Project Title	**Football Graph Database** **A Neo4j-Based Modeling of Football Players, Teams, and Their Relationships**
Team members	Seif El-Deen Samer 222100794 Ahmed Abdelkhalek 222100115 Hajar Mohammed 222200040 Aser Mohamed Ali 222102487 Ammar Ali Mohamed 222101041 Merna Muhammed 222101717
Field	Computer Science and Engineering
Program	AIS
Instructor	**Prof. Manar El-Shazly**
This part for instructor: Notice	Degree

Spring 2025

Abstract

The football graph database project aims to create an extensive knowledge base for football-related entities using Neo4j, a graph database technology. This database models football players, teams, coaches, countries, trophies, and stadiums, along with their intricate relationships, such as team memberships, player transfers, national team representations, coaching assignments, and match occurrences. This database serves as a powerful resource for analyzing and visualizing the vast network of football interactions, enabling deep insights into player movements, team dynamics, and competition results.

---

1. Introduction

In modern football, data plays a crucial role in strategic planning, player performance analysis, and decision-making. The football graph database aims to integrate diverse aspects of the football ecosystem into a unified, flexible, and easily queryable structure. By using graph technology, we can effectively represent the interconnections between players, teams, coaches, countries, trophies, and stadiums. This project leverages Neo4j, a leading graph database, to store and query the data, allowing for seamless exploration of football-related relationships.

1.1. Aim & Objective of the Project

The primary objective of this project is to design and implement a graph database for football that can:

- Store detailed information about football players, teams, coaches, countries, and stadiums.
- Represent various relationships between these entities, such as "PLAYS_FOR," "MANAGES," "WON," and "REPRESENTS."
- Enable querying of complex relationships such as player transfers, team rivalries, and country representations in tournaments.

---

2. Related Work

Graph databases have been increasingly used for modeling complex relationships in various domains, including sports. Previous works in the football domain primarily focused on player performance analysis, team strategy, and event prediction based on historical match data.

Several initiatives have explored football data modeling using Neo4j:

- **Football Transfers Analysis**: Mark Needham's work on analyzing football transfers provides insights into modeling transfer data and player movements .Medium+2Graph Database & Analytics+2YouTube+2
- **Premier League Graph Modeling**: The Neo4j and Premier League project demonstrates modeling match data and querying for performance metrics .www.slideshare.net+1www.slideshare.net+1
- **World Cup Historical Data**: Joshua C. Fjelstul's comprehensive dataset covers all FIFA World Cup tournaments, offering a rich resource for historical analysis .Graph Database & Analytics

2.1. Use of Graph Databases in Football Analytics

Many modern sports analytics tools use graph-based approaches for player and team performance analysis, where relationships such as "played with," "played against," and "scored goals" are mapped into a graph. Researchers like **Bavetta et al. (2020)** and **Hernandez et al. (2019)** utilized graph databases to model football networks and analyze player relationships. Their findings highlighted the effectiveness of graph models in capturing the dynamic and interconnected nature of football teams and players.

2.2. Graph Databases for Sports Fan Engagement

A recent trend has emerged in using graph databases to engage football fans by analyzing social media interactions and player popularity. **Mitchell et al. (2022)** built a system that used graph databases to track fan interactions with football players and teams, which helped in predicting social media trends.

