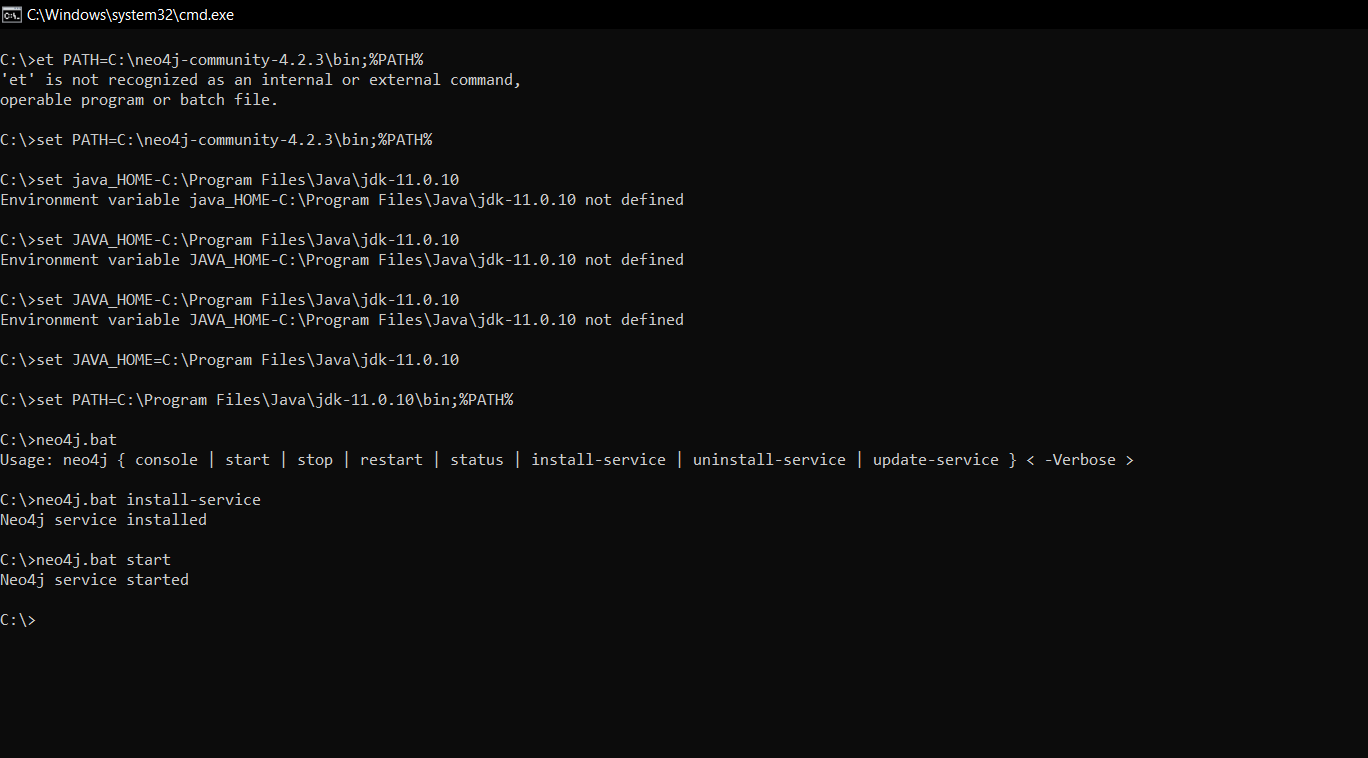
**AIM: WORKING WITH Neo4j DATABASE**

* **What is Neo4j?**

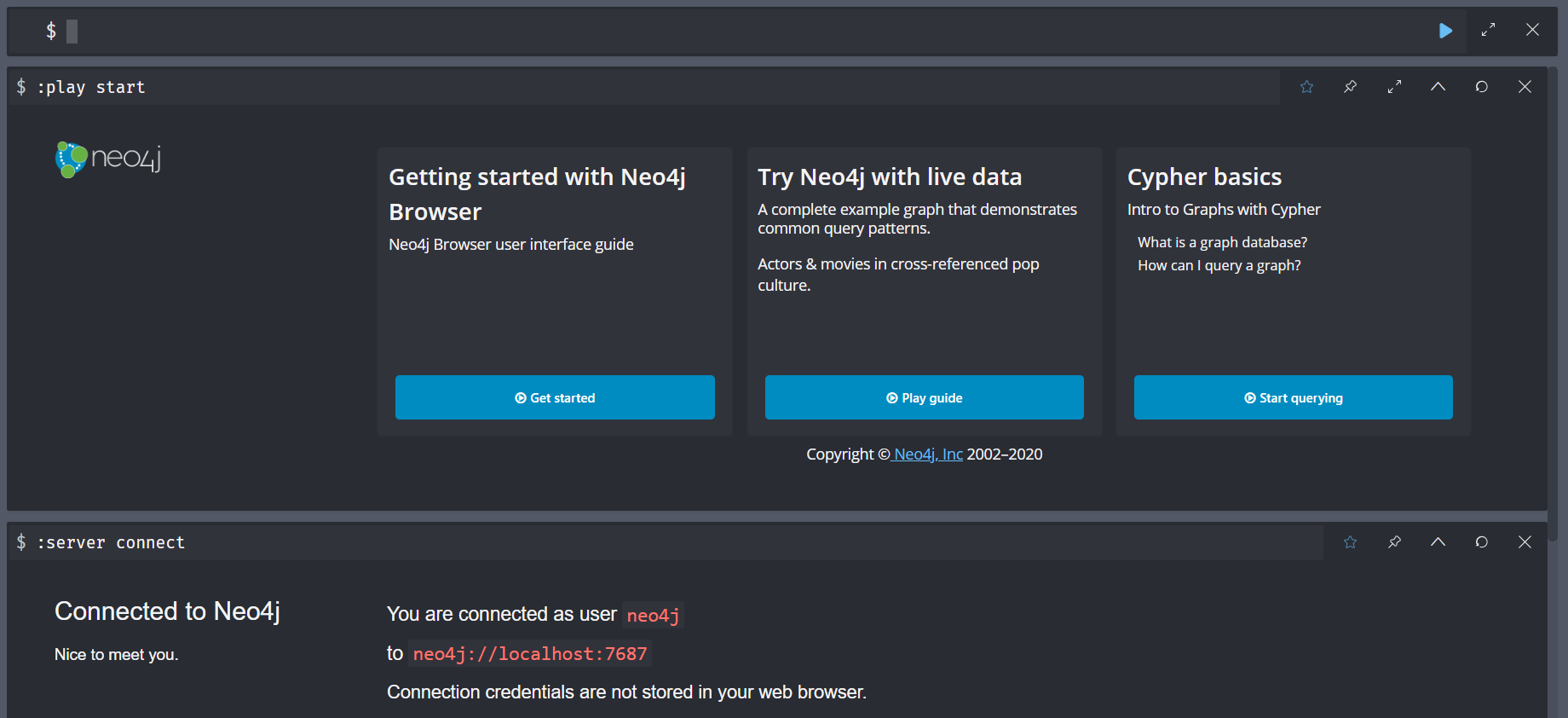
Neo4j is a graph database management system that stores and manages data as nodes, edges, and properties.

* **Uses**
* Neo4j is used in a wide range of applications, such as social networking, recommendation engines, fraud detection, and supply chain management.
* It is particularly useful for applications that involve complex relationships between data points.
* **Pros:**
* Neo4j's strength is its ability to query complex relationships and quickly traverse large datasets.
* It is a highly scalable database system and can handle large amounts of data with ease.
* Its query language, Cypher, is easy to learn and use.
* **Cons:**
* Neo4j can be more complex to set up and manage compared to traditional relational databases.
* It may not be suitable for applications that require heavy write operations.
* **Installation:**
