

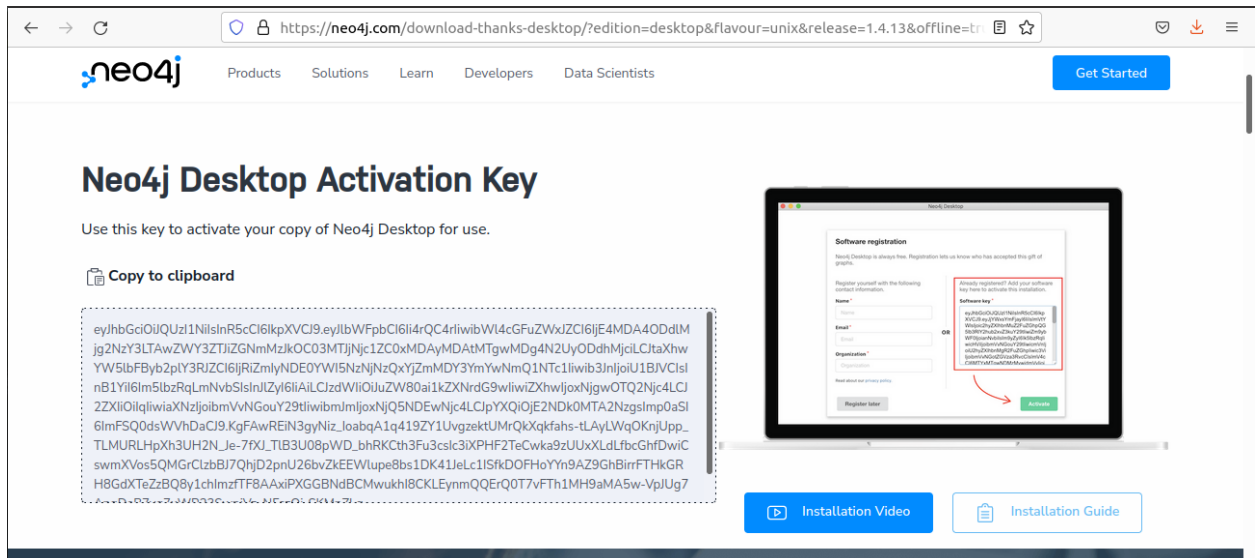
Experiment 7

Ojas Patil 2019130048,
Jeet Mistry 2020401071,
Sahil Chorghe 2020301073

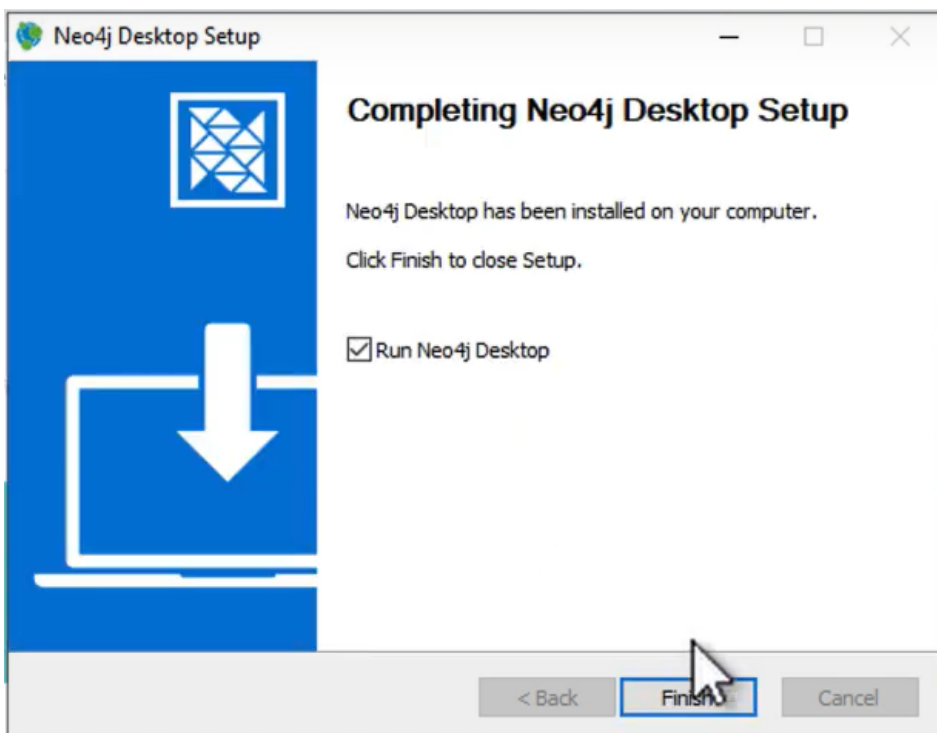
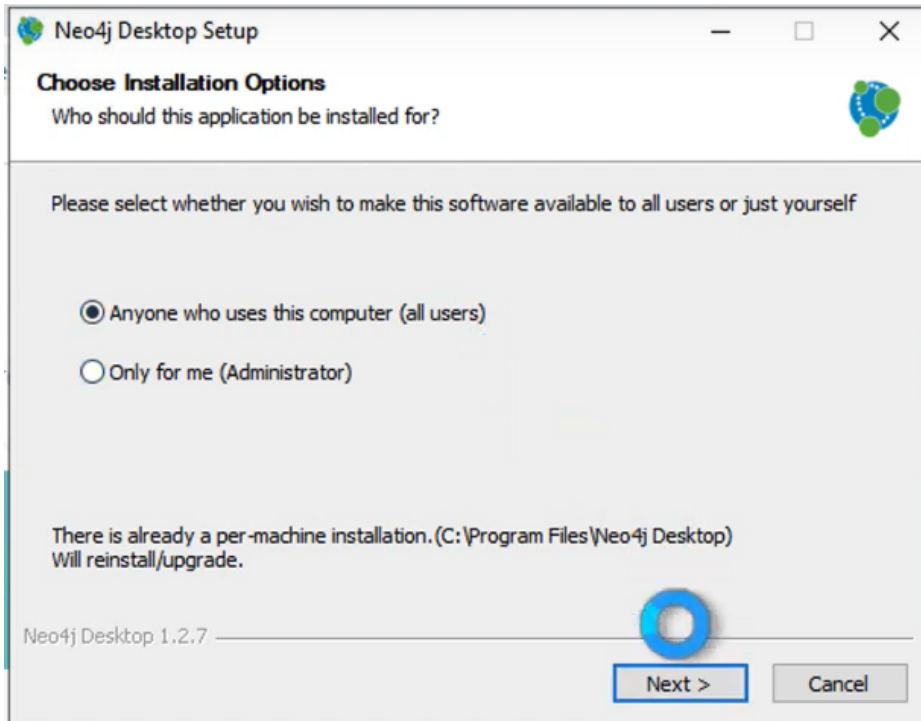
Aim: To demonstrate graph DB

