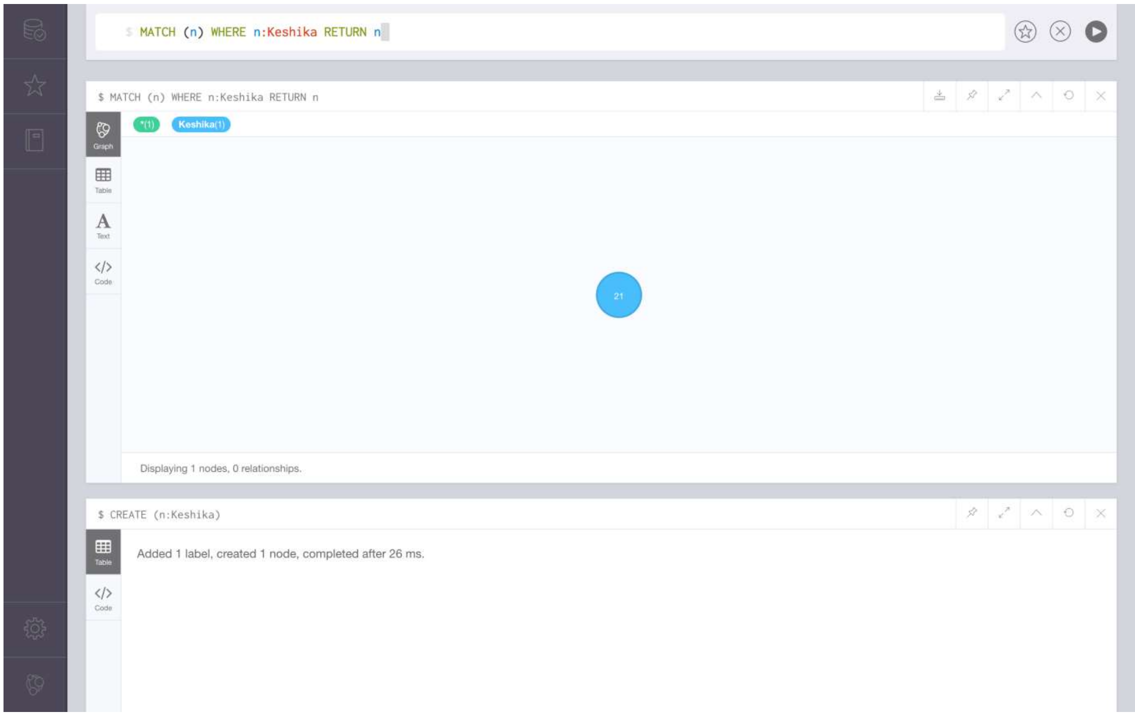
```
$ CREATE (n:Keshika)
```

```
$ CREATE (n:Keshika)
Added 1 label, created 1 node, completed after 26 ms.

Added 1 label, created 1 node, completed after 26 ms.
```

```
$ MATCH (n) RETURN n
(no changes, no records)

Completed in less than 1 ms.
```

```
$ MATCH (n) DELETE n
```

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

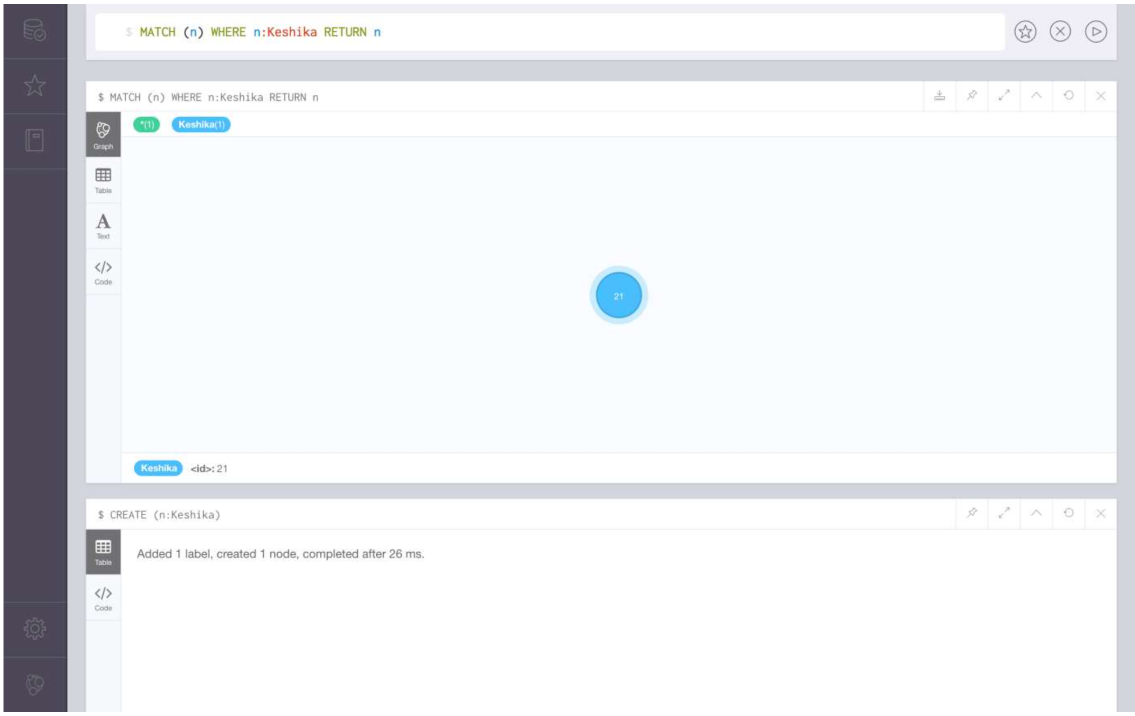


```
$ MATCH (n) WHERE n:Keshika RETURN n
```

```
$ MATCH (n) WHERE n:Keshika RETURN n
*(1)  Keshika(1)

Displaying 1 nodes, 0 relationships.
```

```
$ CREATE (n:Keshika)
Added 1 label, created 1 node, completed after 26 ms.
```

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



```
$ MATCH (n) WHERE n:Keshika RETURN n
```

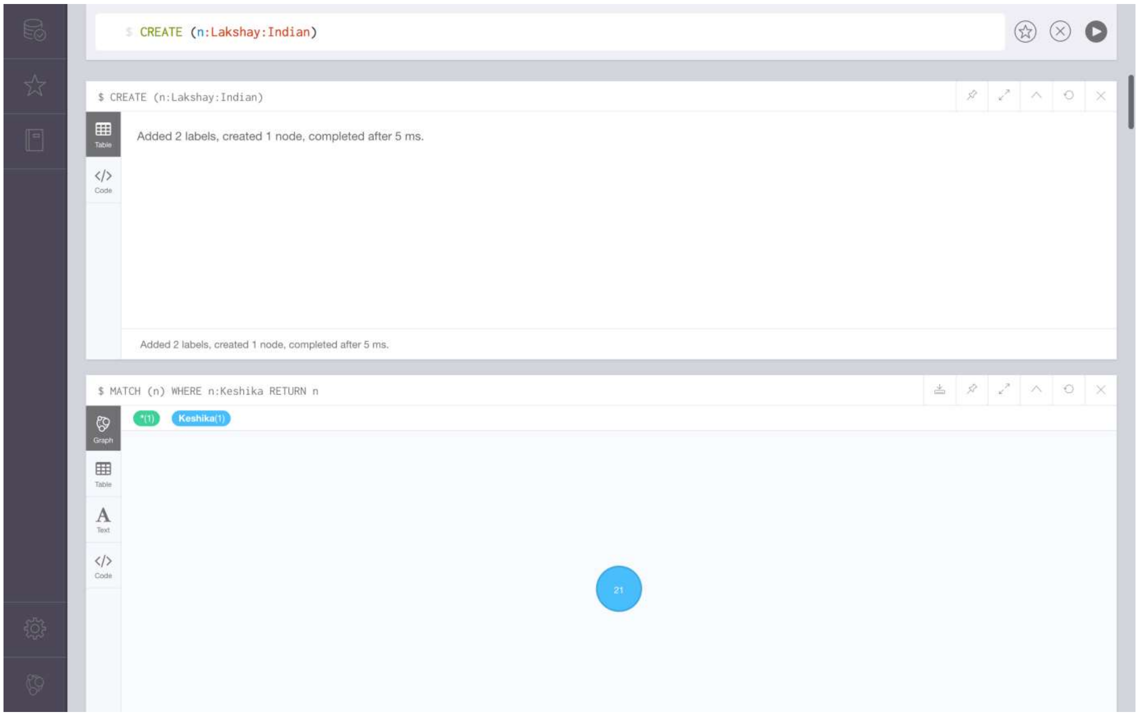
```
$ MATCH (n) WHERE n:Keshika RETURN n
```

```
<id>: 21
```

```
$ CREATE (n:Keshika)
```

```
Added 1 label, created 1 node, completed after 26 ms.
```

20) CREATE(n:Lakshay:Indian)



```
$ CREATE (n:Lakshay:Indian)
```

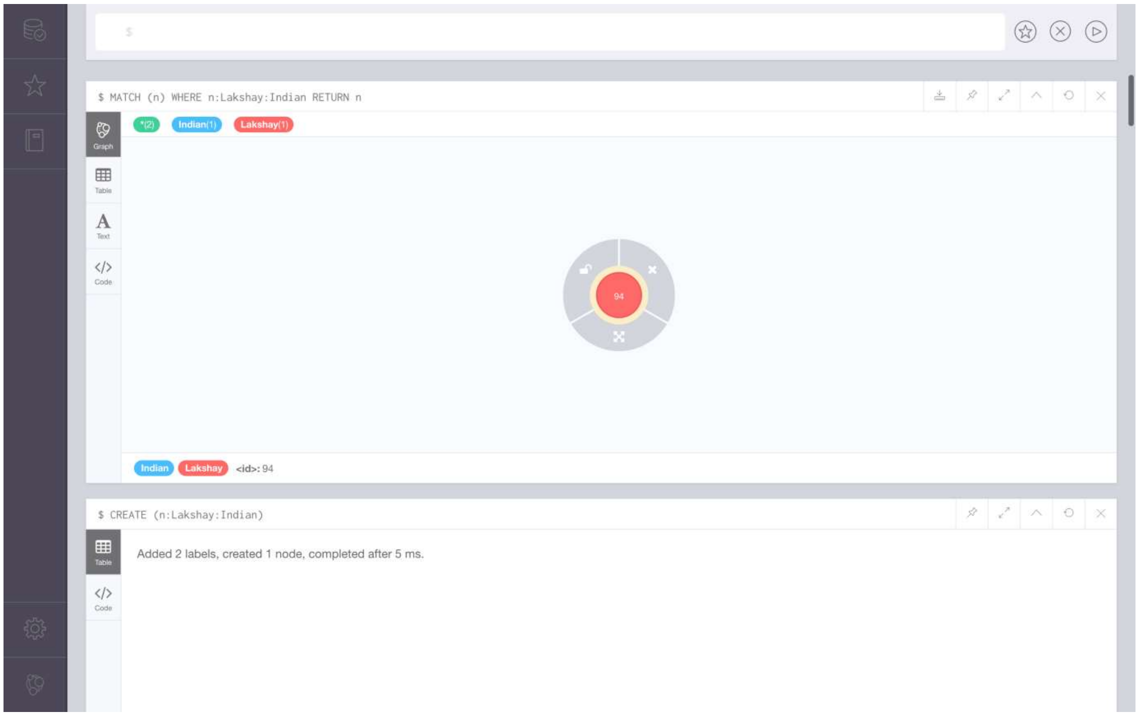
```
$ CREATE (n:Lakshay:Indian)
```

```
Added 2 labels, created 1 node, completed after 5 ms.
```

```
Added 2 labels, created 1 node, completed after 5 ms.
```

```
$ MATCH (n) WHERE n:Keshika RETURN n
```