* set NEO4J\_HOME=C:\neo4j-community-3.5.15
* set PATH=C:\neo4j-community-3.5.15\bin;%PATH%
* set NEO4J\_HOME= C:\Program Files\Java\jdk-11.0.10
* set PATH= C:\Program Files\Java\jdk-11.0.10\bin;%PATH%
* neo4j.bat install-service
* neo4j.bat start



Type in browser:

[**http://localhost:7474**](http://localhost:7474)

****

1. **Create 7 nodes in people label having properties name, department, address, age.**

CREATE (:people {name: "Guru", department: "IT", address: "1234 Main Street, Mumbai, India", age: 22}),

(:people {name: "Ansh", department: "CE", address: "5678 Second Street, New Delhi, India", age: 20}),

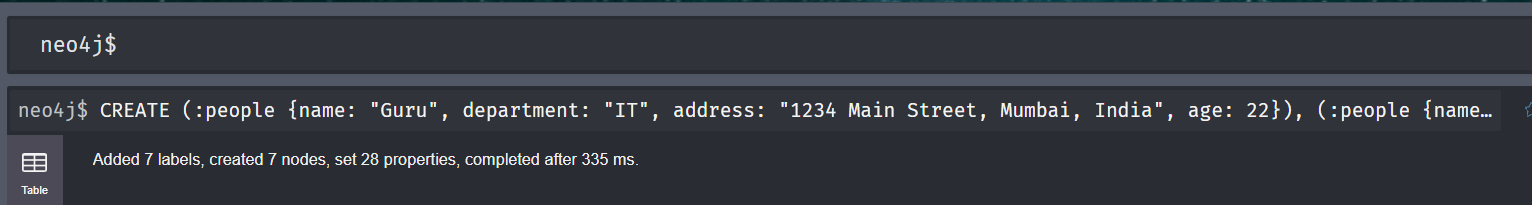
(:people {name: "Aman", department: "IT", address: "910 Third Street, Bangalore, India", age: 21}),

(:people {name: "Dhruv", department: "CE", address: "1212 Fourth Street, Kolkata, India", age: 19}),

(:people {name: "Aryan", department: "IT", address: "3434 Fifth Street, Pune, India", age: 20}),