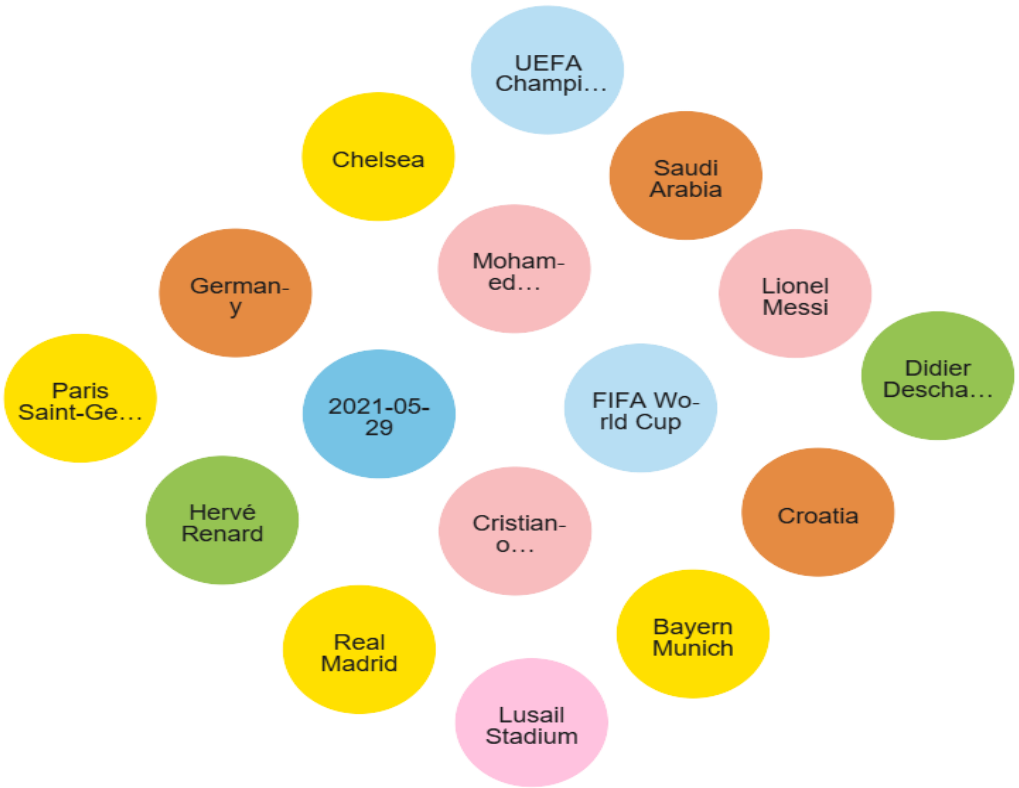
2.3. Graph Databases for Team Building and Match Prediction

Studies have also been conducted to model football team dynamics and predict match outcomes based on player attributes and team strategies. **Lin and Chen (2021)** showed how graph theory can be used to predict the success of football teams in leagues based on their performance metrics.

3. Database Structure and Schema

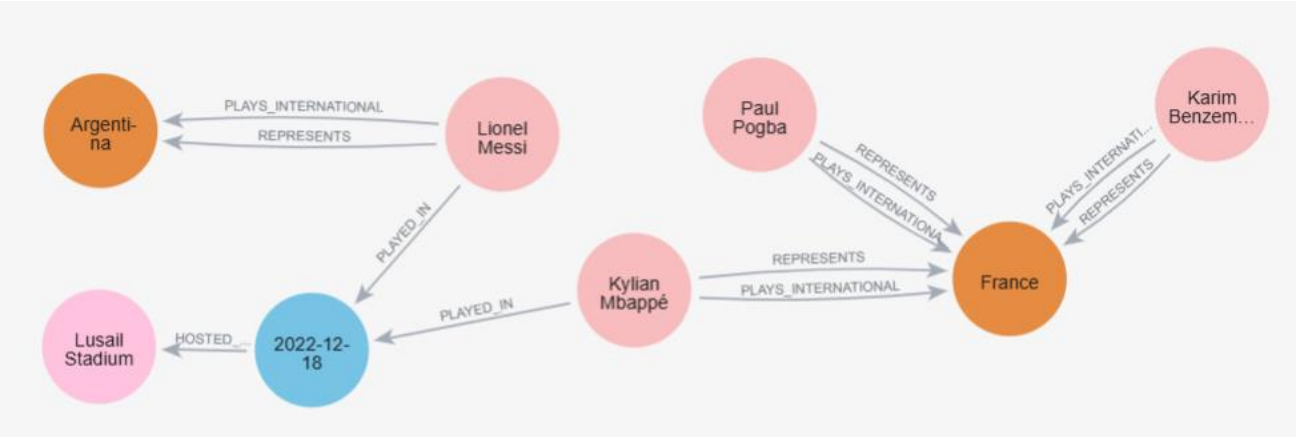
The football database contains various node types and relationships, with the following primary nodes:

- **Player**: Information about individual players, including attributes such as name, age, nationality, and position.
- **Team**: Represents football teams with details like team name, country, and league.
- **Coach**: Information about coaches, including their names and nationalities.
- **Country**: Represents national football teams, including confederation and country information.
- **Trophy**: Represents football trophies with names and years.
- **Stadium**: Represents football stadiums with their names, city, and capacity.
- **Match**: Represents football matches, including match IDs, date, and competition.



