# DBMS LAB

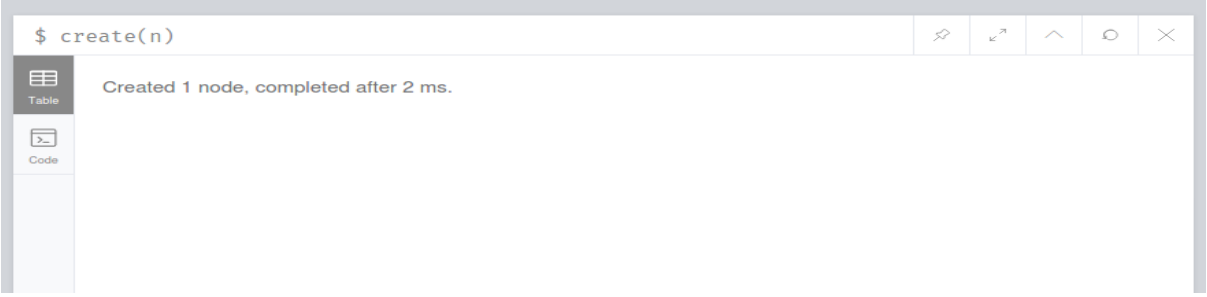
## Week 2

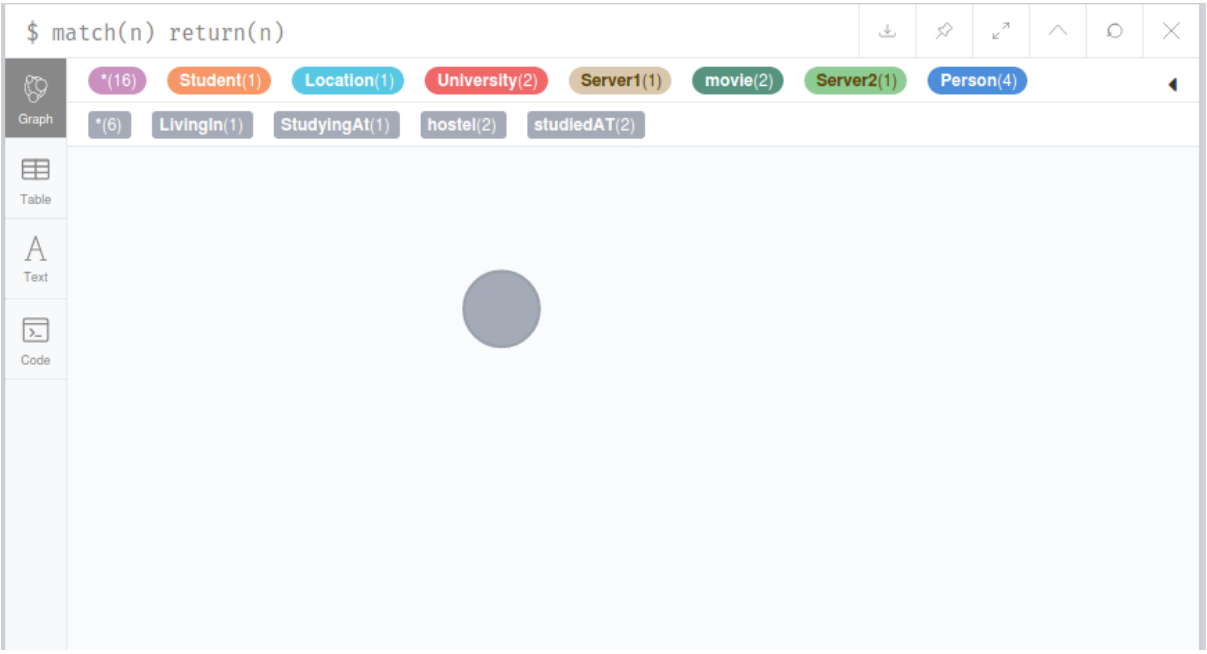
Name : Abhishek Aditya BS
SRN : PES1UG19CS019
Section : A
Semester : 5th