(:people {name: "Avi", department: "CE", address: "5656 Sixth Street, Chennai, India", age: 20}),

(:people {name: "Het", department: "IT", address: "7878 Seventh Street, Hyderabad, India", age: 21})



1. **create 7 nodes in film label having properties title, releasedyear,directorname.**

CREATE (:film {title: "3 Idiots", releasedyear: 2009, directorname: "Rajkumar Hirani"}),

(:film {title: "Lagaan", releasedyear: 2001, directorname: "Ashutosh Gowariker"}),

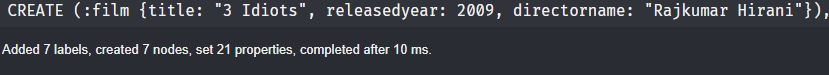
(:film {title: "Taare Zameen Par", releasedyear: 2007, directorname: "Aamir Khan"}),

(:film {title: "Dilwale Dulhania Le Jayenge", releasedyear: 1995, directorname: "Aditya Chopra"}),

(:film {title: "Kabhi Khushi Kabhie Gham", releasedyear: 2001, directorname: "Karan Johar"}),

(:film {title: "PK", releasedyear: 2014, directorname: "Rajkumar Hirani"}),

(:film {title: "Bahubali: The Beginning", releasedyear: 2015, directorname: "S. S. Rajamouli"})

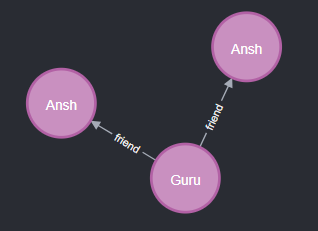


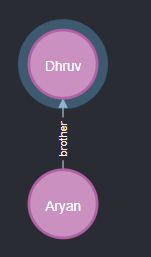
1. **Create relationships called friend, brother, and favourite nodes.**

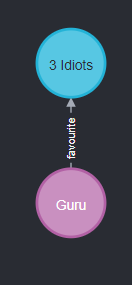
match (a088:people {name:"Guru"}),(b088:people {name:"Ansh"}) merge (a088)-[r:friend]->(b088) return a088,b088,r

MATCH (a088:people {name: "Aryan"}), (b088:people {name: "Dhruv"})MERGE (a088)-[r:brother]->(b088)RETURN a088, b088, r

MATCH (a088:people {name: "Guru"}), (b088:film {title: "3 Idiots"})MERGE (a088)-[r:favourite]->(b088)RETURN a088, b088, r





****

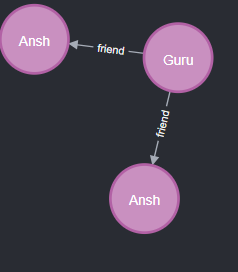
1. **What are the favourite movies of Aman**

MATCH (a088:people {name: "Guru"})-[r:favourite]->(b088:film)RETURN b088,a088,r



1. **Who are friends of Aman**

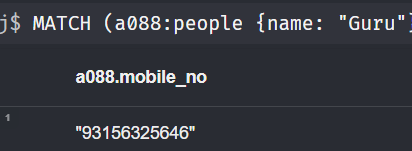
MATCH (a088:people {name: "Guru"})-[r:friend]-(b088:people)RETURN b088,a088,r

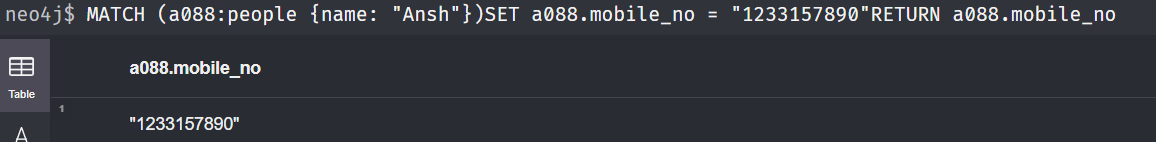


1. **Add a property called mobile\_no for person Aman and Ansh**

MATCH (a088:people {name: "Guru"})SET a088.mobile\_no = "93156325646"RETURN a088.mobile\_no

MATCH (a088:people {name: "Ansh"})SET a088.mobile\_no = "1233157890"RETURN a088.mobile\_no





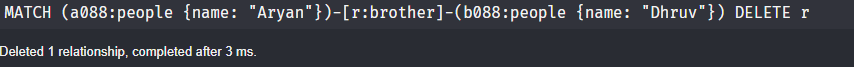
1. **Remove mobile \_ no property for person Aman**

MATCH (a088:people {name: "Guru"})REMOVE a088.mobile\_no RETURN a088



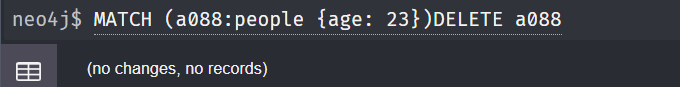
1. **Delete a relationship called brother**

MATCH (a088:people {name: "Aryan"})-[r:brother]-(b088:people {name: "Dhruv"})DELETE r