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



```
$ MATCH (n) WHERE n:Lakshay:Indian RETURN n
```

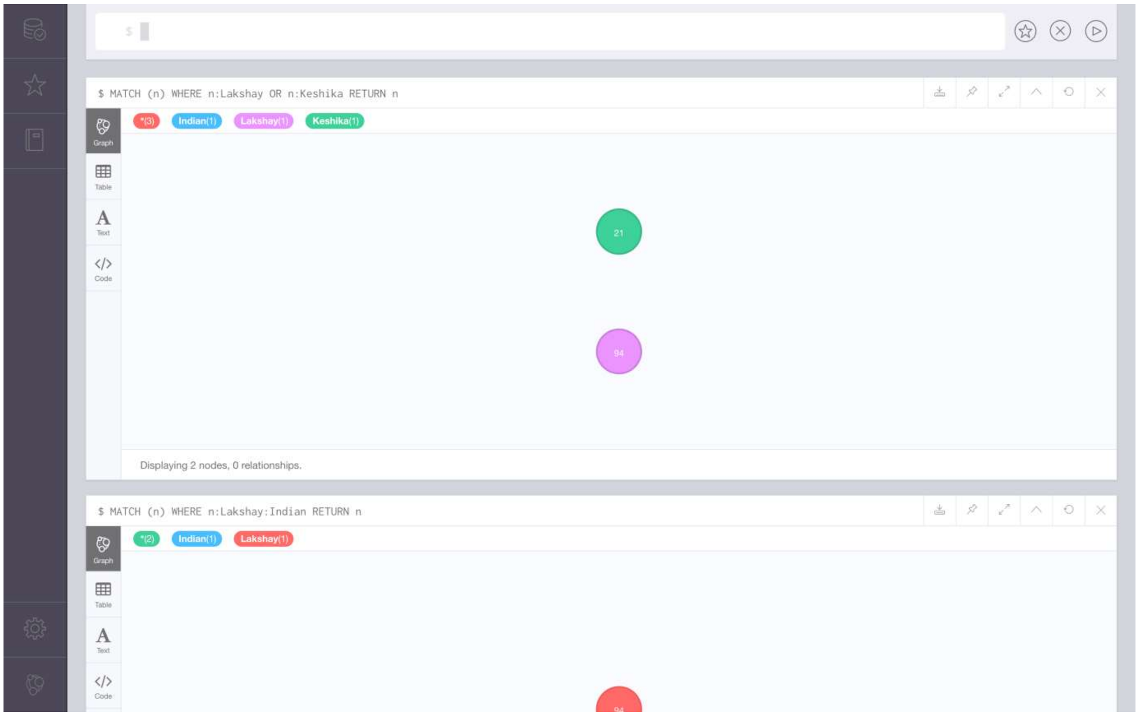
*(2) Indian(1) Lakshay(1)

Indian Lakshay <id>: 94

```
$ CREATE (n:Lakshay:Indian)
```

Added 2 labels, created 1 node, completed after 5 ms.

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



```
$ MATCH (n) WHERE n:Lakshay OR n:Keshika RETURN n
```

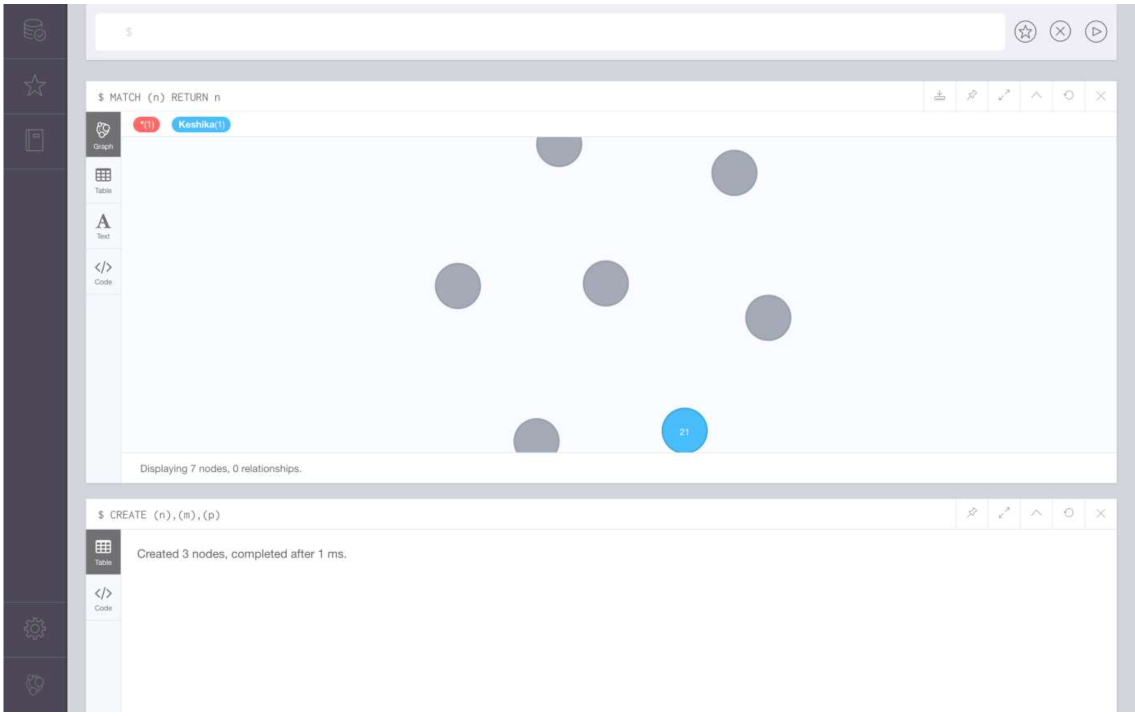
*(3) Indian(1) Lakshay(1) Keshika(1)

Displaying 2 nodes, 0 relationships.

```
$ MATCH (n) WHERE n:Lakshay:Indian RETURN n
```

*(2) Indian(1) Lakshay(1)

23) MATCH (n) RETURN n;



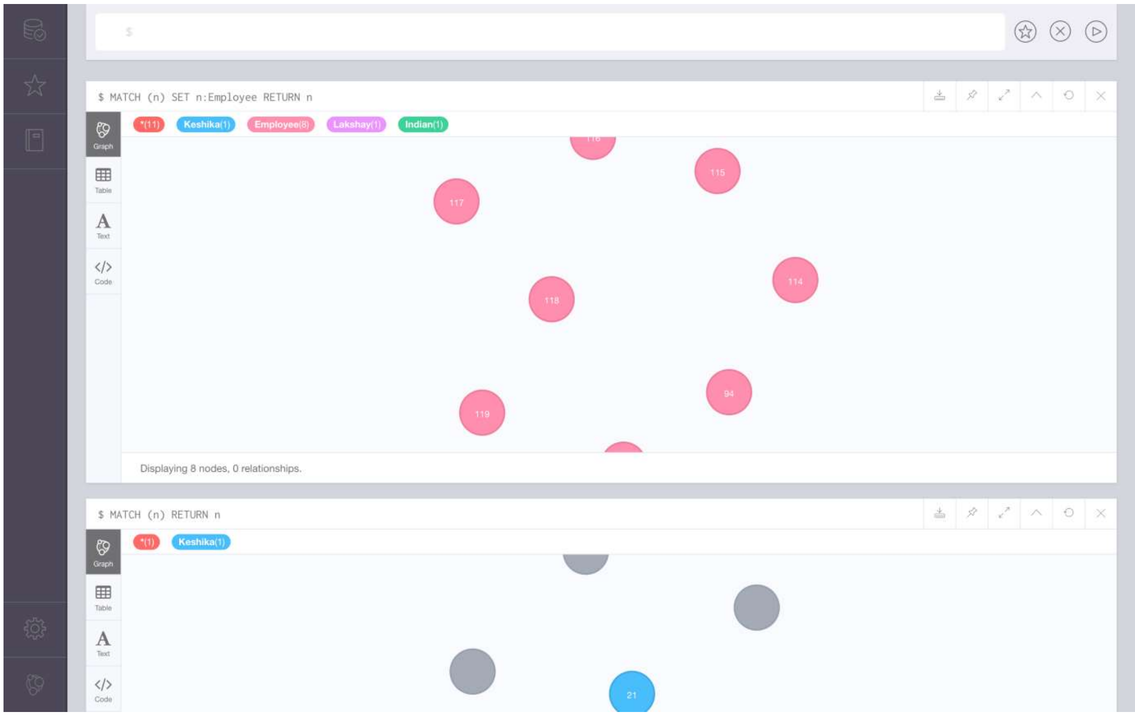
```
$ MATCH (n) RETURN n
```

Displaying 7 nodes, 0 relationships.

```
$ CREATE (n),(m),(p)
```

Created 3 nodes, completed after 1 ms.