## 1. Create node and relationships

### 1.1 Create a single node :

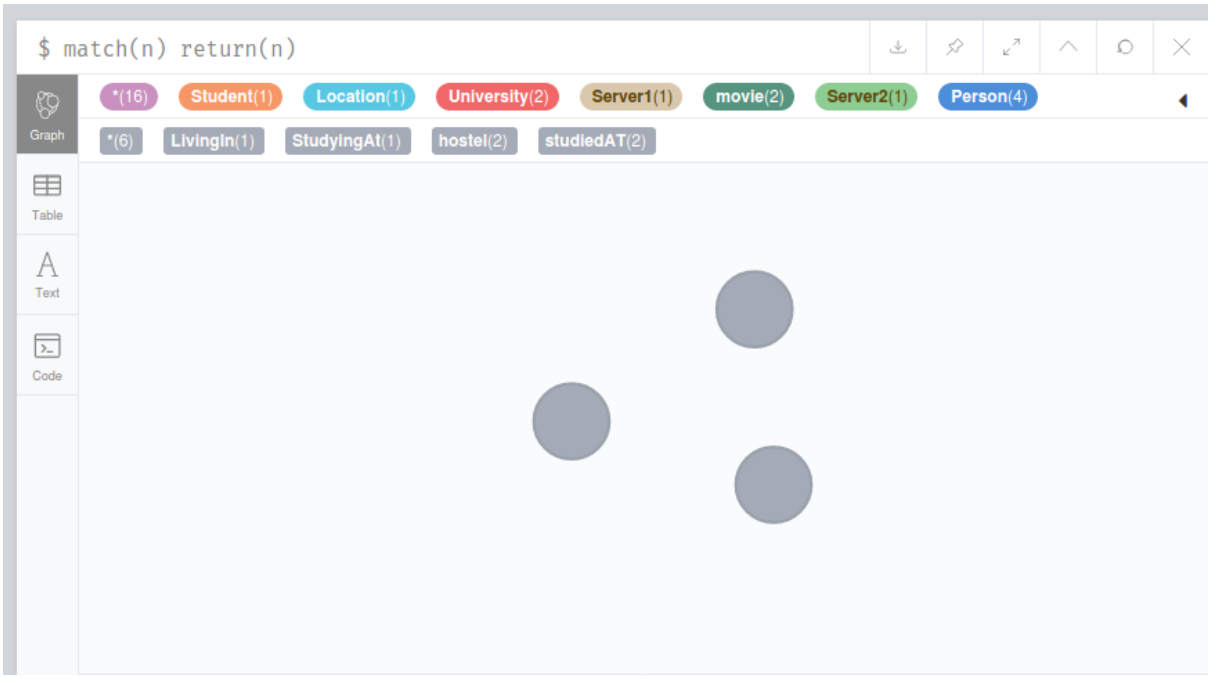
**Syntax: create (n)**



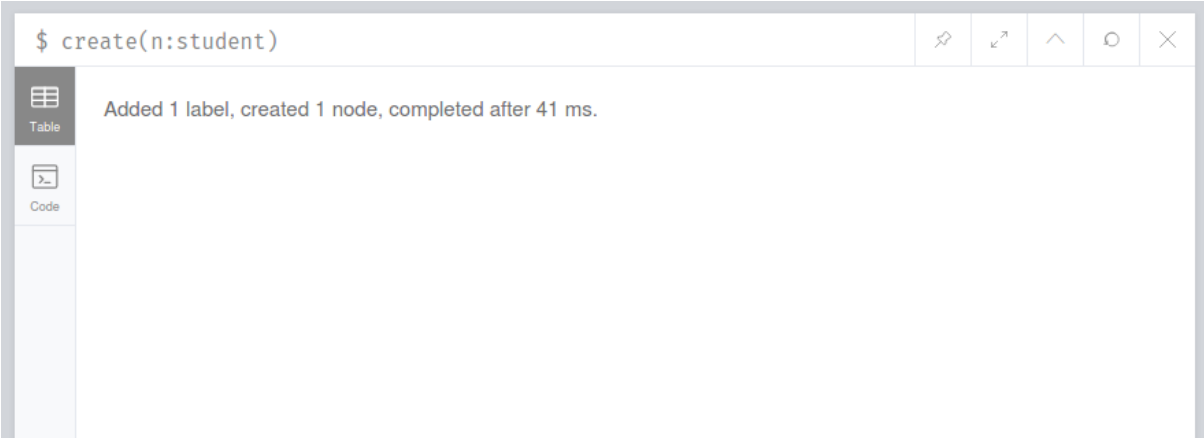
## 1.2 Create multiple nodes

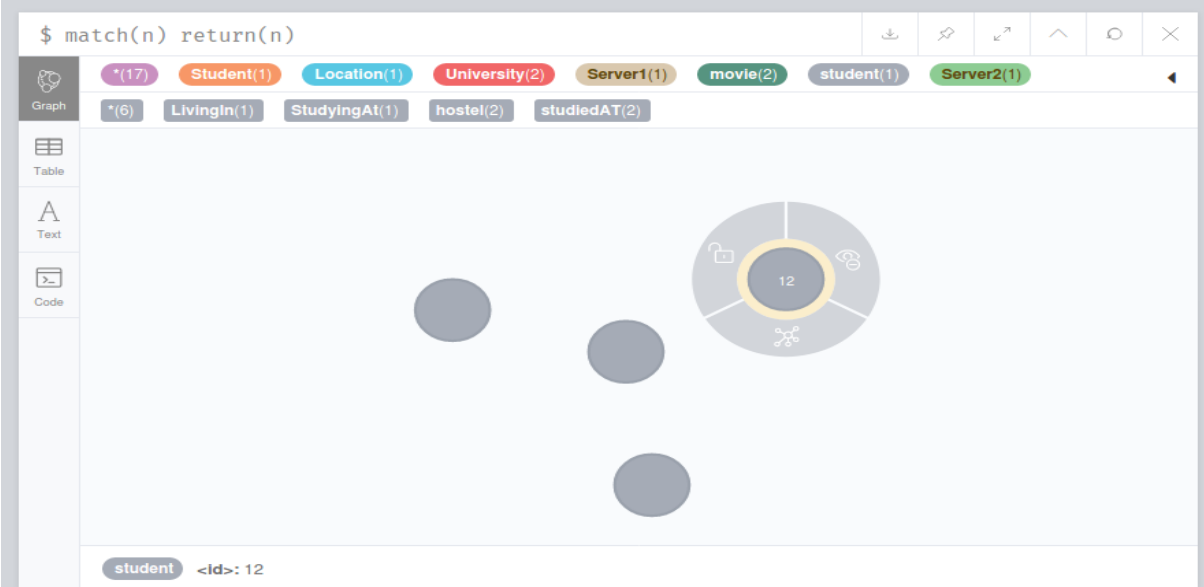