Procedure:

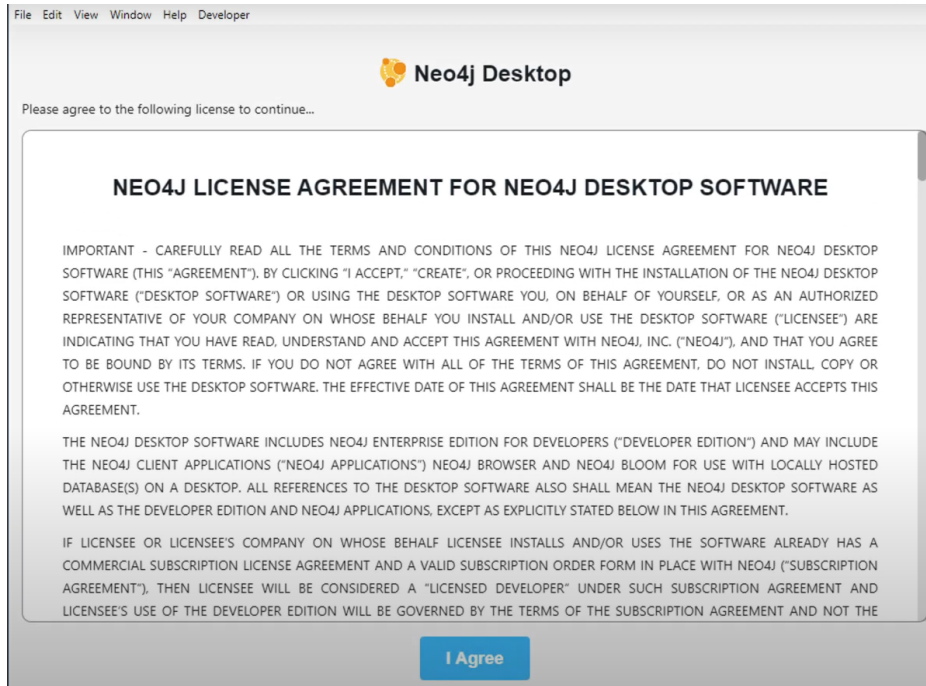
1. Download Neo4j from [here](#).
2. Being a first time user, you have to register yourself by providing appropriate details.
3. Copy and store the Activation code somewhere safely.