1. **Delete a node having age 23**

MATCH (a088:people {age: 23})DELETE a088



1. **Display details of node whose name is Avi and age is 23**

MATCH (a088:people {name: "Avi", age: 20})RETURN a088



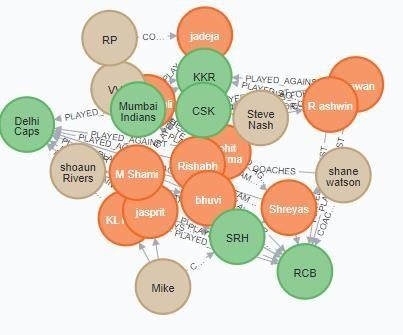
1. **display details of node having name Avi or Aman**

match (a088:people) where a088.name="Avi" or a088.name="Guru" return a088



Exercise -2

MATCH (n) RETURN n

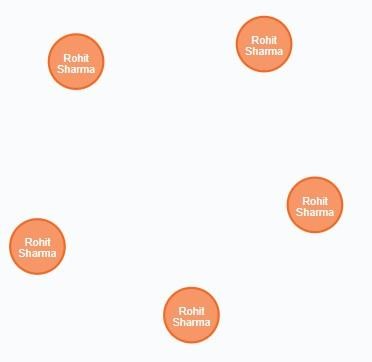


MATCH (player:PLAYER) RETURN player.name, player.height



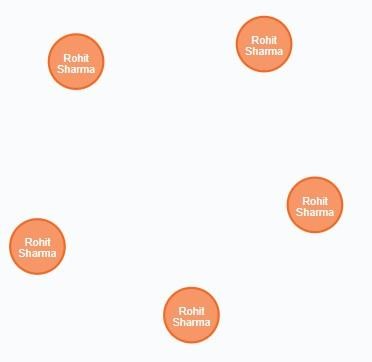
Q1))// Nodes where name is Rohit Sharma //

MATCH (player:PLAYER) WHERE player.name = "Rohit Sharma" RETURN player



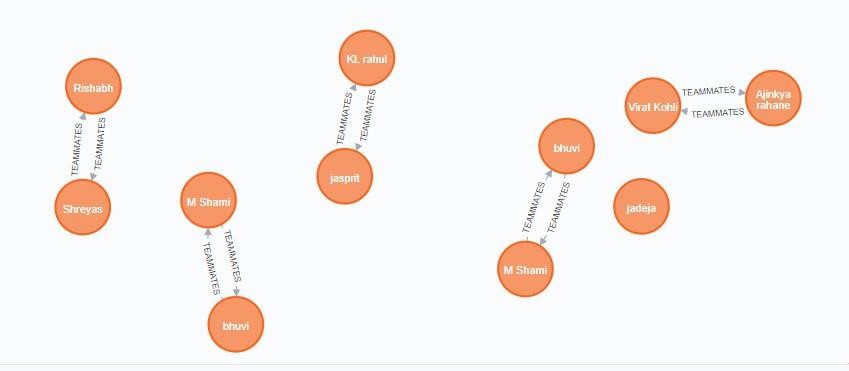
Q2)// Nodes where name is Rohit Sharma //

MATCH (player:PLAYER {name: "Rohit Sharma"}) RETURN player



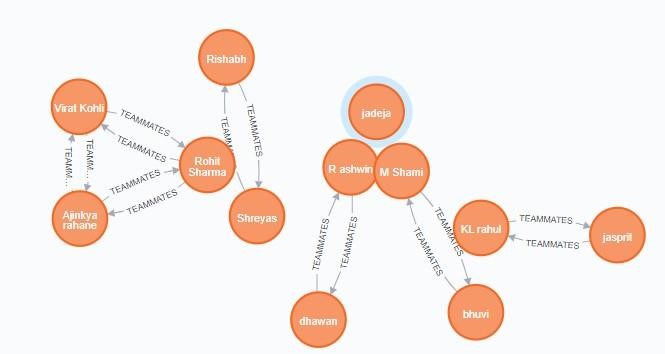
Q3)// Nodes where name is not Rohit Sharma