**Syntax: create(n),(m)**


## 1.3 Create a node with a label

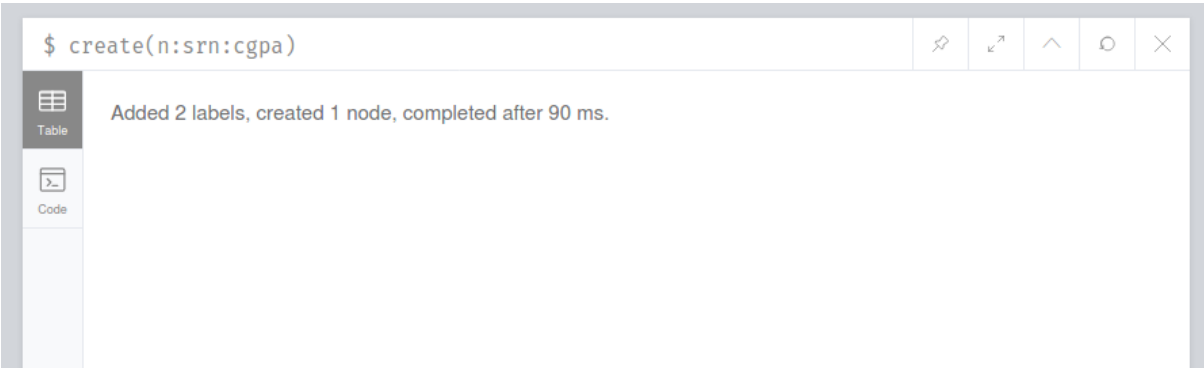
**Syntax: create(n:label name)**

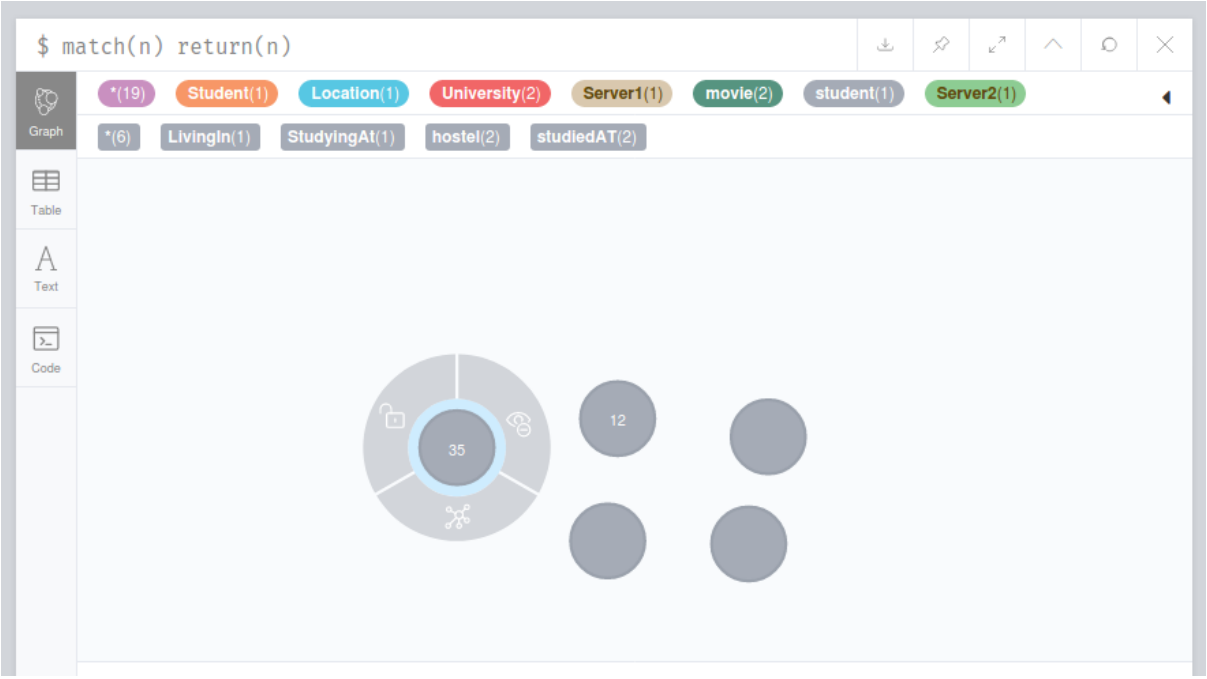



# 1.4 Create a node with multiple labels

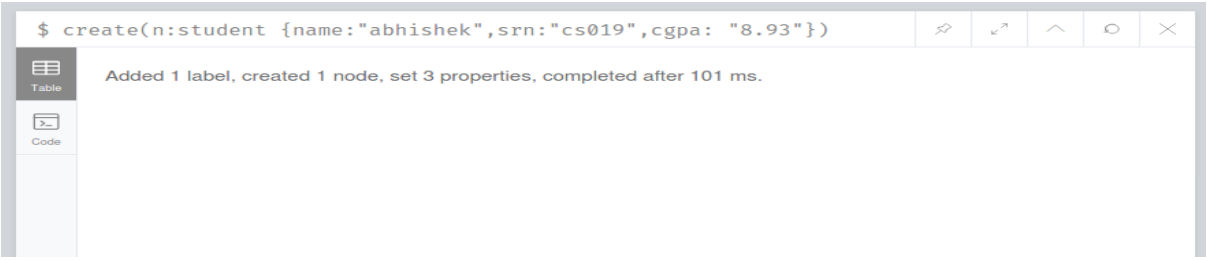