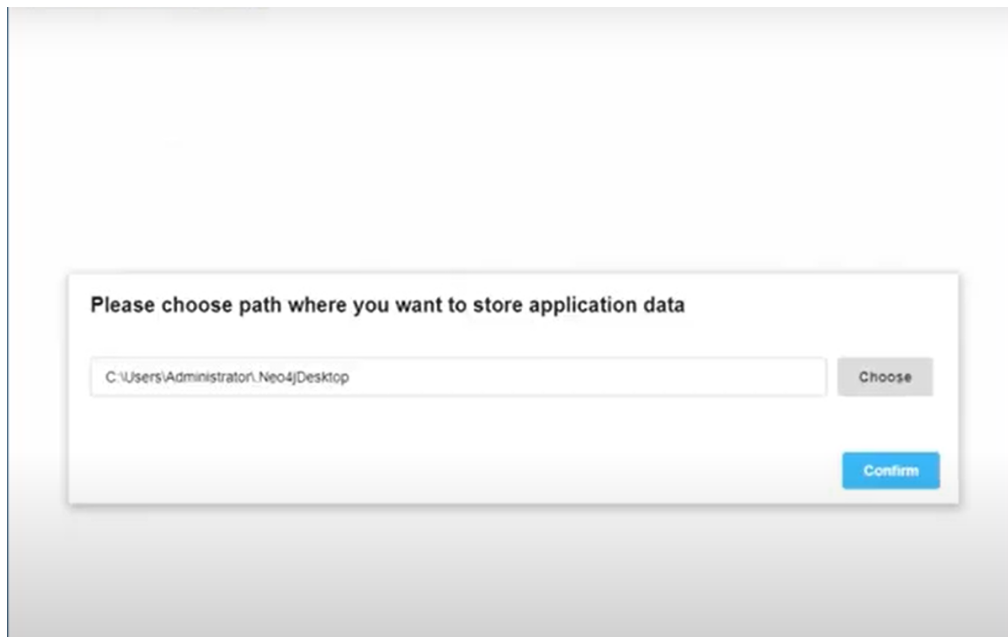
4. Now, run the downloaded setup as administrator, and select the option, “Anyone who uses this computer (all users)” and click next. Go on clicking ‘Next’ until you see the ‘Install’ button. Now click the ‘Install’ button and click finish on completion.



