NEO4J EXPLAINATIONS AND CODES

1. CREATING DATABASE:

Examples of a few entries that are done on cypher are given below and each of 2 queries are attached for every type of relationships in order to understand the basic structure of our dataset.

a. Player detail:

```
CREATE
(russell:PLAYER{name:"Russell Westbrook", age: 33, number: 0,
height: 1.91, weight: 91}),
(lebron:PLAYER{name:"LeBron James", age: 36, number: 6, height:
2.06, weight: 113})
```

This query creates nodes with different players in NBA and their biographic details .

b. Coach detail:

```
(frank:COACH{name: "Frank Vogel"}),
(taylor:COACH{name: "Taylor Jenkins"})
```

This query creates the nodes for coaches that are there in teams and has their names.

c. Team detail:

```
(lakers:TEAM{name:"LA Lakers"}),
(memphis:TEAM{name:"Memphis Grizzlies"})
```

This query allots names for every teams that are there in NBA and creates their nodes.

d. Teammates relationship:

```
(lebron)-[:TEAMMATES]-> (russell),
(lebron)<-[:TEAMMATES]- (russell)
```

Here, this is a relationship examplar where the relationship name is "TEAMMATES" that shows which player is the teammate of which other players.

e. <u>Coaches relationship:</u>
(frank)-[:COACHES]->(lebron),
(frank)-[:COACHES]->(anthony)

This relationship represents which coach node coaches which player node and the name of the relationship is "COACHES" which helps us understand relationships between player and coaches and which coach belongs to which player.

f. <u>Plays for relationships:</u>
(lebron)-[:PLAYS_FOR {salary: 40000000}]-> (lakers),
(russell)-[:PLAYS_FOR {salary: 33000000}]-> (lakers)

In this relationship exampler we can see that it relates which player playes for which team and a variable salary is attached showing us the price at which that player is playing for. Name of relationship is "PLAYS_FOR".

g. <u>Coaches for relationship:</u>
(frank)-[:COACHES_FOR]->(lakers),
(taylor)-[:COACHES_FOR]->(memphis)

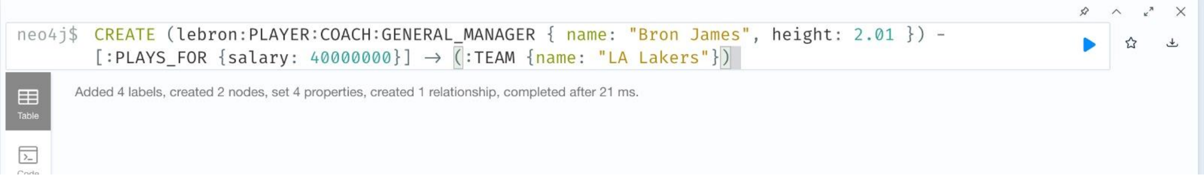
In the relationship given here, the name is "COACHES_FOR" and this relationship shows what coach in the dataset coaches which team specifically.

h. <u>Played against relationship:</u>
(lebron)-[:PLAYED_AGAINST {minutes: 38, points: 32, assists: 6, rebounds: 6, turnovers: 2}]-> (memphis),
(russell)-[:PLAYED_AGAINST {minutes: 29, points: 16, assists: 12, rebounds: 11, turnovers: 16}]-> (memphis)

In the relationship here, we are creating links to all the players that ever played against each other and shows which player played against who hence connecting the nodes.