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

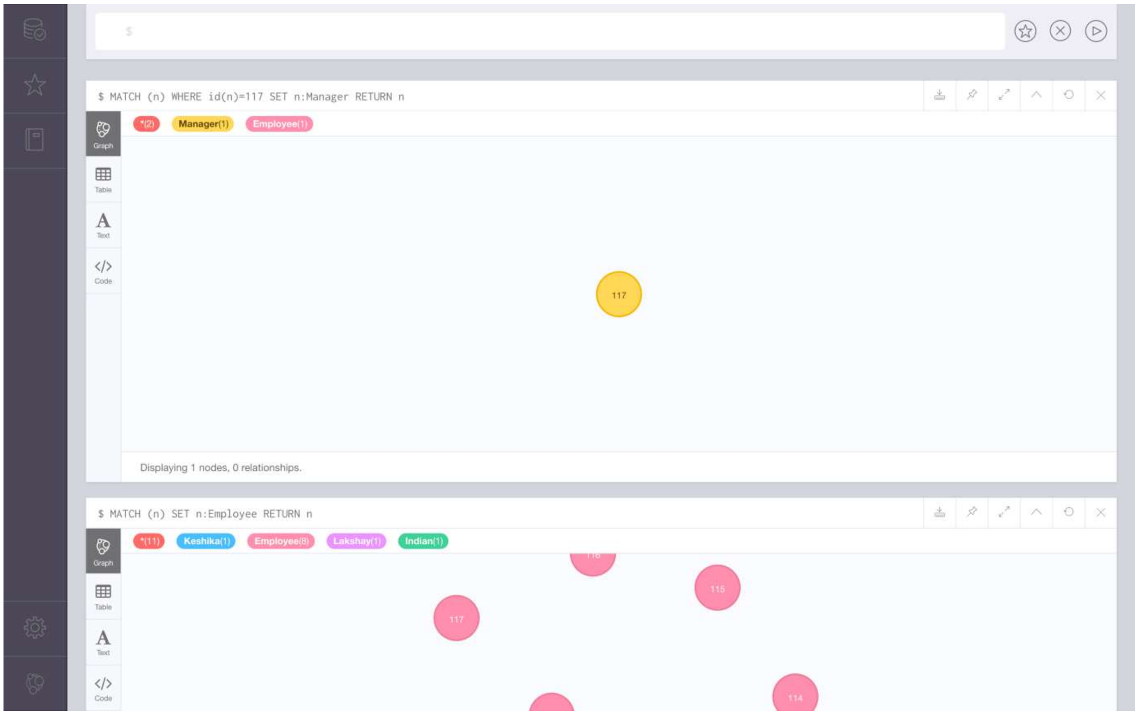


```
$ MATCH (n) SET n:Employee RETURN n
```

Displaying 8 nodes, 0 relationships.

```
$ MATCH (n) RETURN n
```

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

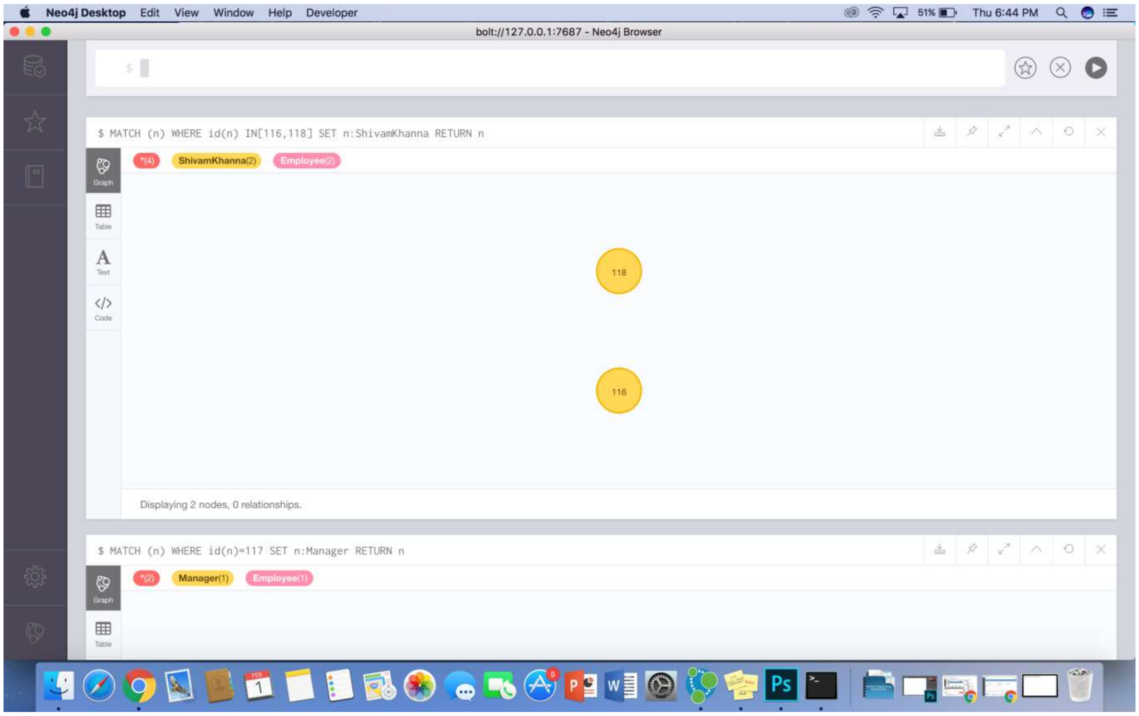


```
$ MATCH (n) WHERE id(n)=117 SET n:Manager RETURN n
```

Displaying 1 nodes, 0 relationships.

```
$ MATCH (n) SET n:Employee RETURN n
```

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

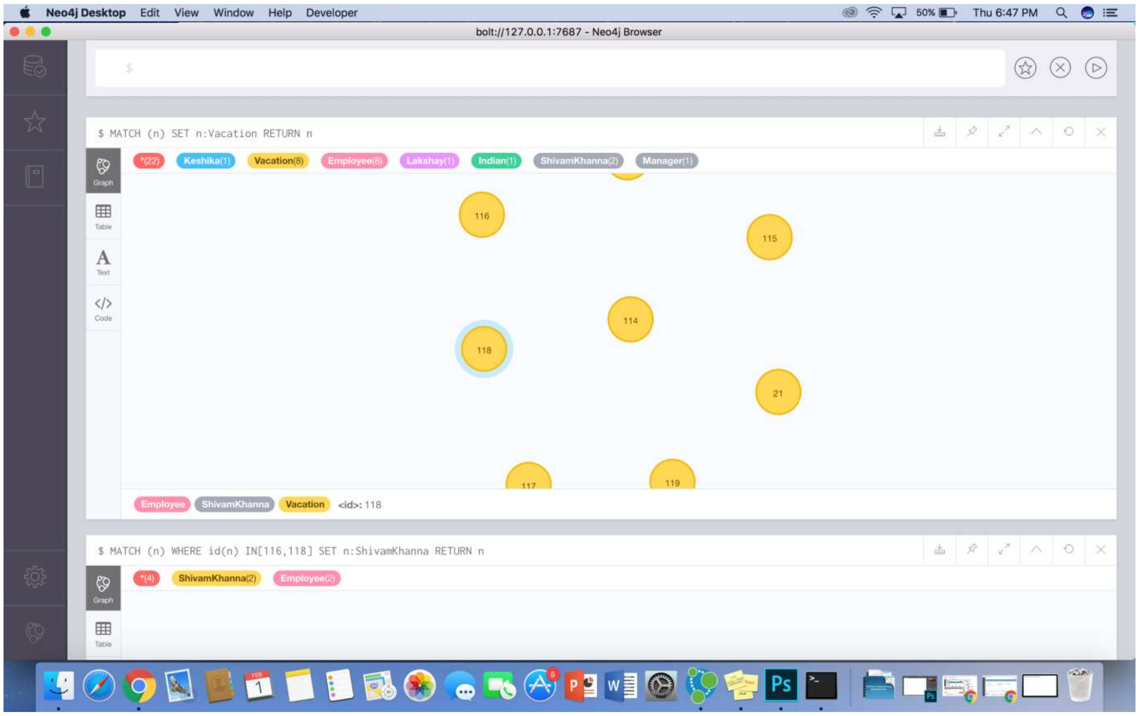


```
$ MATCH (n) WHERE id(n) IN[116,118] SET n:ShivamKhanna RETURN n
```

Displaying 2 nodes, 0 relationships.

```
$ MATCH (n) WHERE id(n)=117 SET n:Manager RETURN n
```

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



```
$ MATCH (n) SET n:Vacation RETURN n
```

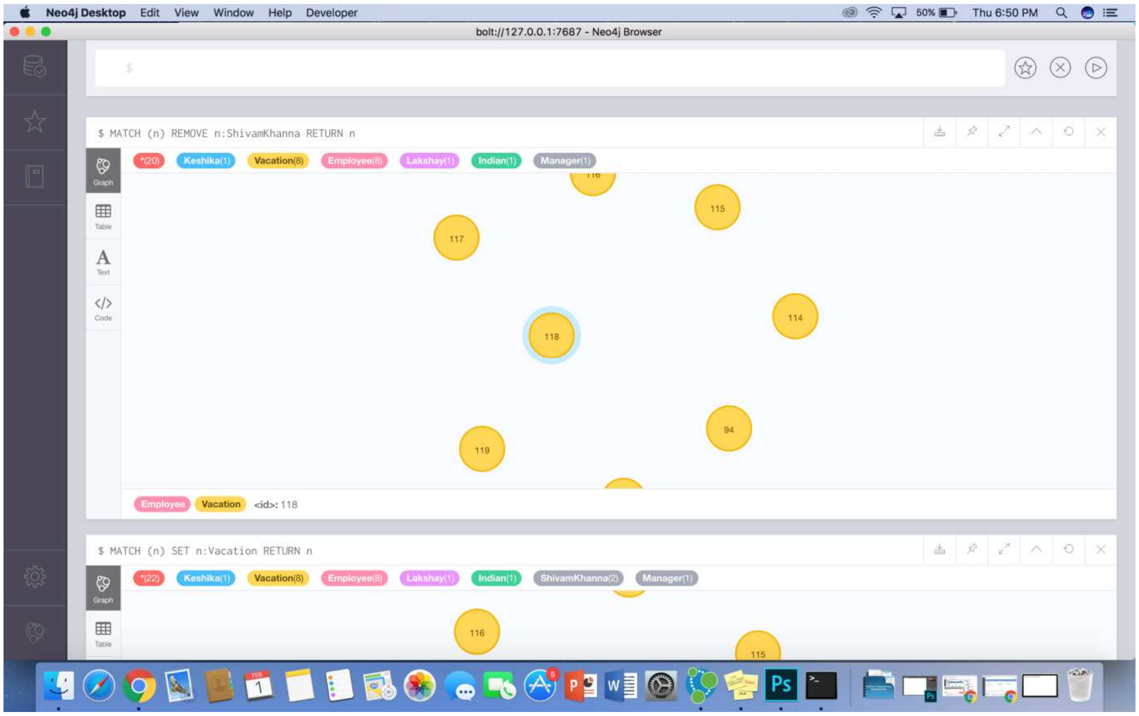
*(22) Keshika(1) Vacation(8) Employee(8) Lakshay(1) Indian(1) ShivamKhanna(2) Manager(1)

Employee ShivamKhanna Vacation <id>: 118

```
$ MATCH (n) WHERE id(n) IN[116,118] SET n:ShivamKhanna RETURN n
```

*(4) ShivamKhanna(2) Employee(2)

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



```
$ MATCH (n) REMOVE n:ShivamKhanna RETURN n
```