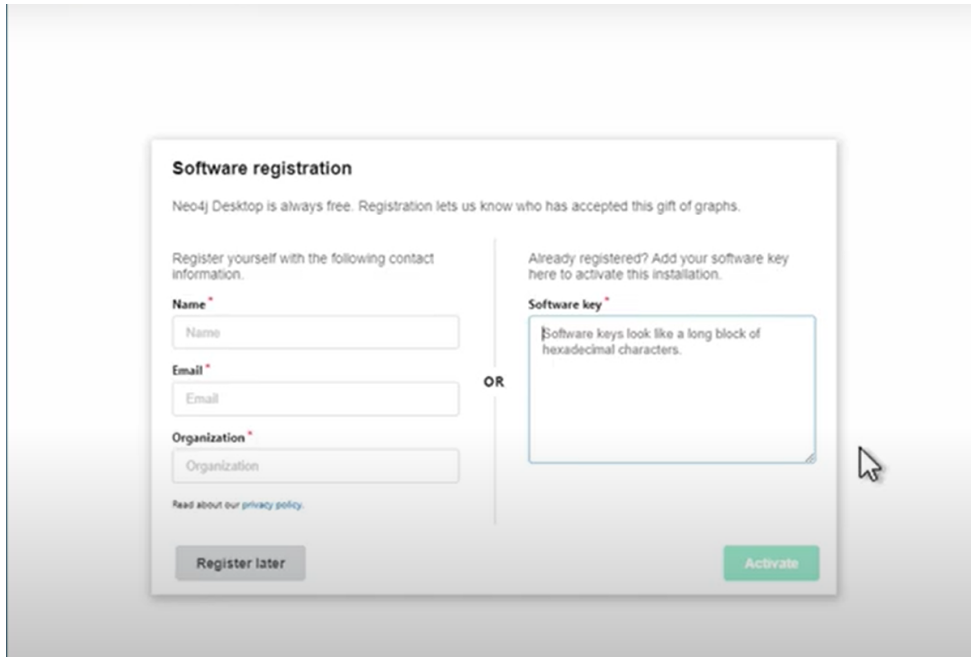
5. Accept the agreement



6. Click 'Confirm'



7. Enter the already copied software key here and press 'activate'



Software registration

Neo4j Desktop is always free. Registration lets us know who has accepted this gift of graphs.

Register yourself with the following contact information.

Name *

Email *

Organization *

[Read about our privacy policy.](#)

OR

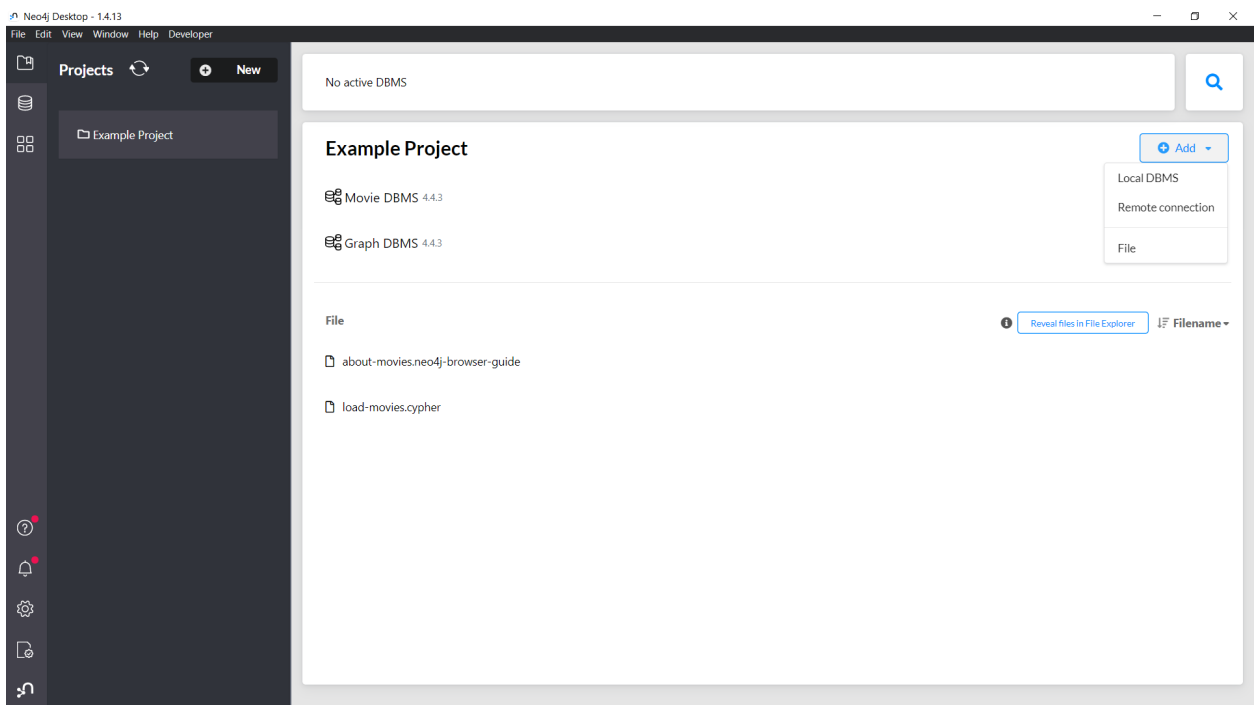
Already registered? Add your software key here to activate this installation.

Software key *

Software keys look like a long block of hexadecimal characters.