2. COMMAND AND OUTPUT EXPLAINATION:

a. To create node for player and show team:

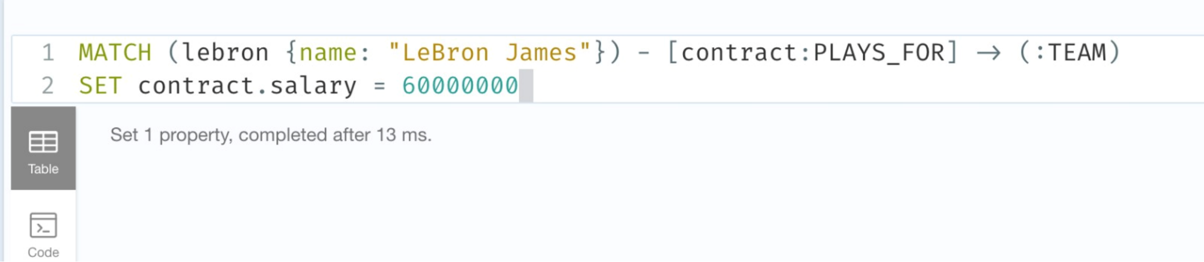


```
neo4j$ CREATE (lebron:PLAYER:COACH:GENERAL_MANAGER { name: "Bron James", height: 2.01 }) -
       [:PLAYS_FOR {salary: 40000000}] → (:TEAM {name: "LA Lakers"})
```

Added 4 labels, created 2 nodes, set 4 properties, created 1 relationship, completed after 21 ms.

In the query as seen here we are creating a player detail named lebron that has name and height in it and continuing to show how much he plays for and what team he is in. That creates 2 nodes and 4 labels with 4 properties as seen in the output.

b. Property updating:

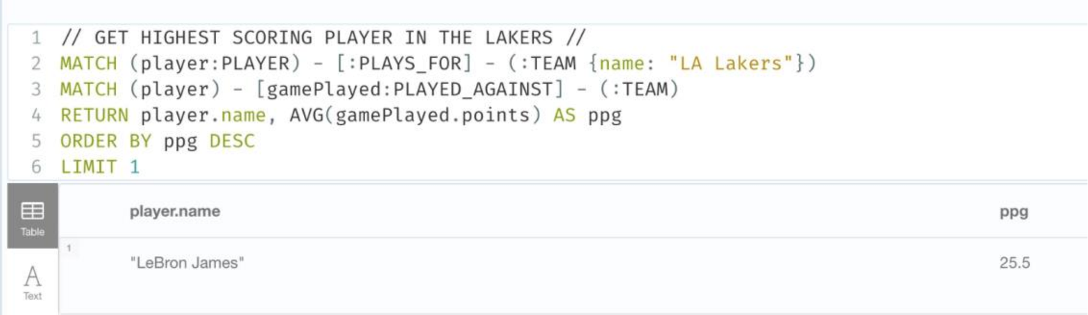


```
MATCH (lebron {name: "LeBron James"}) - [contract:PLAYS_FOR] → (:TEAM)
SET contract.salary = 60000000
```

Set 1 property, completed after 13 ms.

In the query here, we are setting the contract salary of lebron james from 40000000 to 60000000 where the relation is PLAYS_FOR and updating the same.

c. To find highest scoring player in Laker(team):