MATCH (player:PLAYER) WHERE player.name <> "Rohit Sharma"RETURN player



Nodes where height is greater than or equal to 2

Match(n:PLAYER) WHERE n.height>=2 return n



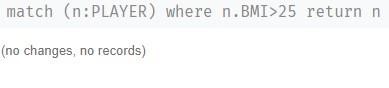
Q5// Nodes where height is less than 2

Match(n:PLAYER) WHERE n.height<2 return n



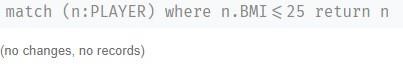
Q6// Nodes with a BMI larger than 25

Match(n:PLAYER) where n.BMI>25 return n



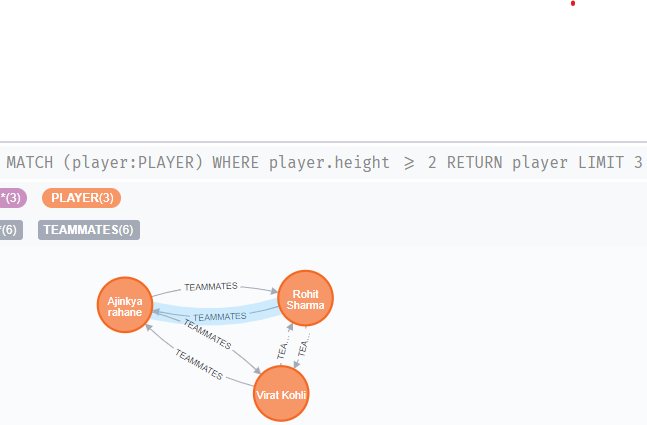
Q7// Nodes with a BMI not larger than 25

Match(n:PLAYER) where n.BMI<= 25 return n



// Limit

MATCH (player:PLAYER) WHERE player.height >= 2RETURN player LIMIT 3



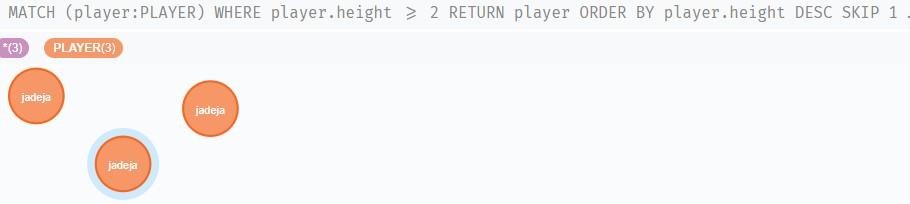
// Skip

MATCH (player:PLAYER) WHERE player.height >= 2RETURN playerSKIP 1 LIMIT 3



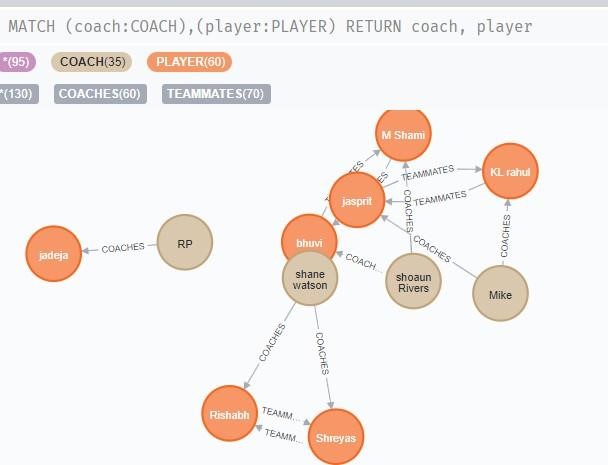
// Orderby

MATCH (player:PLAYER) WHERE player.height >= 2RETURN playerSKIP 1 ORDER BY player.height DESC LIMIT 3