[Register later](#) [Activate](#)

8. Click 'Add' button and select 'Local DBMS'




9. Provide a suitable name and a strong password. Then click Create.

No active DBMS


Example Project

[+ Add](#)

Name


 Graph DBMS

Password

 password

Version


4.4.3



 Cancel [✓ Create](#)

10. Now click 'Start' on the newly created DBMS. Then, click Open


Example Project


[+ Add](#)

 Movie DBMS 4.4.3

 Graph DBMS 4.4.3 [Start](#) [↗ Open](#) 

This list of databases is cached, start the DBMS to refresh the list.

 system

 neo4j (default)

11. Write your command on the shell line given at the top of the window that just has appeared.

```
neo4j@bolt://localhost:7687/neo4j - Neo4j Browser
File Edit View Window Help Developer
neo4j$ CREATE (Student:Ojas{name:"Ojas", dept:"COMPS", since:2019}) RETURN Student;
```

Commands:

a) Create Node:

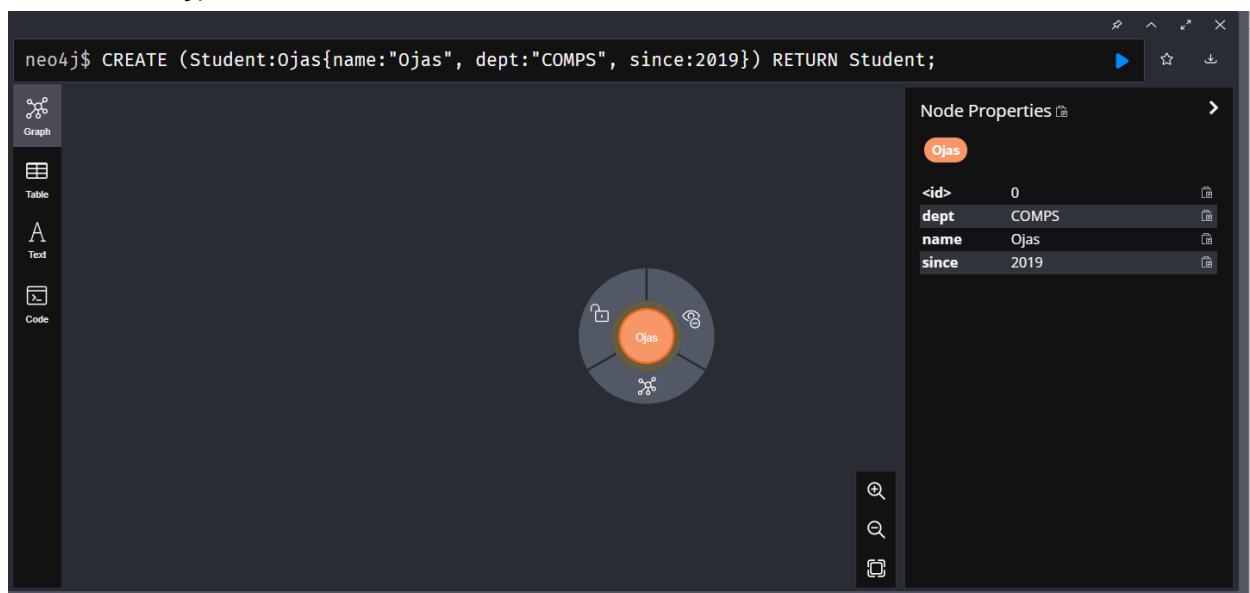
```
neo4j$ CREATE (Student:Ojas{name:"Ojas", dept:"COMPS", since:2019}) RETURN Student;
```

Syntax:

CREATE (Node:Label{*properties of the node in the key-value format*}) RETURN Node;

Example:

CREATE (Student:Ojas{name:"Ojas", dept:"COMPS", since:2019}) RETURN Student;