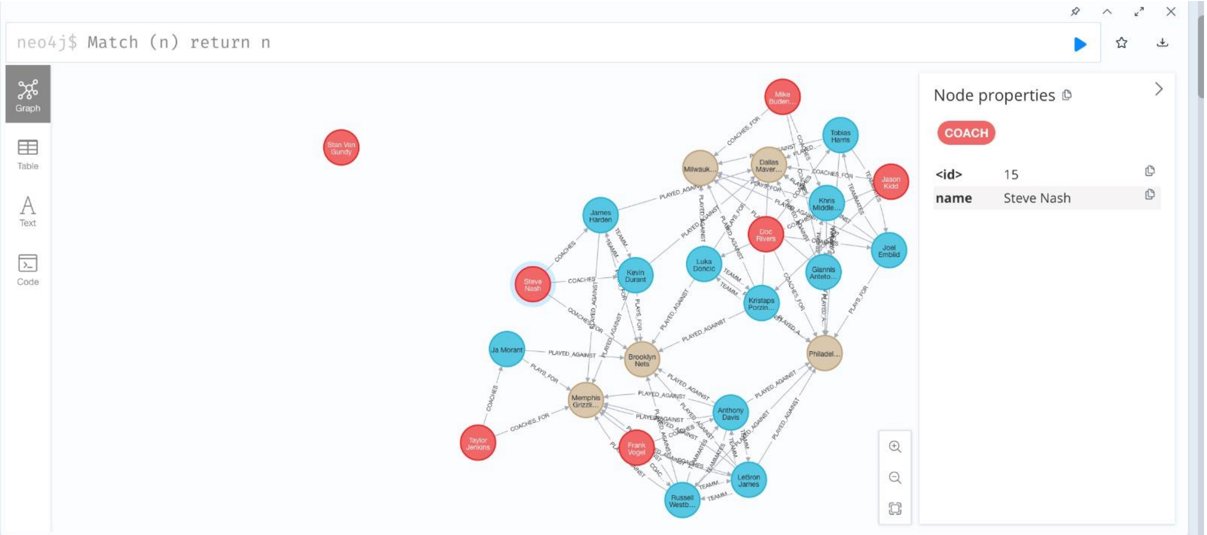


```
// GET HIGHEST SCORING PLAYER IN THE LAKERS //
MATCH (player:PLAYER) - [:PLAYS_FOR] - (:TEAM {name: "LA Lakers"})
MATCH (player) - [gamePlayed:PLAYED_AGAINST] - (:TEAM)
RETURN player.name, AVG(gamePlayed.points) AS ppg
ORDER BY ppg DESC
LIMIT 1
```

	player.name	ppg
1	"LeBron James"	25.5

In the query here, we are trying to find who is the highest scoring player in the LA Lakers (lakers) team and we are matching the player that plays for lakers and who has played against all the teams and on the basis of that, returning the avg points scored by each player and then finding out who has the highest score of all by placing the scores in descending order and the highest scoring player is returned who is Lebron James in this case.

d. Showing entire dataset connections in the form of graph network:



```
neo4j$ Match (n) return n
```