## Syntax: CREATE (n:label1:label2)

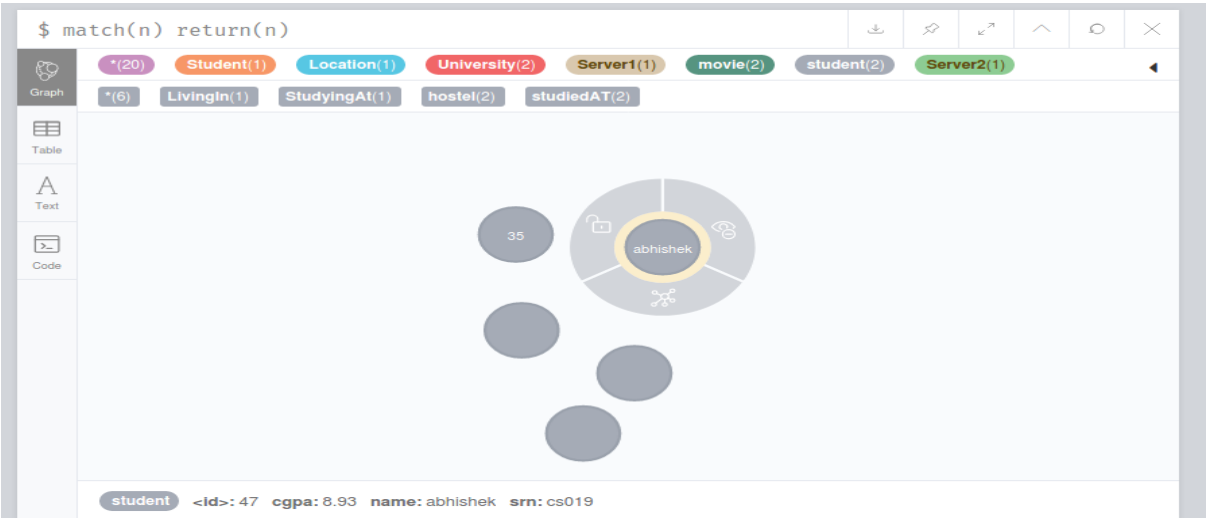





## 1.5 Create node and add labels and properties

**Syntax: Create(n:lablename {properties and values})**



## 1.6 Create nodes with parameters as properties

**Syntax: Define the property with parameter name.Add the parameter using the create clause**

**CREATE (n:Person $props) RETURN n**

```
$ :params {"props":{"name": "Vishal",srn:"cs571",cgpa:"9.0"}}
```

```
{
  "props": {
    "name": "Vishal",
    "srn": "cs571",
    "cgpa": "9.0"
  }
}
```

See `:help param` for usage of the `:param` command.