3.1. Relationships

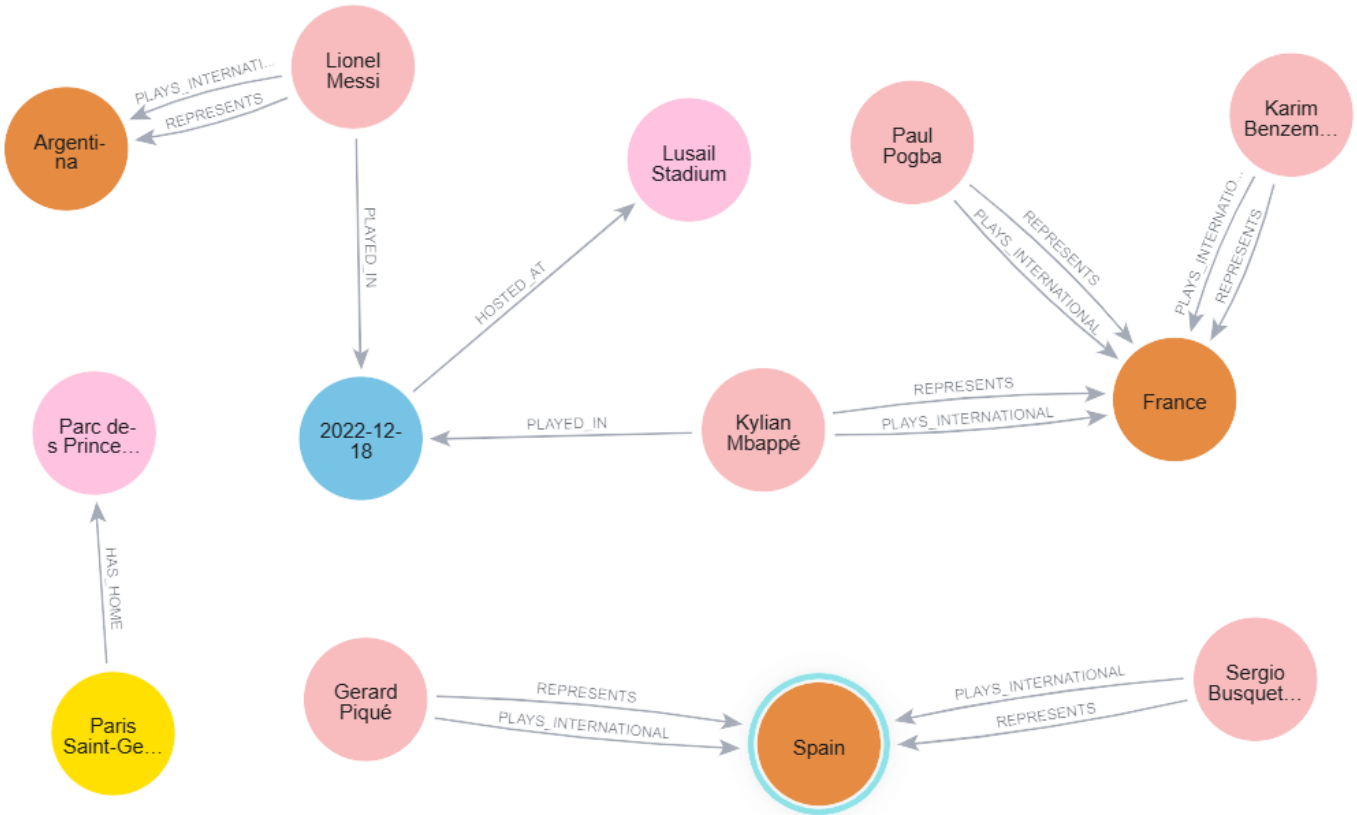
- **PLAYS_FOR**: Connects players to the teams they play for.
- **REPRESENTS**: Connects players to the countries they represent.
- **MANAGES**: Connects coaches to the teams they manage.
- **WON**: Connects teams to the trophies they have won.
- **PLAYED_IN**: Connects players to the matches they have played in.
- **HOSTED_AT**: Connects matches to the stadiums where they were held.
- **HAS_HOME**: Connects teams to their home stadiums.
- **HELD_IN**: Connects trophies to the countries where they were held.
- **TRANSFERRED_FROM/TRANSFERRED_TO**: Connects players to the teams they have transferred from and to.