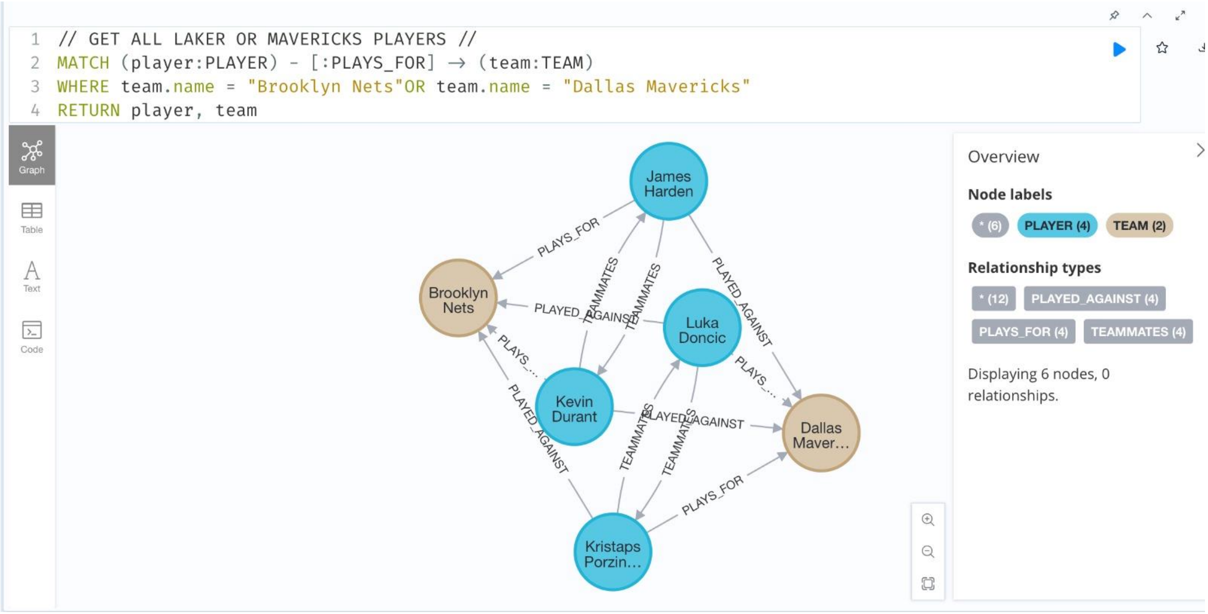
Node properties

COACH

<id>	15
name	Steve Nash

In the plot here, we can see that the query return all the possible node connections amongst all the nodes and their respective relationships and it gives us an overview of the dataset and database and what connections are there in order to provide us with a broader perspective so that we can infer what all details we can fetch from it. Also, there are 3 different nodes that we see which are for “Players”, “Coaches” and “Teams” that are Blue, Red and Beige respectively.

e. Query to get all lakers or mavericks players and check the network:

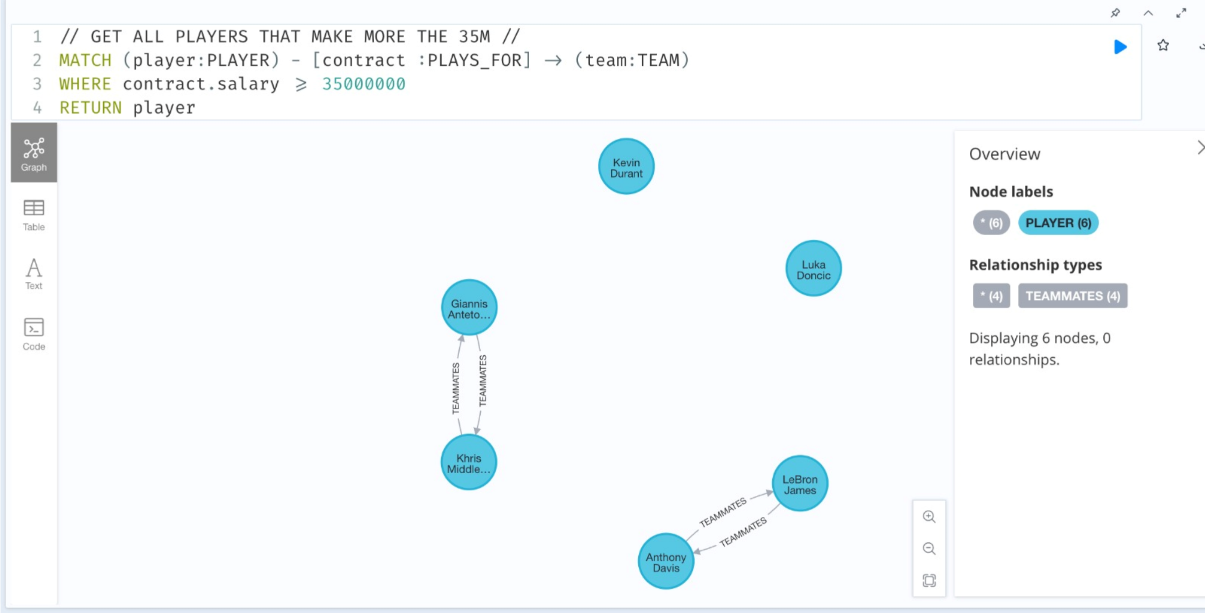


```
// GET ALL LAKER OR MAVERICKS PLAYERS //
MATCH (player:PLAYER) - [:PLAYS_FOR] -> (team:TEAM)
WHERE team.name = "Brooklyn Nets"OR team.name = "Dallas Mavericks"
RETURN player, team
```

Displaying 6 nodes, 0 relationships.