Successfully set your parameters.

```
$ create(n:student $props) return(n)
```

Graph | Table | Text | Code

*(1) student(1)

Vishal

student <id>: 36 cgpa: 9.0 name: Vishal srn: cs571

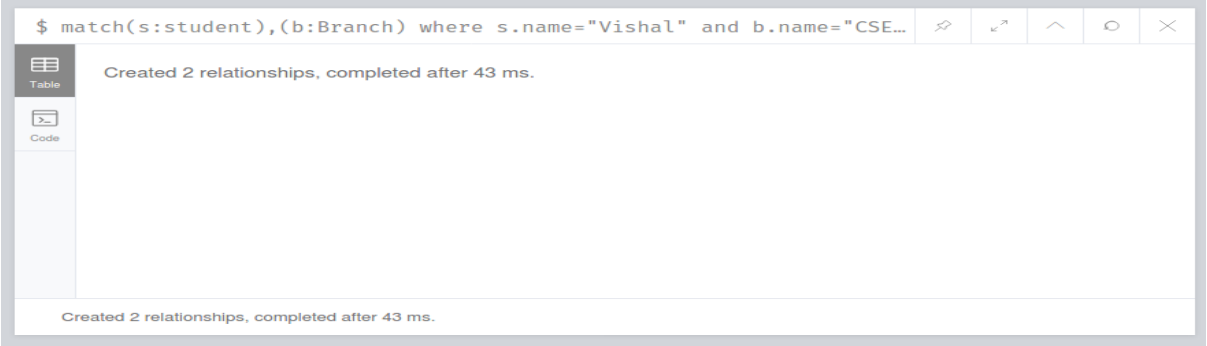
## 2. Create Relationships between the nodes