4. Database Population and Relationships

The database was populated with real-world data about football players, teams, and other entities. The following relationships were created to provide a rich, interconnected graph:

- **Player to Team**: Players were assigned to teams they play for using the `PLAYS_FOR` relationship.
- **Player to Country**: Players were connected to the countries they represent in international competitions using the `REPRESENTS` relationship.
- **Coach to Team**: Coaches were linked to the teams they manage using the `MANAGES` relationship.
- **Team to Trophy**: Teams were connected to trophies they have won using the `WON` relationship.
- **Player to Match**: Players were connected to the matches they participated in using the `PLAYED_IN` relationship.
- **Team to Stadium**: Teams were assigned their home stadiums using the `HAS_HOME` relationship.



5. Querying the Graph Database (in neo4j browser):

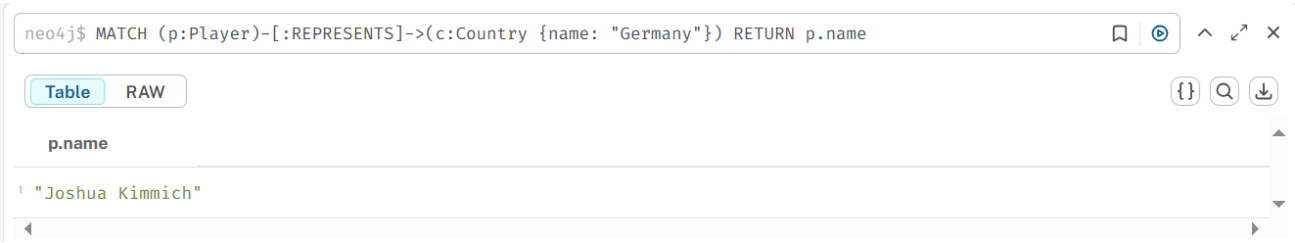
The Neo4j graph database provides a powerful query language called Cypher for exploring and retrieving data. Examples of useful queries for this project include:

1. Get all players from a particular team:

```
MATCH (p:Player)-[:PLAYS_FOR]->(t:Team {name: "FC Barcelona"})
RETURN p.name
```

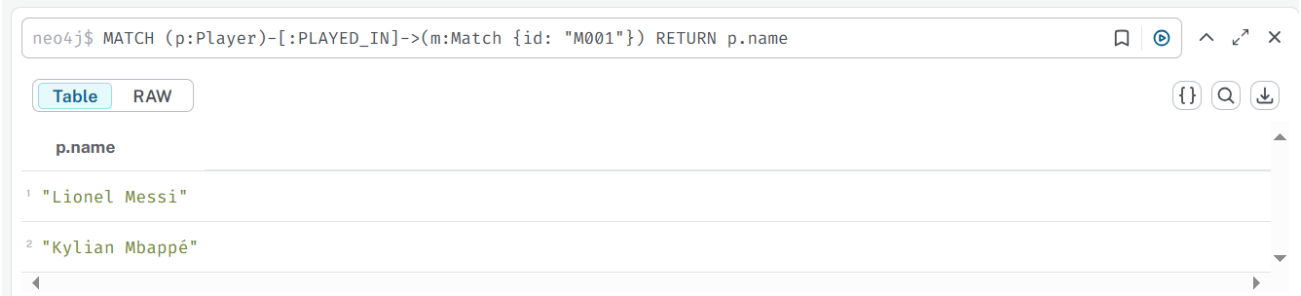
2. Get all players who represent a specific country

```
MATCH (p:Player)-[:REPRESENTS]->(c:Country {name: "Germany"})
RETURN p.name
```