This query here matches and returns all the players in LA Lakers or Dallas Mavericks and connect them with either of the 2 teams and also shows relations among each player hence giving us a better relation understanding of who played against whom. There can be seen 3 relationships seen here namely plays_for, teammates and plays_against to clarify what each player node is to other nodes.

f. Player earnings analysis:

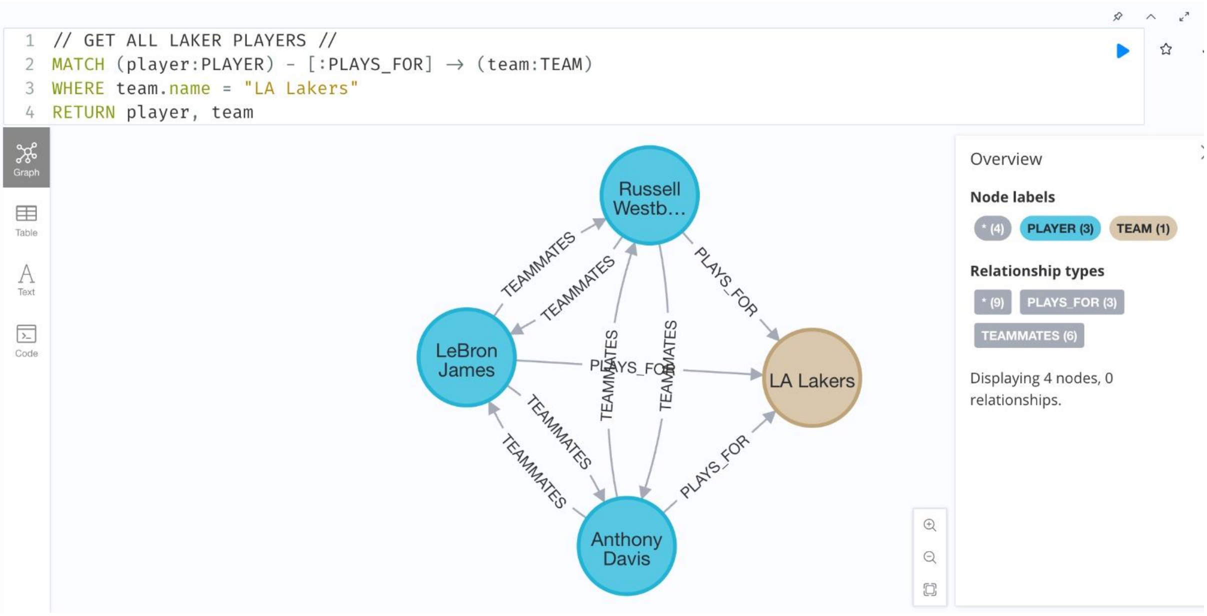


```
// GET ALL PLAYERS THAT MAKE MORE THE 35M //
MATCH (player:PLAYER) - [contract :PLAYS_FOR] -> (team:TEAM)
WHERE contract.salary >= 35000000
RETURN player
```

Displaying 6 nodes, 0 relationships.

The query as shown helps us understand the earnings of the players in the database and here we have a constraint that only gives us the players who earn more than 35 million dollars. Not just that but we also get an analysis weather any of the players are in the same team or not and hence giving us a better understanding of the economic stability of the teams as well since we also get the number of players from 1 team here. We can use this query to analyse the player economy and change the value accordingly to understand and predict that for how much that player can be sold to another team and so on.

g. Team player details:



```
// GET ALL LAKER PLAYERS //
MATCH (player:PLAYER) - [:PLAYS_FOR] → (team:TEAM)
WHERE team.name = "LA Lakers"
RETURN player, team
```

Overview

Node labels

* (4) PLAYER (3) TEAM (1)

Relationship types

* (9) PLAYS_FOR (3)

TEAMMATES (6)

Displaying 4 nodes, 0 relationships.

From the query here, we are printing all the players in a specific team individually to understand what players are in what teams and fetch the details pictorially rather than reading the data through. We can use this query to see total number of players in the same team from the dataset like here we have the players in lakers are there are 2 relationships seen one being teammate showing what player are teammates to each other and the other plays_for to show each player playing for what team. This analysis can be very useful when we wish to compare maybe that in the top 10 player list, how many of them belong to a same team giving us an upper edge to analyse the training strength of a team.