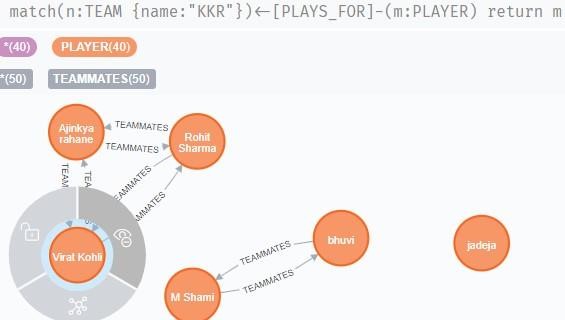
// Query for multiple nodes

MATCH (coach:COACH), (player:PLAYER) RETURN coach, player



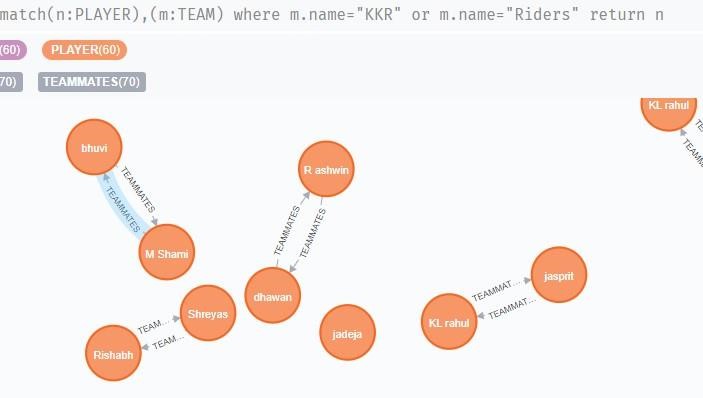
// GET ALL KKR PLAYERS //

Match(n:TEAM {name:”KKR”}<-[P LAYS\_FOR]-(m:PLAYER) return m

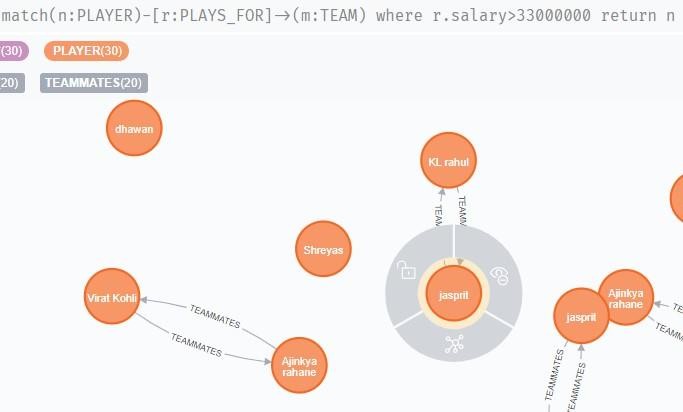


// GET ALL KKR OR royal PLAYERS //

Match(N:PLAYER),(m:TEAM) where m.name=”KKR” or m.name=”Riders” return n



// GET ALL PLAYERS THAT MAKE MORE THE 35M // Mathc(n:PLAYER)[r:PLAYS\_FOR]->(m:TEAM) where r.salary>33000000 return n

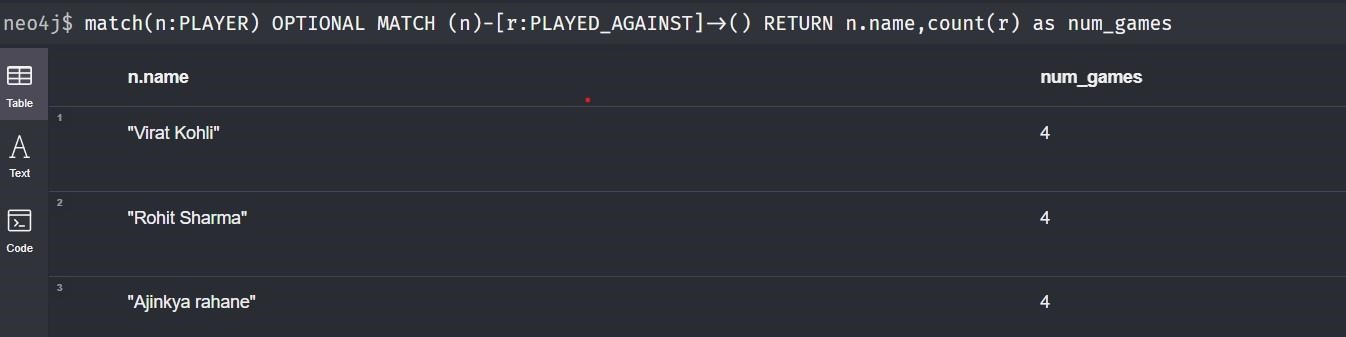


// GET ALL OF viratS TEAMMATES THAT MAKE MORE THAN 40M //

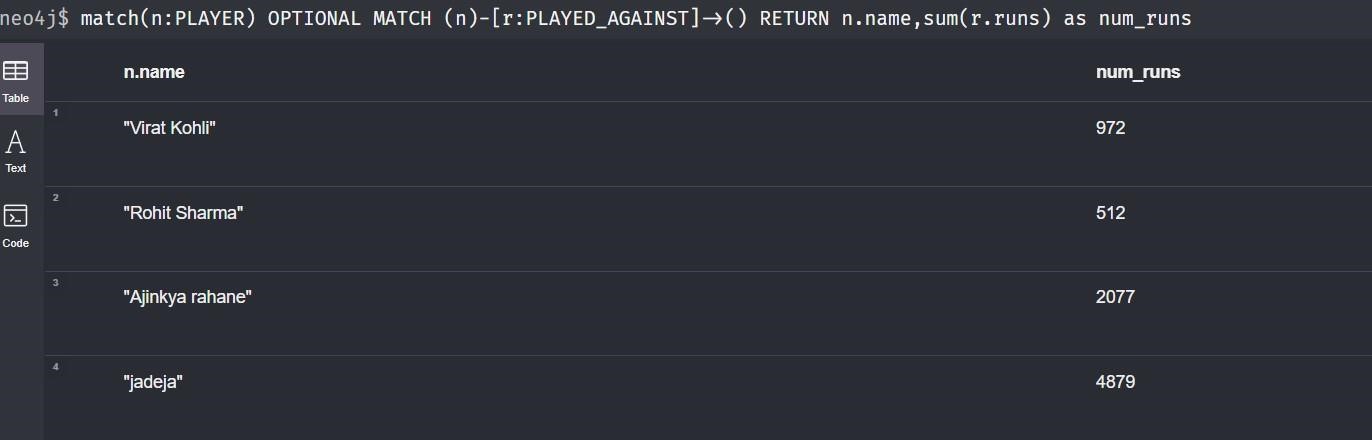
Mathc(n:PLAYER)-[r:TEAMMATES]->(m:PLAYER) where n.name=”virat” return m.salary>40000000



// GET PLAYERS AND NUMBER OF GAMES PLAYED // match(n:PLAYER) OPTIONAL MATCH (n)-[r:PLAYED\_AGAINST]->() RETURN n.name,count(r) as num\_games