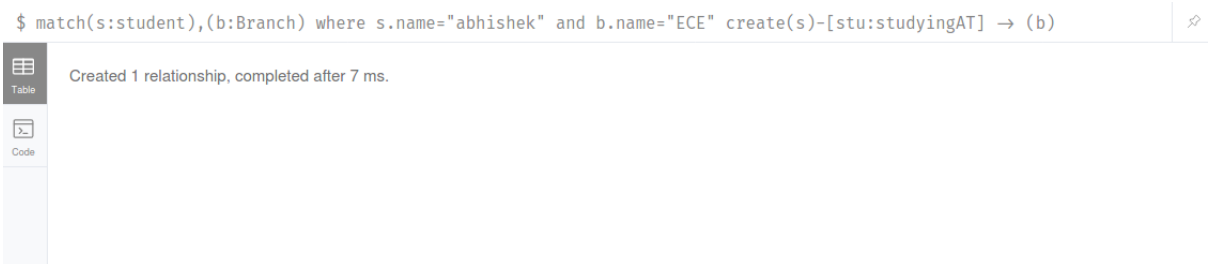
**Syntax:**

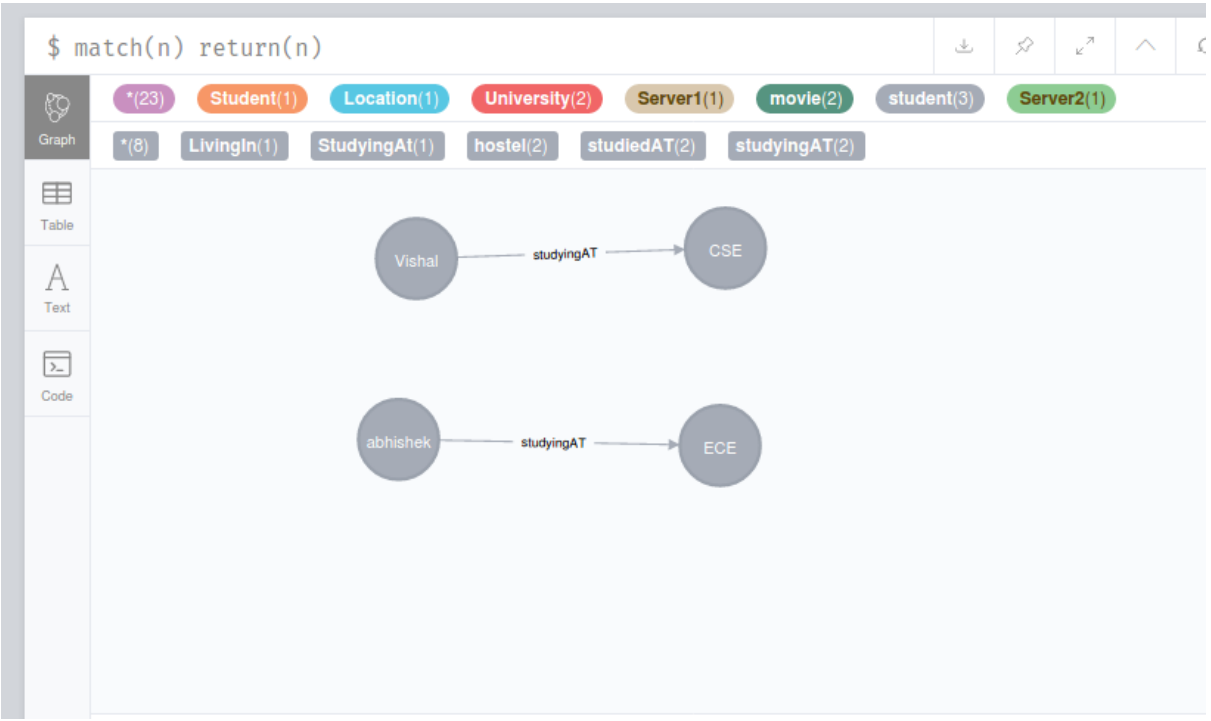
**Match (node1) ,(node2)**

**Where condition**

**Create (node1) [relation type ] ->(node2)**



```
$ match(s:student),(c:student) where s.name="Vishal" and c.name="abhishek" create(s)-[stu:friendOf] → (c)
```