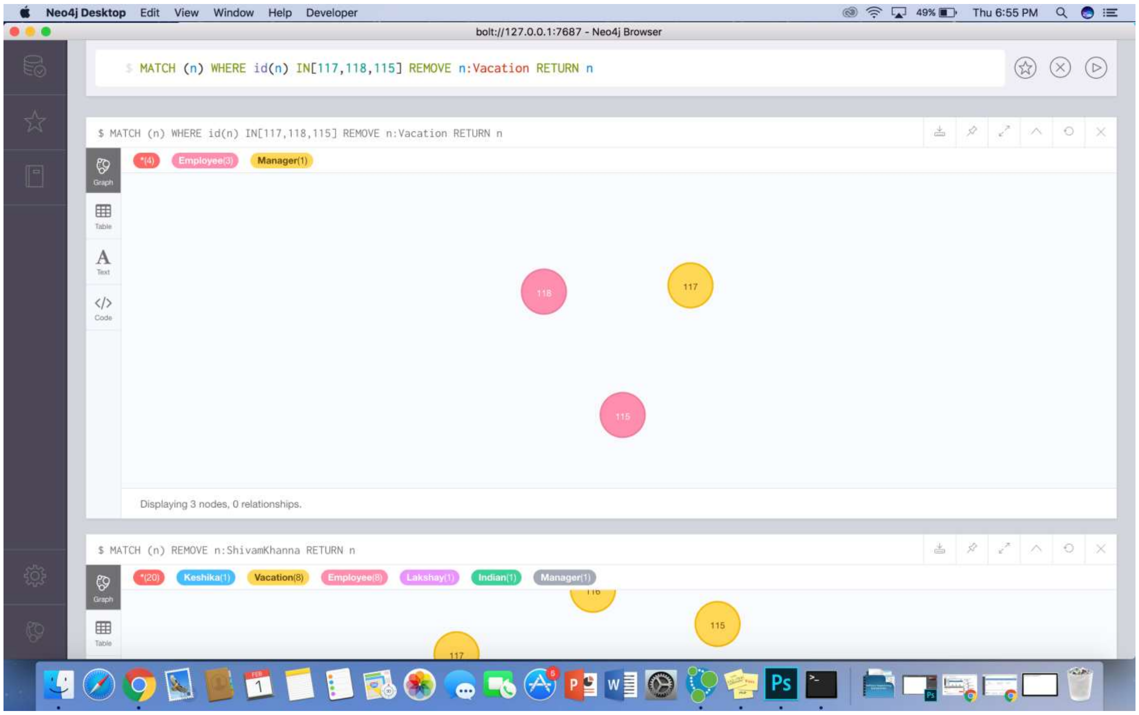
*(20) Keshika(1) Vacation(8) Employee(8) Lakshay(1) Indian(1) Manager(1)

Employee Vacation <id>: 118

```
$ MATCH (n) SET n:Vacation RETURN n
```

*(22) Keshika(1) Vacation(8) Employee(8) Lakshay(1) Indian(1) ShivamKhanna(2) Manager(1)

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



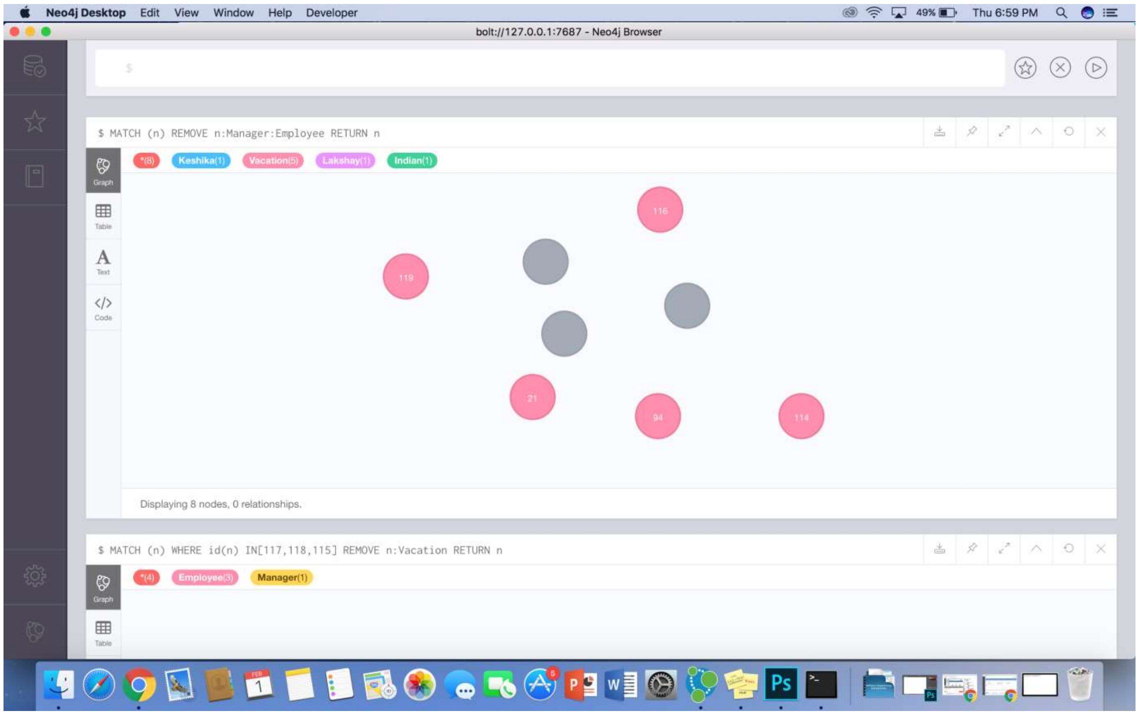
```
$ MATCH (n) WHERE id(n) IN[117,118,115] REMOVE n:Vacation RETURN n
```

```
$ MATCH (n) WHERE id(n) IN[117,118,115] REMOVE n:Vacation RETURN n
```

Displaying 3 nodes, 0 relationships.

```
$ MATCH (n) REMOVE n:ShivamKhanna RETURN n
```

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

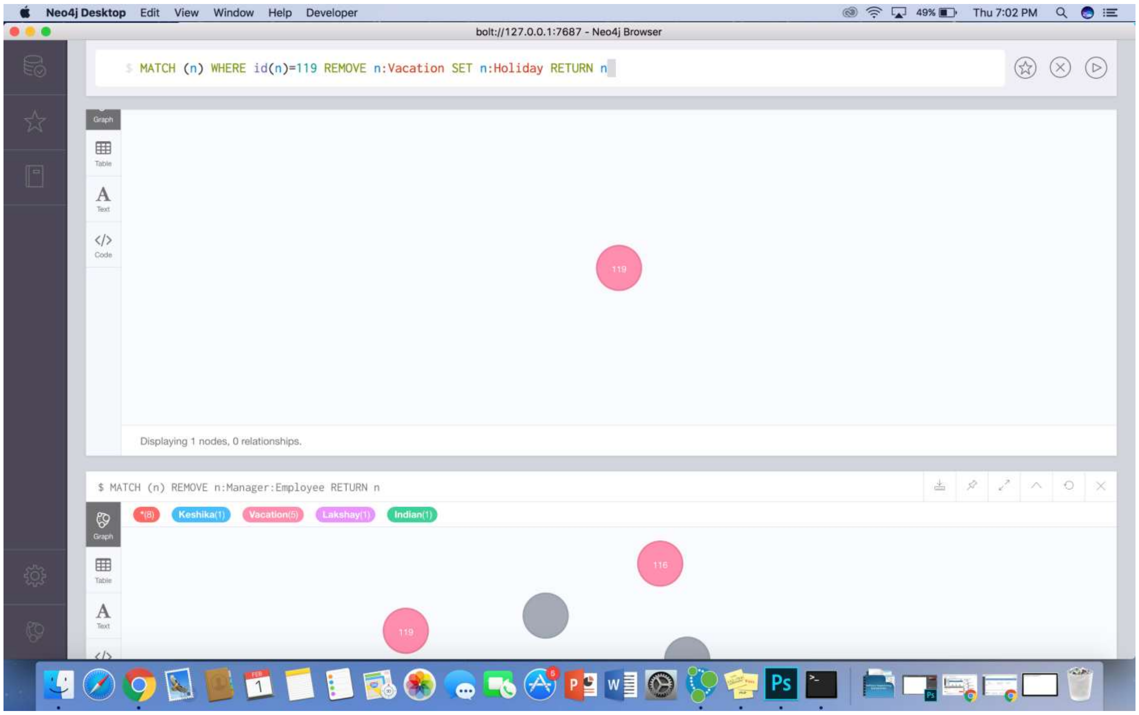


```
$ MATCH (n) REMOVE n:Manager:Employee RETURN n
```

Displaying 8 nodes, 0 relationships.

```
$ MATCH (n) WHERE id(n) IN[117,118,115] REMOVE n:Vacation RETURN n
```

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


Neo4j Desktop Edit View Window Help Developer
49% Thu 7:02 PM
bolt://127.0.0.1:7687 - Neo4j Browser
Graph
Table
Text
Code
119
116
119

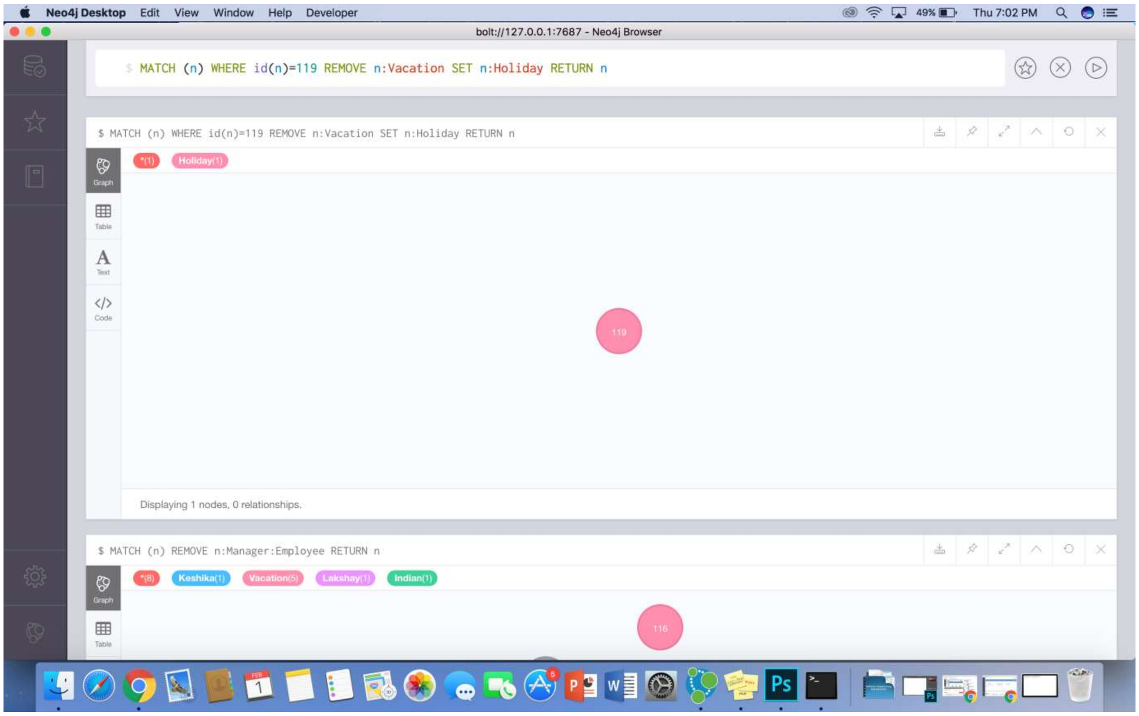
```
$ MATCH (n) WHERE id(n)=119 REMOVE n:Vacation SET n:Holiday RETURN n
```

Displaying 1 nodes, 0 relationships.

```
$ MATCH (n) REMOVE n:Manager:Employee RETURN n
```

*(8) Keshika(1) Vacation(5) Lakshay(1) Indian(1)

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


Neo4j Desktop Edit View Window Help Developer
49% Thu 7:02 PM
bolt://127.0.0.1:7687 - Neo4j Browser
Graph
Table
Text
Code
119
116

```
$ MATCH (n) WHERE id(n)=119 REMOVE n:Vacation SET n:Holiday RETURN n
```

```
$ MATCH (n) WHERE id(n)=119 REMOVE n:Vacation SET n:Holiday RETURN n
```