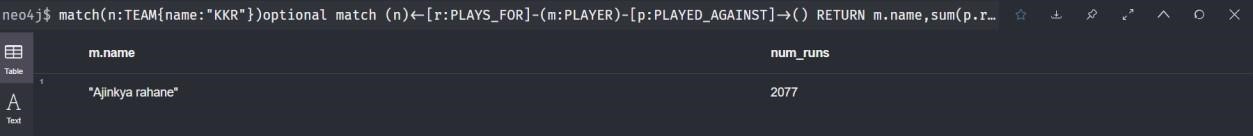
// GET PLAYERS AND runs PER GAME // match(n:PLAYER) OPTIONAL MATCH (n)-[r:PLAYED\_AGAINST]->() RETURN n.name,sum(r.runs) as num\_runs



// GET HIGHEST SCORING PLAYER IN THE Riders //

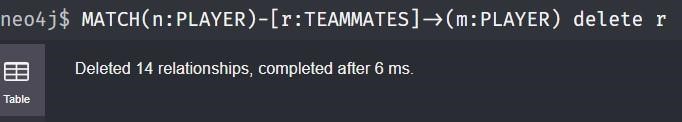
match(n:TEAM{name:"KKR"})optional match (n)<-[r:PLAYS\_FOR]-

(m:PLAYER)[p:PLAYED\_AGAINST]->() RETURN m.name,sum(p.runs) as num\_runs order by num\_runs DESC LIMIT 1

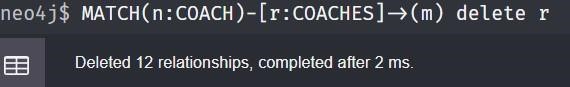


// Delete relationship

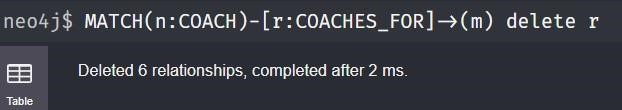
MATCH(n:PLAYER)-[r:TEAMMATES]->(m:PLAYER) delete r



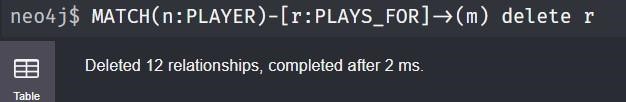
MATCH(n:COACH)-[r:COACHES]->(m) delete r



MATCH(n:COACH)-[r:COACHES\_FOR]->(m) delete r



MATCH(n:PLAYER)-[r:PLAYS\_FOR]->(m) delete r

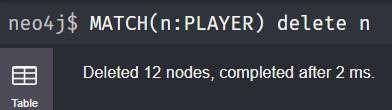


MATCH(n:PLAYER)-[r:PLAYED\_AGAINST]->(m) delete r

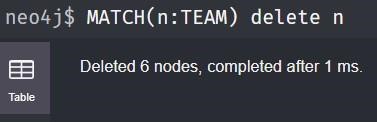


// Delete Node

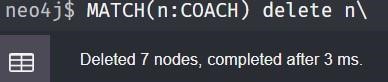
MATCH(n:PLAYER) delete n



MATCH(n:TEAM) delete n

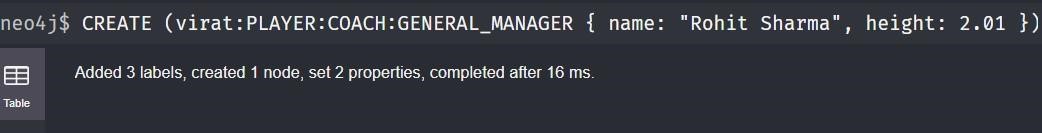


MATCH(n:COACH) delete n

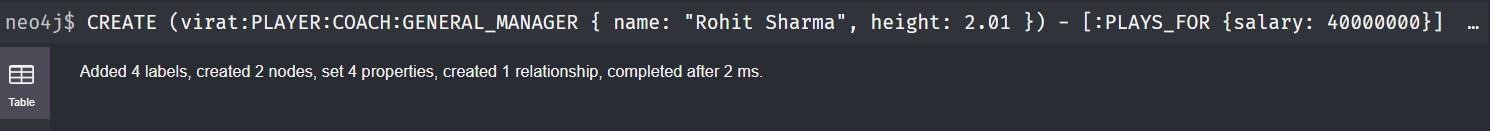


CREATE (virat:PLAYER:COACH:GENERAL\_MANAGER { name: "Rohit Sharma",

height: 2.01 })



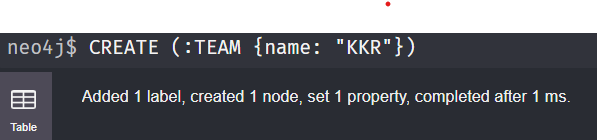
CREATE (virat:PLAYER:COACH:GENERAL\_MANAGER { name: "Rohit Sharma", height: 2.01 }) - [:PLAYS\_FOR {salary: 40000000}] -> (:TEAM {name: "KKR"})