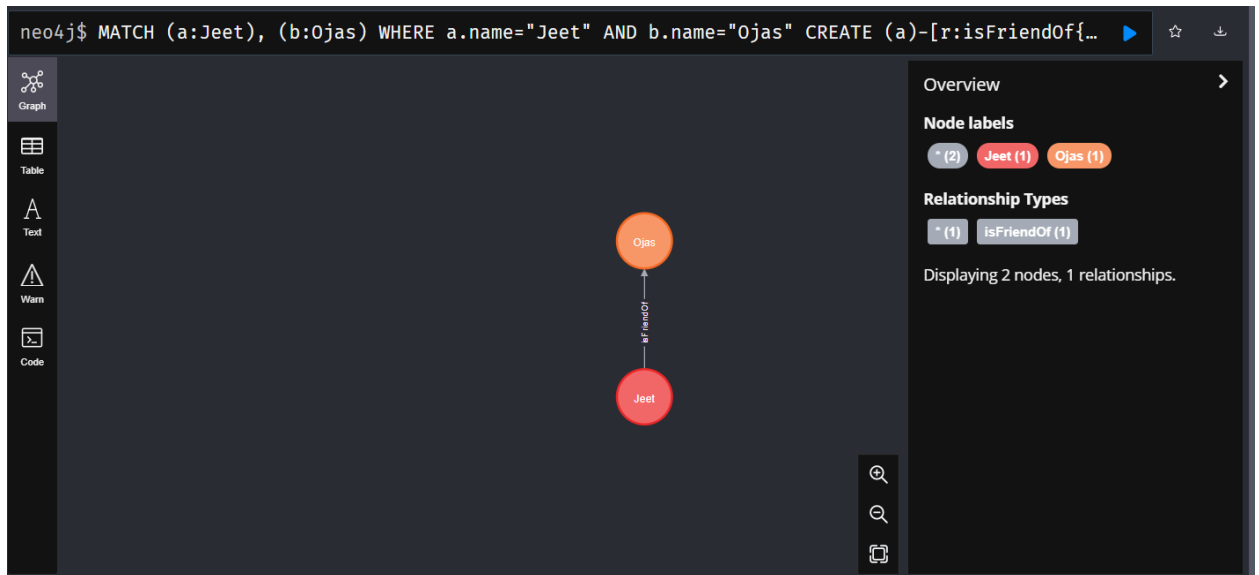
b) Create Relationship:

Syntax:

MATCH (a:label), (b:label) WHERE *conditions_if_any* CREATE
(a) - [r: isFriendOf{*properties of the relation in the key-value format*}]
-> (b) RETURN a,b;

Example:

MATCH (a:Jeet), (b:Ojas) WHERE a.name="Jeet" AND
a.name="Ojas" CREATE
(a)-[r:isFriendOf{knowsBcoz:"ADBMS"}]->(b) RETURN a,b;



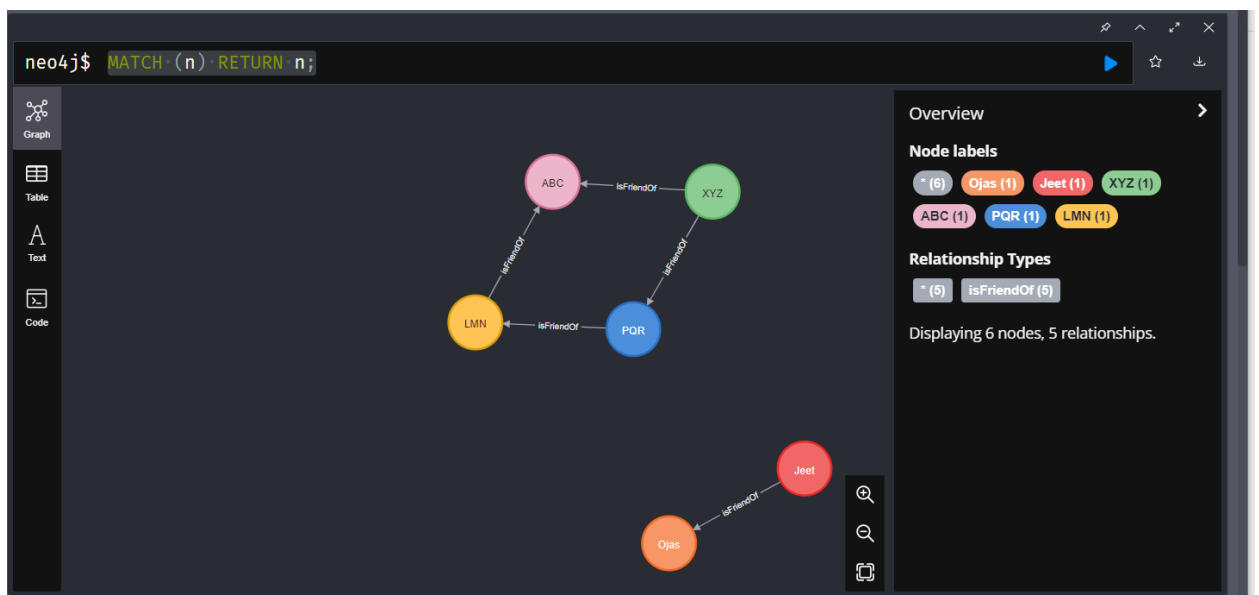
c) Select nodes/ relationships

Syntax:

`MATCH (n) WHERE conditions_if_any RETURN n`

Example:

`MATCH (n) RETURN n;`



d) Update node/ relationships

Syntax:

`MATCH (Node) SET Node.key=value RETURN Node;`

Example:

`MATCH (Teacher:KKD) SET Teacher.AreasOfInterest="OS, DBMS, CCN, Cloud Computing" RETURN Teacher;`

Before:

The screenshot shows the Neo4j web interface. The command bar at the top contains the query: `neo4j$ CREATE (Teacher:KKD{name:"Prof. Kailas Devadkar", dept:"IT", since:2003}) RETURN Teacher;`. The left sidebar has icons for Graph, Table, Text, and Code. The main graph area displays a single node labeled "Prof. Kailas Dev...". The right sidebar, titled "Node Properties", shows the properties for the selected node (KKD):

| Property | Value |
|----------|-----------------------|
| <id> | 4 |
| dept | IT |
| name | Prof. Kailas Devadkar |
| since | 2003 |

After:

The screenshot shows the Neo4j web interface after the update. The command bar at the top contains the query: `neo4j$ MATCH (Teacher:KKD) SET Teacher.AreasOfInterest="OS, DBMS, CCN, Cloud Computing" RETURN ...`. The left sidebar is the same. The main graph area displays the same node. The right sidebar, titled "Node Properties", shows the updated properties for the selected node (KKD):

| Property | Value |
|-----------------|--------------------------------|
| <id> | 4 |
| AreasOfInterest | OS, DBMS, CCN, Cloud Computing |
| dept | IT |
| name | Prof. Kailas Devadkar |
| since | 2003 |

This commands updates value of an existing key and creates a new key if it doesn't exist

e) Delete Node/ relationship

Syntax:

1. For detaching relationship

```
MATCH (Node1)-[r:relationship]->(Node2) DELETE r
```

2. For deleting node:

```
MATCH (Node:label) DETACH DELETE Node
```

Example:


1. For detaching relationship


```
MATCH (Raul)-[r:PLAYER_OF]->(It) DELETE r
```

2. For deleting node:

```
MATCH (Kohli:player) DETACH DELETE Kohli
```

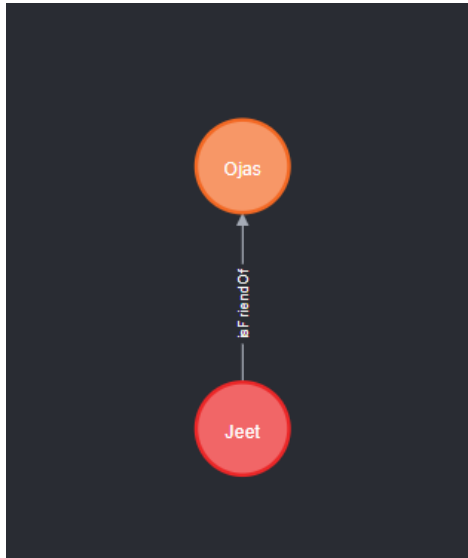
```
neo4j$ MATCH (Student:Ojas) DETACH DELETE Student;
```


Table

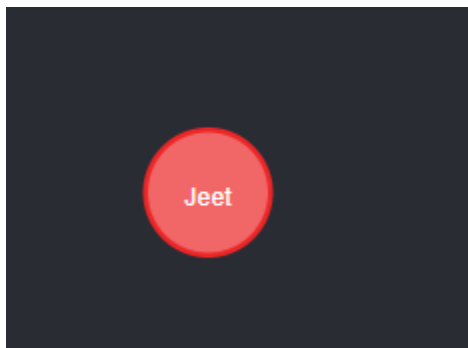

Code

Deleted 1 node, deleted 1 relationship, completed after 9 ms.

Before:



After:



Conclusion:

GraphDB is new and a fast alternative to existing RDBMS. GraphDB has given rise to new areas. It is ideal for highly associated data. Doesn't work well in case of distributed databases. Switching to GraphDB is the requirement of the time.