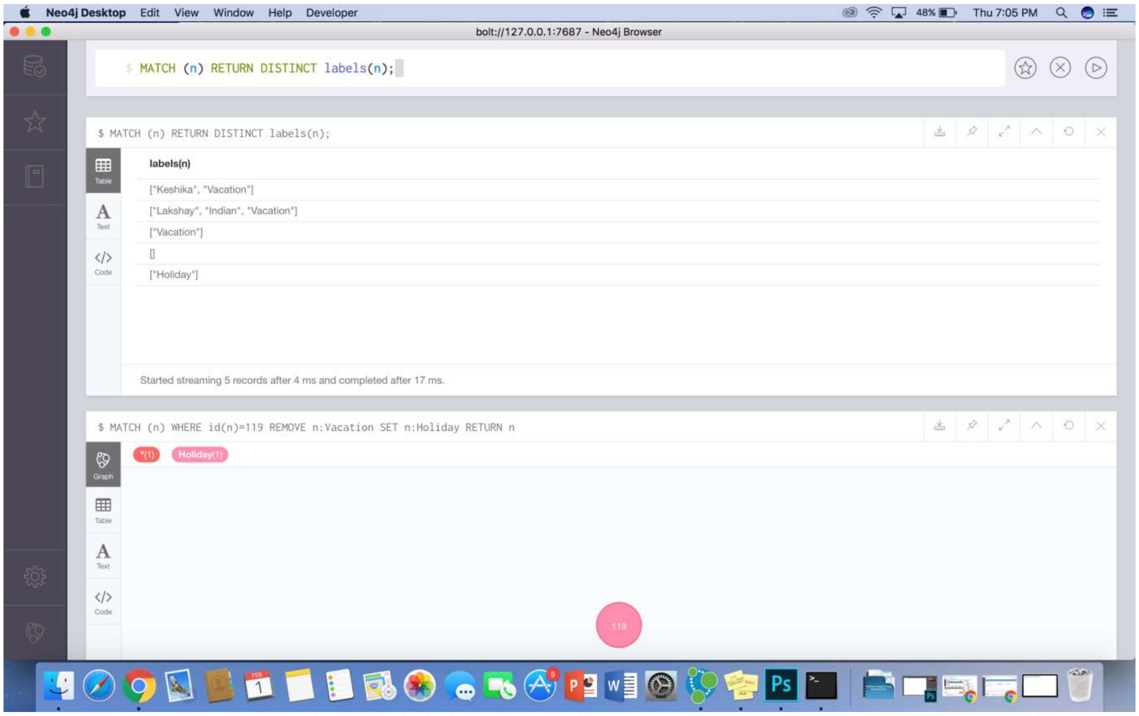
*(1) Holiday(1)

Displaying 1 nodes, 0 relationships.

```
$ MATCH (n) REMOVE n:Manager:Employee RETURN n
```

*(8) Keshika(1) Vacation(5) Lakshay(1) Indian(1)

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



```
$ MATCH (n) RETURN DISTINCT labels(n);
```

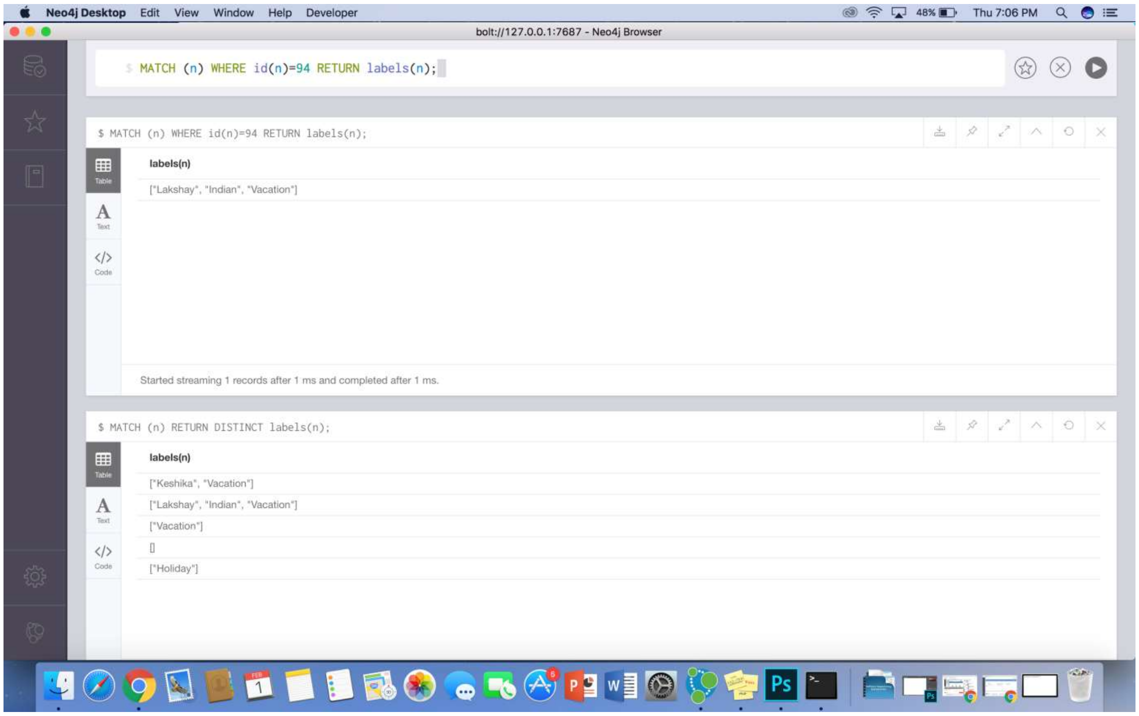
```
$ MATCH (n) RETURN DISTINCT labels(n);
```

labels(n)
["Keshika", "Vacation"]
["Lakshay", "Indian", "Vacation"]
["Vacation"]
[]
["Holiday"]

Started streaming 5 records after 4 ms and completed after 17 ms.

```
$ MATCH (n) WHERE id(n)=119 REMOVE n:Vacation SET n:Holiday RETURN n
```

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



```
$ MATCH (n) WHERE id(n)=94 RETURN labels(n);
```

```
$ MATCH (n) WHERE id(n)=94 RETURN labels(n);
```

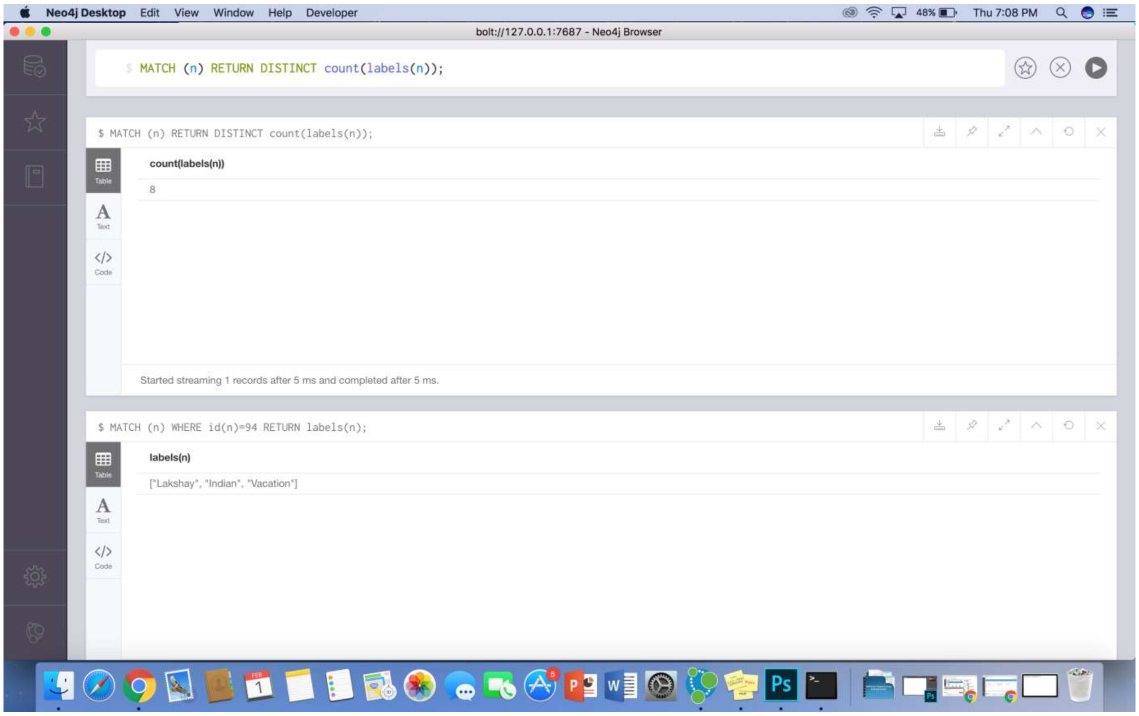
labels(n)
["Lakshay", "Indian", "Vacation"]

Started streaming 1 records after 1 ms and completed after 1 ms.

```
$ MATCH (n) RETURN DISTINCT labels(n);
```

labels(n)
["Keshika", "Vacation"]
["Lakshay", "Indian", "Vacation"]
["Vacation"]
[]
["Holiday"]

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



```
$ MATCH (n) RETURN DISTINCT count(labels(n));
```

```
$ MATCH (n) RETURN DISTINCT count(labels(n));
```

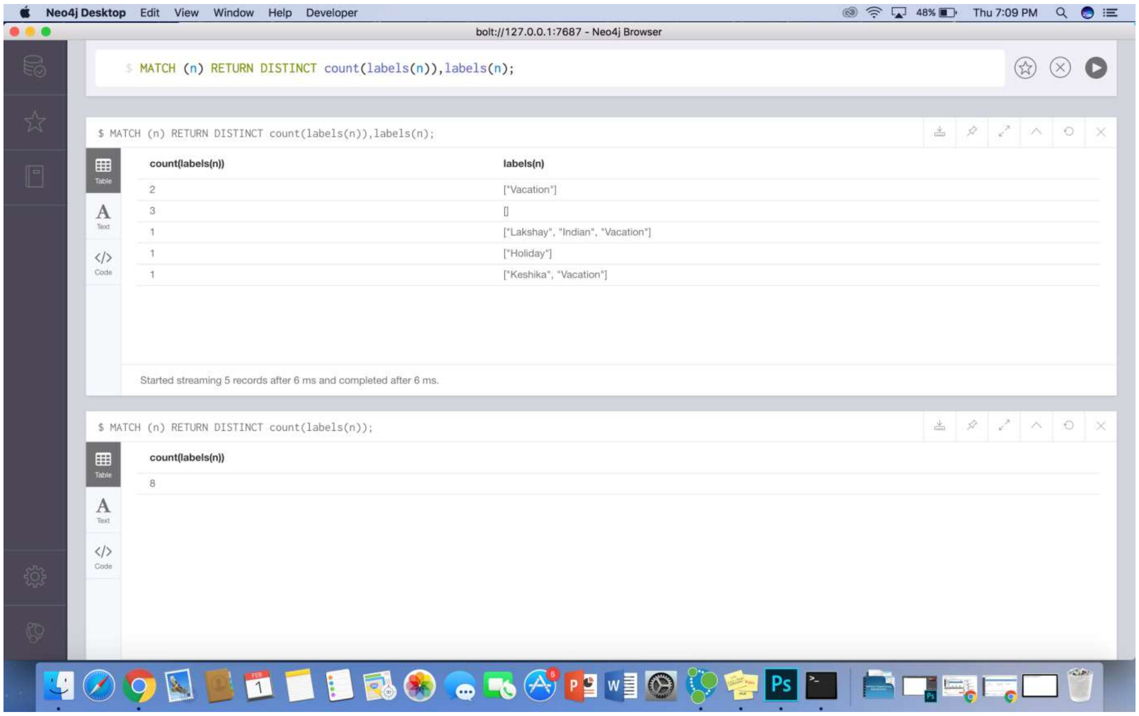
count(labels(n))
8

Started streaming 1 records after 5 ms and completed after 5 ms.

```
$ MATCH (n) WHERE id(n)=94 RETURN labels(n);
```

labels(n)
["Lakshay", "Indian", "Vacation"]