Table

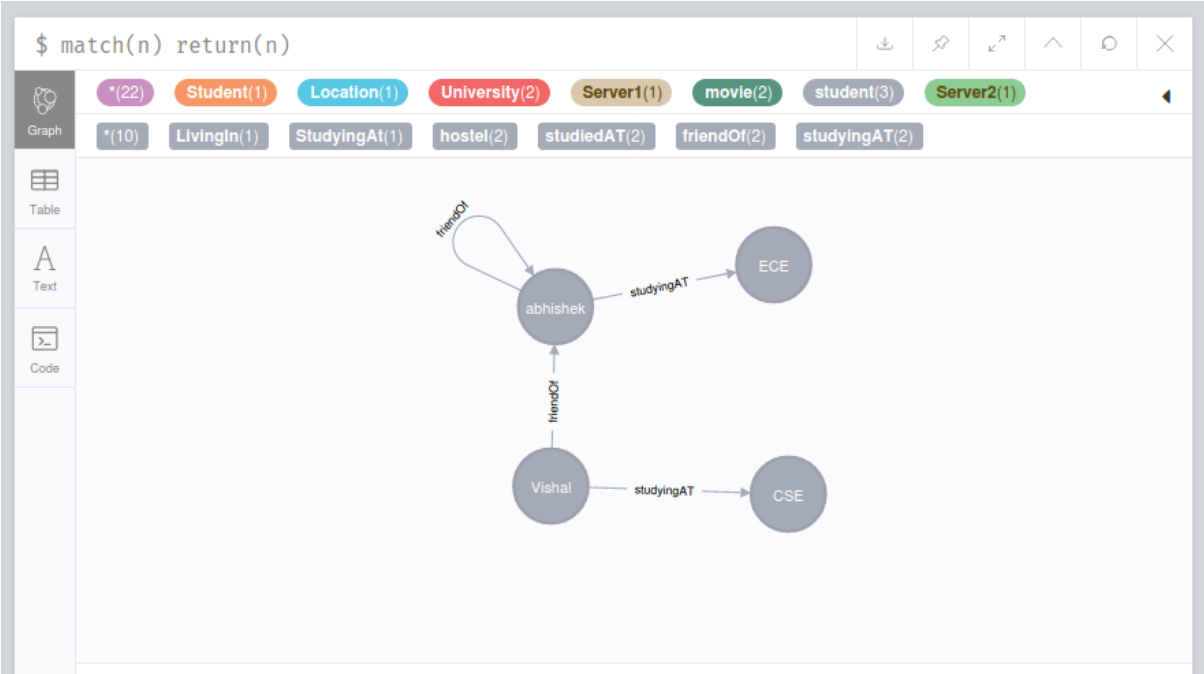
Code

Created 2 relationships, completed after 5 ms.

# 3. Read nodes and attributes (Node finding)

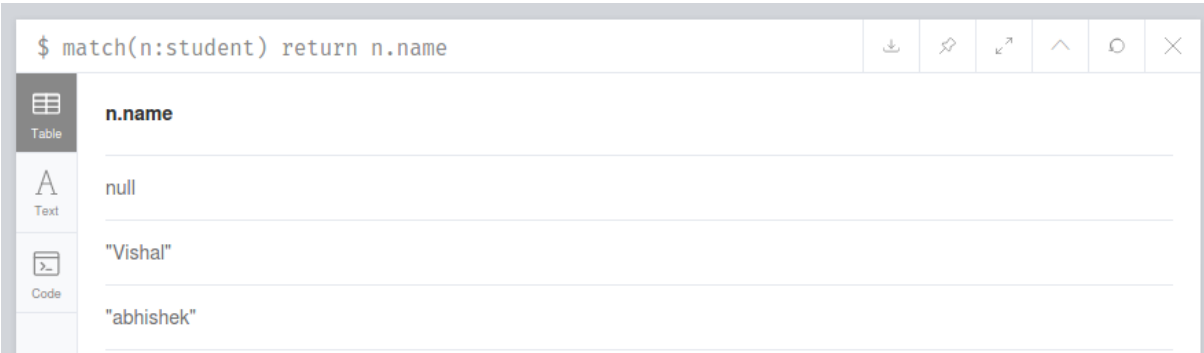
## 3.1 Get all nodes

## Syntax: Match(n) return(n)




## 3.2 Get all nodes with a label

Eg: MATCH (movie:Movie) RETURN movie.title



## 3.3 Related nodes

Eg: MATCH (director {name: 'Oliver Stone'})--(movie) RETURN movie.title

```
$ match(student {name: 'Vishal'})--(Branch) return Branch.year
```

| Branch.year |
| --- |
| null |
| "3" |

## 4 Update or set a value

**Syntax: MATCH (n:Node)**

**Set n. propertyvalue = 'newvalue'**

```
$ match(n:student {name:"abhishek"}) set n.cgpa=10
```

Set 1 property, completed after 3 ms.

```
$ match(student {name: 'abhishek'}) return student.cgpa
```

| student.cgpa |
| --- |
| 10 |

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

# 5. Delete operation

## 5.1 Delete the relationship

```
$ match(n:student {name:'Vishal'})-[r:studyingAT]→() delete r
```

Table

Code

Deleted 2 relationships, completed after 3 ms.

Deleted 2 relationships, completed after 3 ms.

```
$ match(n:student {name:'abhishek'})-[r:studyingAT]→() delete r
```

Table

Code

Deleted 1 relationship, completed after 3 ms.

```
$ match(n:student {name:'abhishek'})-[r:friendOf]→() delete r
```

Table

Code

Deleted 1 relationship, completed after 5 ms.

```
$ match(n) return(n)
```

Graph

Table

Text

Code

*(23) Student(1) Location(1) University(2) Server1(1)

*(7) LivingIn(1) StudyingAt(1) hostel(2) studiedAT(2) frien

## 5.2 Delete the node

**Syntax: Match(n) delete (n)**