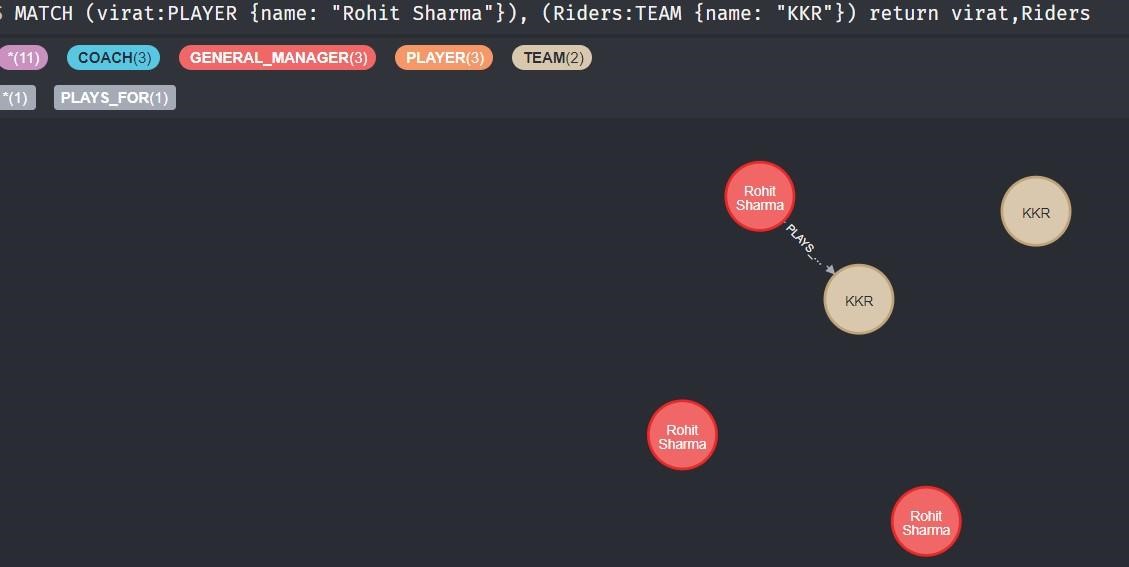
CREATE (virat:PLAYER:COACH:GENERAL\_MANAGER { name: "Rohit Sharma", height: 2.01 })



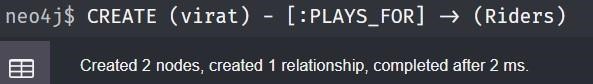
CREATE (:TEAM {name: "KKR"})



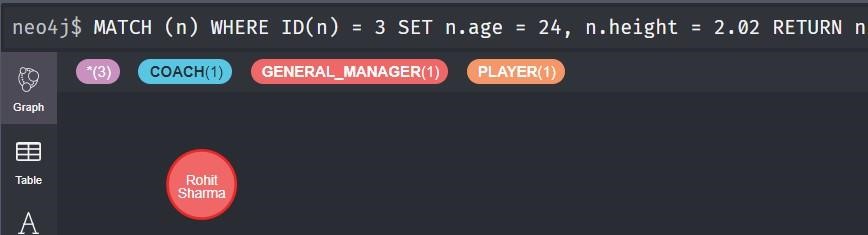
MATCH (virat:PLAYER {name: "Rohit Sharma"}), (Riders:TEAM {name: "KKR"}) return virat,Riders



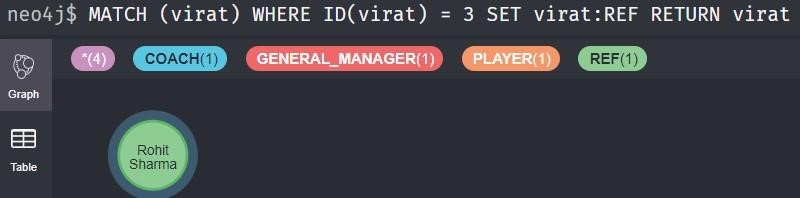
CREATE (virat) - [:PLAYS\_FOR] -> (Riders)



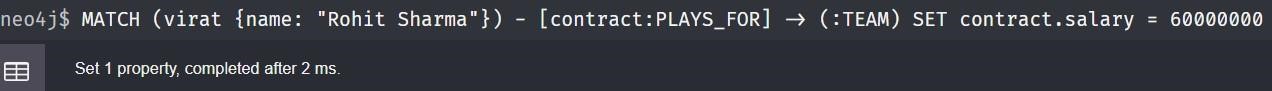
MATCH (n) WHERE ID(n) = 3 SET n.age = 24, n.height = 2.02 RETURN n



MATCH (virat) WHERE ID(virat) = 3 SET virat:REF RETURN virat



MATCH (virat {name: "Rohit Sharma"}) - [contract:PLAYS\_FOR] -> (:TEAM) SET contract.salary = 60000000



MATCH (virat) WHERE ID(virat) = 3 REMOVE virat:REF RETURN virat



MATCH (virat) WHERE ID(virat) = 3 REMOVE virat.age RETURN virat