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



```
$ MATCH (n) RETURN DISTINCT count(labels(n)),labels(n);
```

```
$ MATCH (n) RETURN DISTINCT count(labels(n)),labels(n);
```

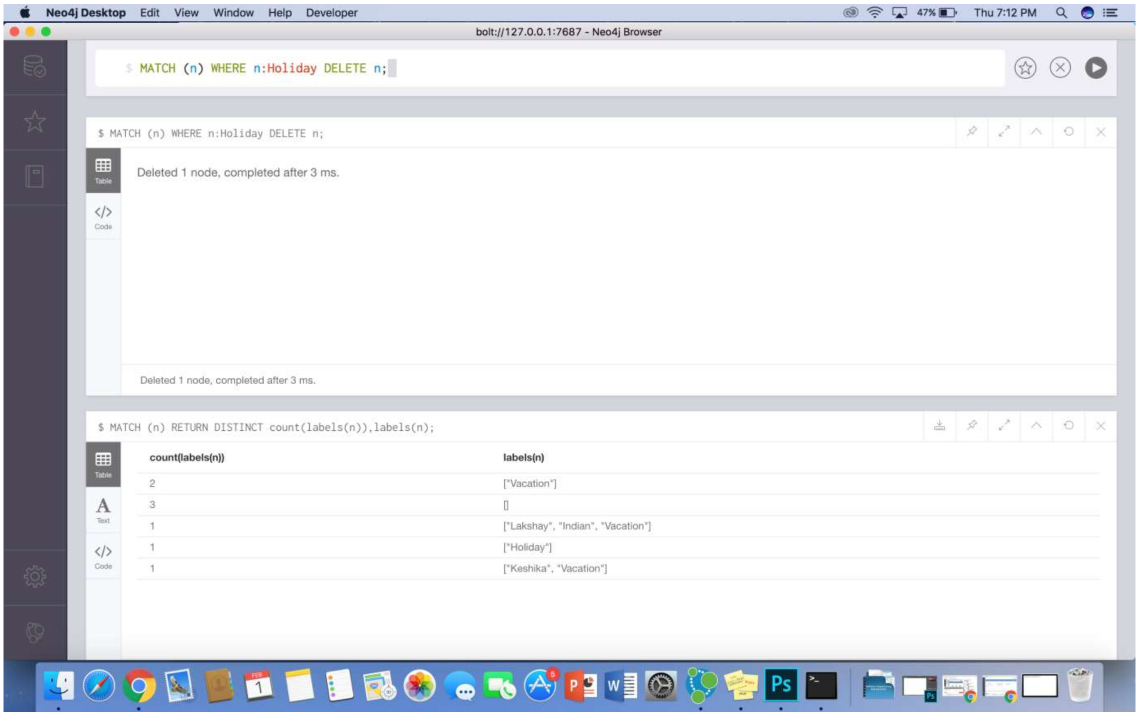
count(labels(n))	labels(n)
2	["Vacation"]
3	[]
1	["Lakshay", "Indian", "Vacation"]
1	["Holiday"]
1	["Keshika", "Vacation"]

Started streaming 5 records after 6 ms and completed after 6 ms.

```
$ MATCH (n) RETURN DISTINCT count(labels(n));
```

count(labels(n))
8

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


Neo4j Desktop Edit View Window Help Developer
47% Thu 7:12 PM
bolt://127.0.0.1:7687 - Neo4j Browser
Table
Code
Table
Text
Code

```
$ MATCH (n) WHERE n:Holiday DELETE n;
```

```
$ MATCH (n) WHERE n:Holiday DELETE n;
```

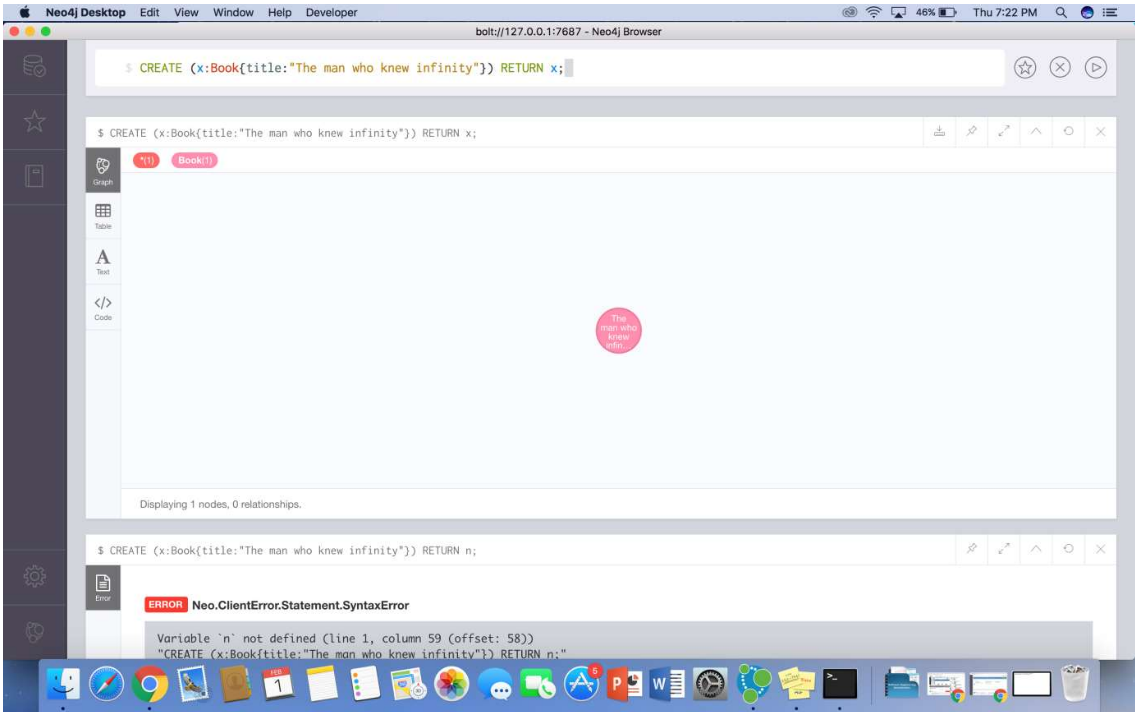
Deleted 1 node, completed after 3 ms.

Deleted 1 node, completed after 3 ms.

```
$ MATCH (n) RETURN DISTINCT count(labels(n)),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;


Neo4j Desktop Edit View Window Help Developer
46% Thu 7:22 PM
bolt://127.0.0.1:7687 - Neo4j Browser
Graph
Table
Text
Code
*(1)
Book(1)
The man who knew infin...
Error

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

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

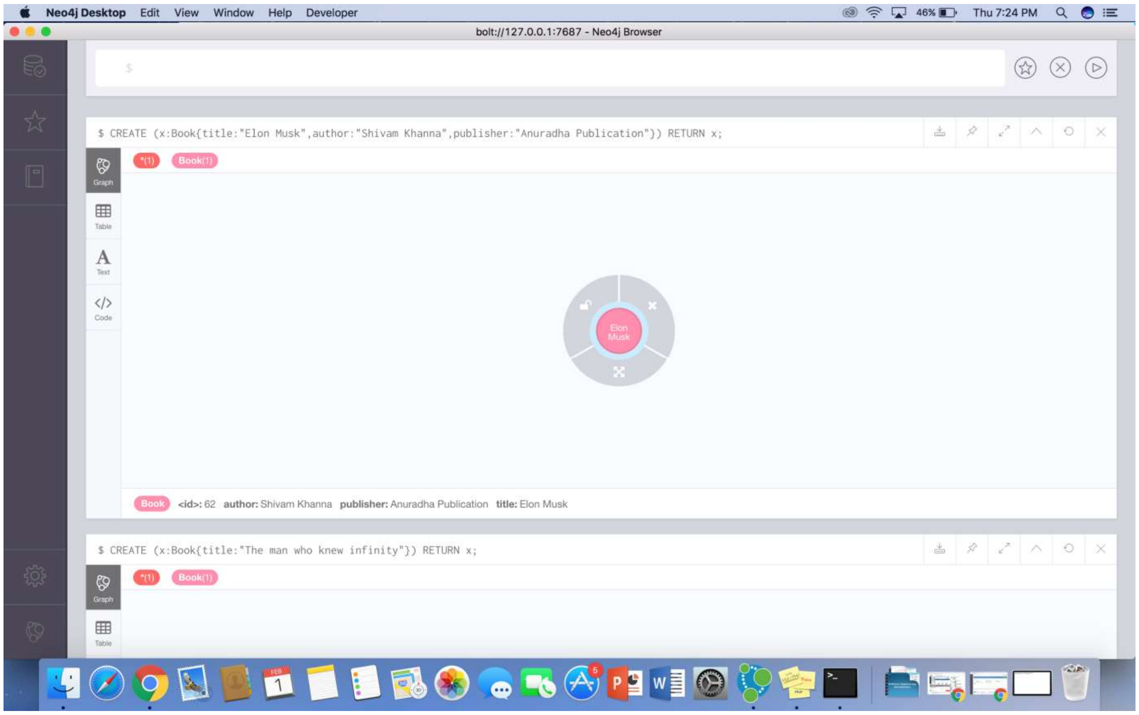
Displaying 1 nodes, 0 relationships.

```
$ CREATE (x:Book{title:"The man who knew infinity"}) RETURN n;
```

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;"
```

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

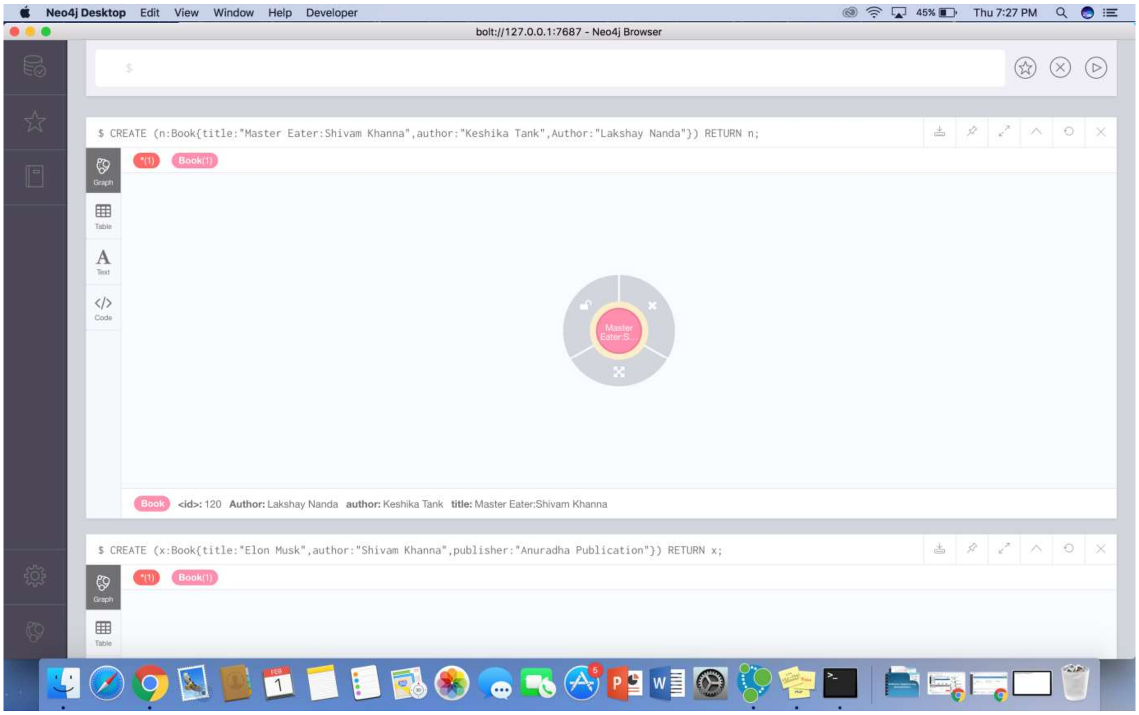

Neo4j Desktop Edit View Window Help Developer
46% Thu 7:24 PM
bolt://127.0.0.1:7687 - Neo4j Browser
Graph
Table
Text
Code
*(1)
Book(1)
Elon Musk

```
$ CREATE (x:Book{title:"Elon Musk",author:"Shivam Khanna",publisher:"Anuradha Publication"}) RETURN x;
```

Book <id>: 62 author: Shivam Khanna publisher: Anuradha Publication title: Elon Musk

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

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

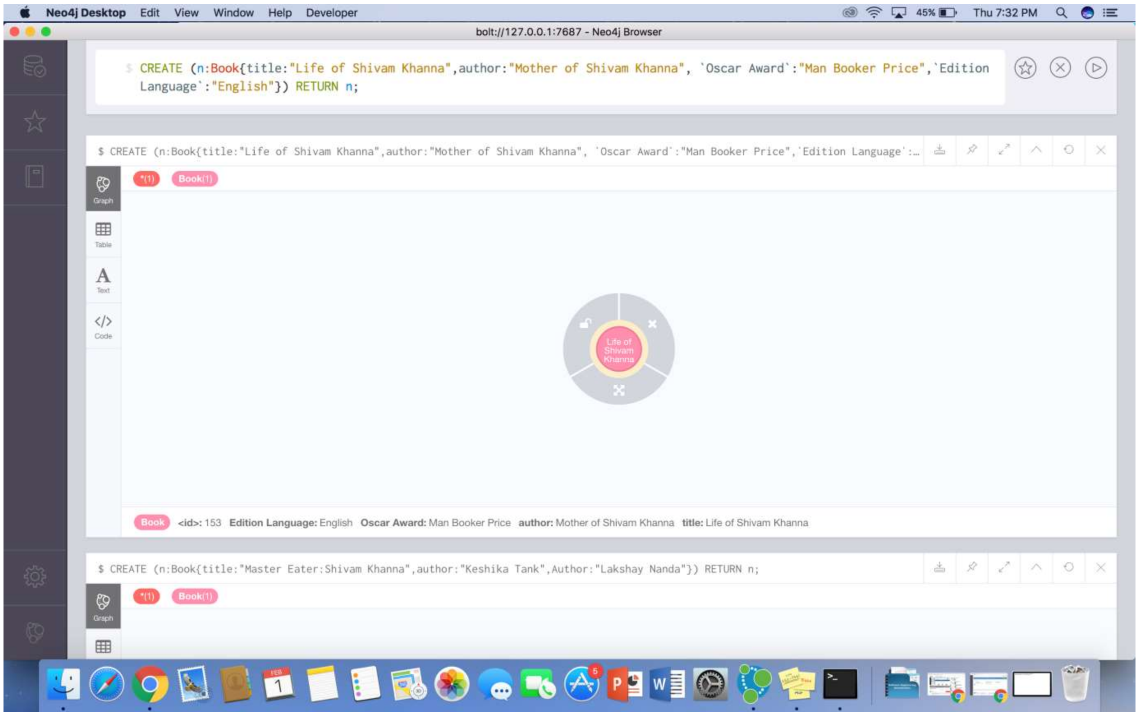

Neo4j Desktop Edit View Window Help Developer
45% Thu 7:27 PM
bolt://127.0.0.1:7687 - Neo4j Browser
Graph
Table
Text
Code
*(1)
Book(1)
Master Eater:S...

```
$ CREATE (n:Book{title:"Master Eater:Shivam Khanna",author:"Keshika Tank",Author:"Lakshay Nanda"}) RETURN n;
```

Book <id>: 120 Author: Lakshay Nanda author: Keshika Tank title: Master Eater:Shivam Khanna

```
$ CREATE (x:Book{title:"Elon Musk",author:"Shivam Khanna",publisher:"Anuradha Publication"}) RETURN x;
```

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



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

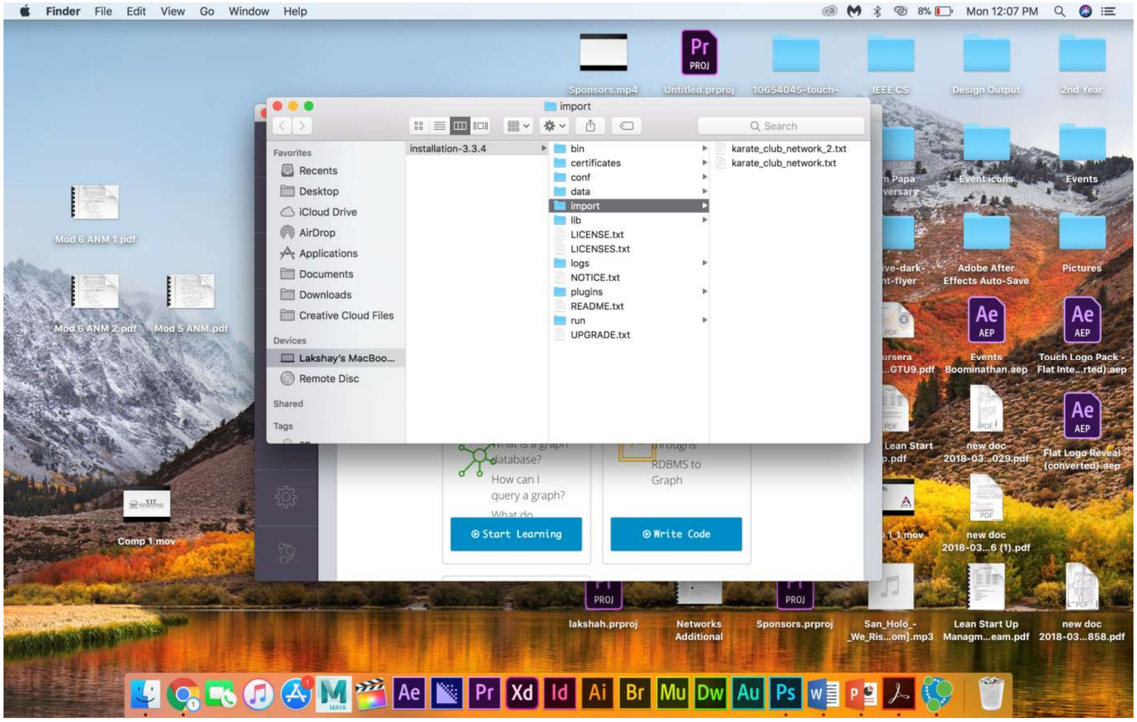
```
$ CREATE (n:Book{title:"Life of Shivam Khanna",author:"Mother of Shivam Khanna", `Oscar Award`:"Man Booker Price",`Edition Language`:…
```

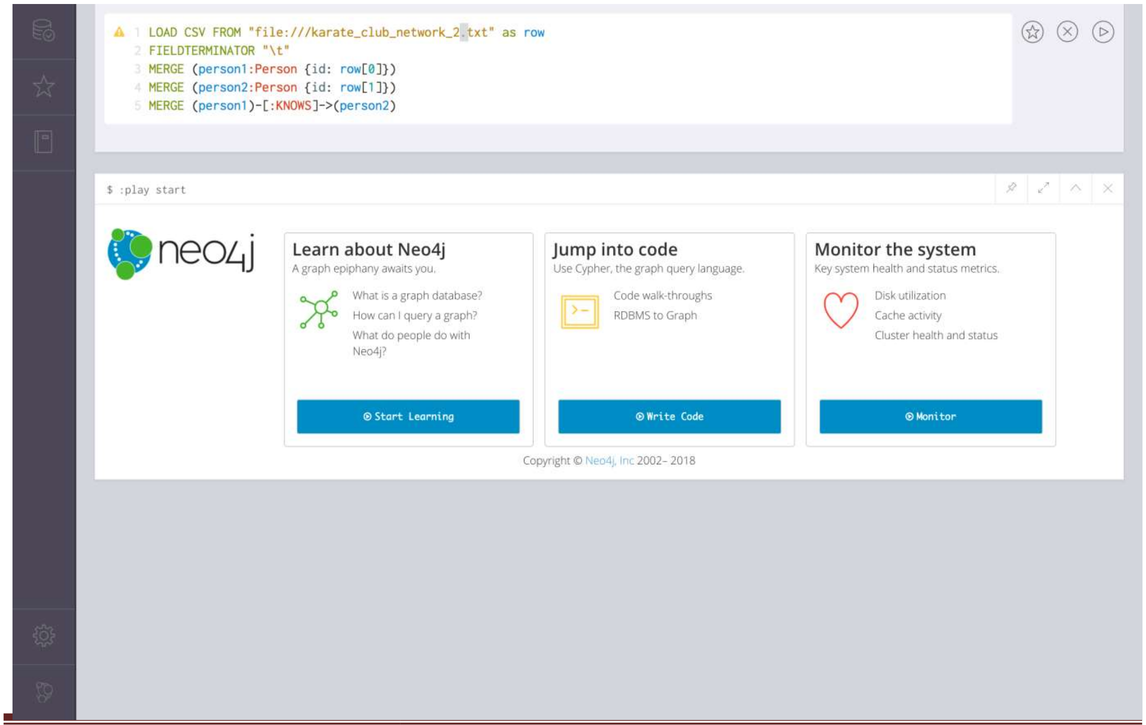
Book <id>: 153 **Edition Language:** English **Oscar Award:** Man Booker Price **author:** Mother of Shivam Khanna **title:** Life of Shivam Khanna

```
$ CREATE (n:Book{title:"Master Eater:Shivam Khanna",author:"Keshika Tank",Author:"Lakshay Nanda"}) RETURN n;
```

Tasks:-

1. SLM clustering

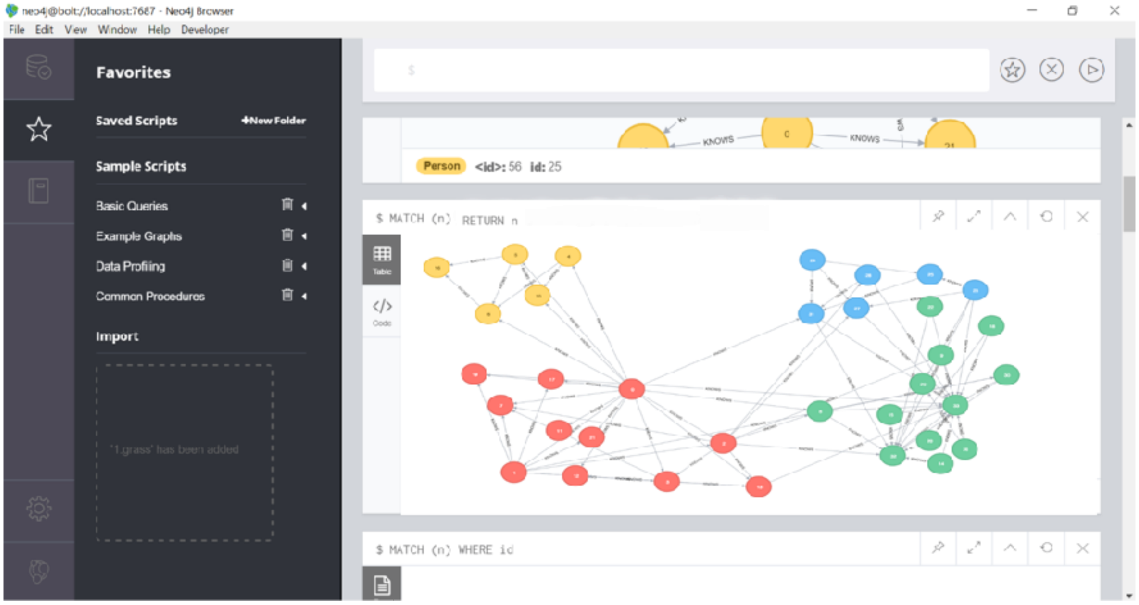




```
LOAD CSV FROM "file:///karate_club_network_2.txt" as row
FIELDTERMINATOR "\t"
MERGE (person1:Person {id: row[0]})
MERGE (person2:Person {id: row[1]})
MERGE (person1)-[:KNOWS]->(person2)
```

```
$ :play start
```

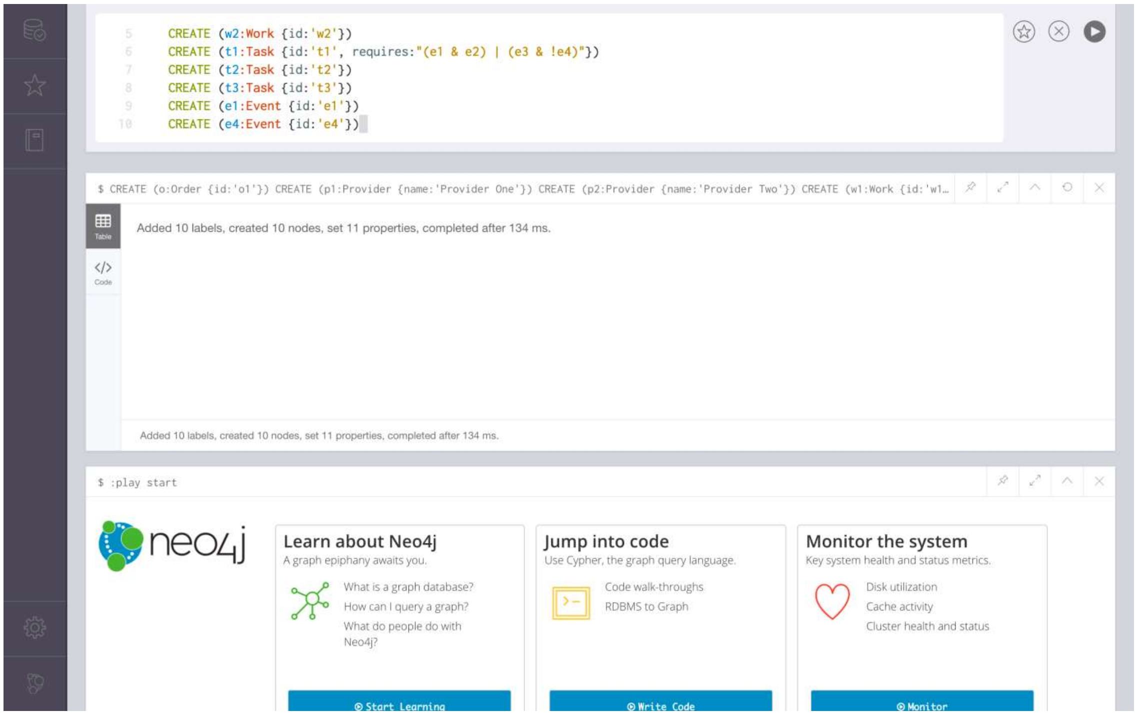
neo4j
Learn about Neo4j
A graph epiphany awaits you.
What is a graph database?
How can I query a graph?
What do people do with Neo4j?
Start Learning
Jump into code
Use Cypher, the graph query language.
Code walk-throughs
RDBMS to Graph
Write Code
Monitor the system
Key system health and status metrics.
Disk utilization
Cache activity
Cluster health and status
Monitor
Copyright © Neo4j, Inc 2002– 2018



```
$ MATCH (n) RETURN n
```

```
$ MATCH (n) WHERE id
```

2. Work order management



```
CREATE (w2:Work {id:'w2'})
CREATE (t1:Task {id:'t1', requires:"(e1 & e2) | (e3 & !e4)"})
CREATE (t2:Task {id:'t2'})
CREATE (t3:Task {id:'t3'})
CREATE (e1:Event {id:'e1'})
CREATE (e4:Event {id:'e4'})
```

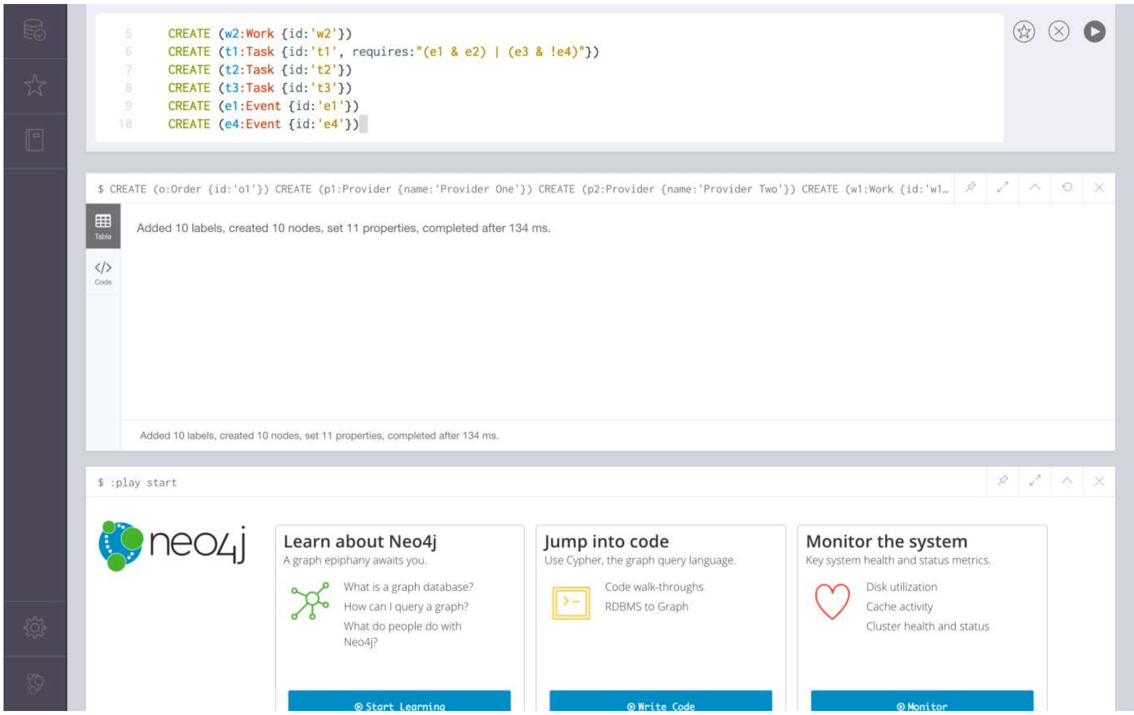
```
$ CREATE (o:Order {id:'o1'}) CREATE (p1:Provider {name:'Provider One'}) CREATE (p2:Provider {name:'Provider Two'}) CREATE (w1:Work {id:'w1…
```

Added 10 labels, created 10 nodes, set 11 properties, completed after 134 ms.

Added 10 labels, created 10 nodes, set 11 properties, completed after 134 ms.

```
$ :play start
```

Table
Code
neo4j
Learn about Neo4j
A graph epiphany awaits you.
What is a graph database?
How can I query a graph?
What do people do with Neo4j?
Start Learning
Jump into code
Use Cypher, the graph query language.
Code walk-throughs
RDBMS to Graph
Write Code
Monitor the system
Key system health and status metrics.
Disk utilization
Cache activity
Cluster health and status
Monitor



```
CREATE (w2:Work {id:'w2'})
CREATE (t1:Task {id:'t1', requires:"(e1 & e2) | (e3 & !e4)"})
CREATE (t2:Task {id:'t2'})
CREATE (t3:Task {id:'t3'})
CREATE (e1:Event {id:'e1'})
CREATE (e4:Event {id:'e4'})
```

```
$ CREATE (o:Order {id:'o1'}) CREATE (p1:Provider {name:'Provider One'}) CREATE (p2:Provider {name:'Provider Two'}) CREATE (w1:Work {id:'w1…
```

```
Added 10 labels, created 10 nodes, set 11 properties, completed after 134 ms.
```

```
Added 10 labels, created 10 nodes, set 11 properties, completed after 134 ms.
```

```
$ :play start
```

Table
Code
neo4j
Learn about Neo4j
A graph epiphany awaits you.
What is a graph database?
How can I query a graph?
What do people do with Neo4j?
Start Learning
Jump into code
Use Cypher, the graph query language.
Code walk-throughs
RDBMS to Graph
Write Code
Monitor the system
Key system health and status metrics.
Disk utilization
Cache activity
Cluster health and status
Monitor

THANK YOU!!