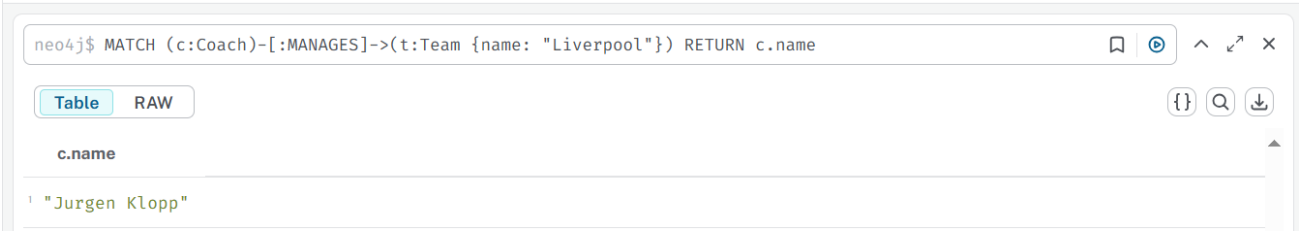
3. Find players who played in a specific match

```
MATCH (p:Player)-[:PLAYED_IN]->(m:Match {id: "M001"})
RETURN p.name
```



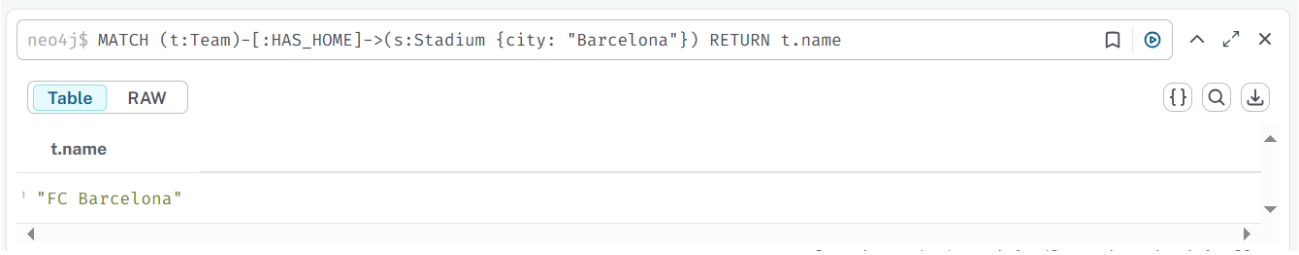
4. Get all coaches who manage a specific team

```
MATCH (c:Coach)-[:MANAGES]->(t:Team {name: "Liverpool"})
RETURN c.name
```



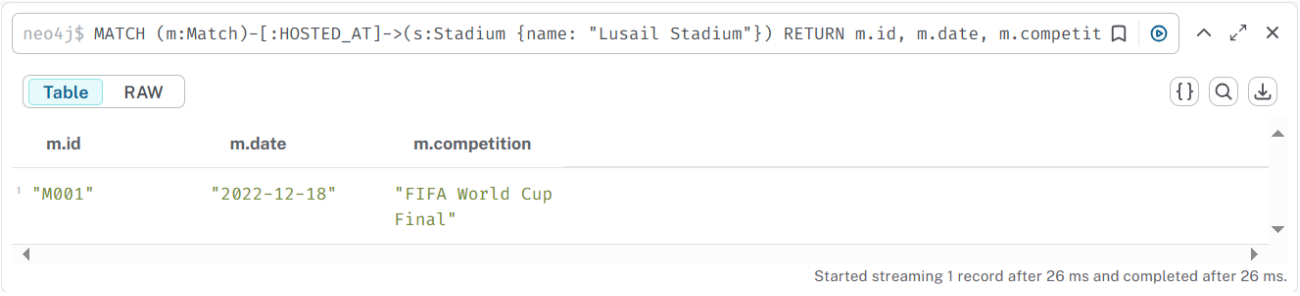
5. Find teams that have a home stadium in a specific city

```
MATCH (t:Team)-[:HAS_HOME]->(s:Stadium {city: "Barcelona"})
RETURN t.name
```



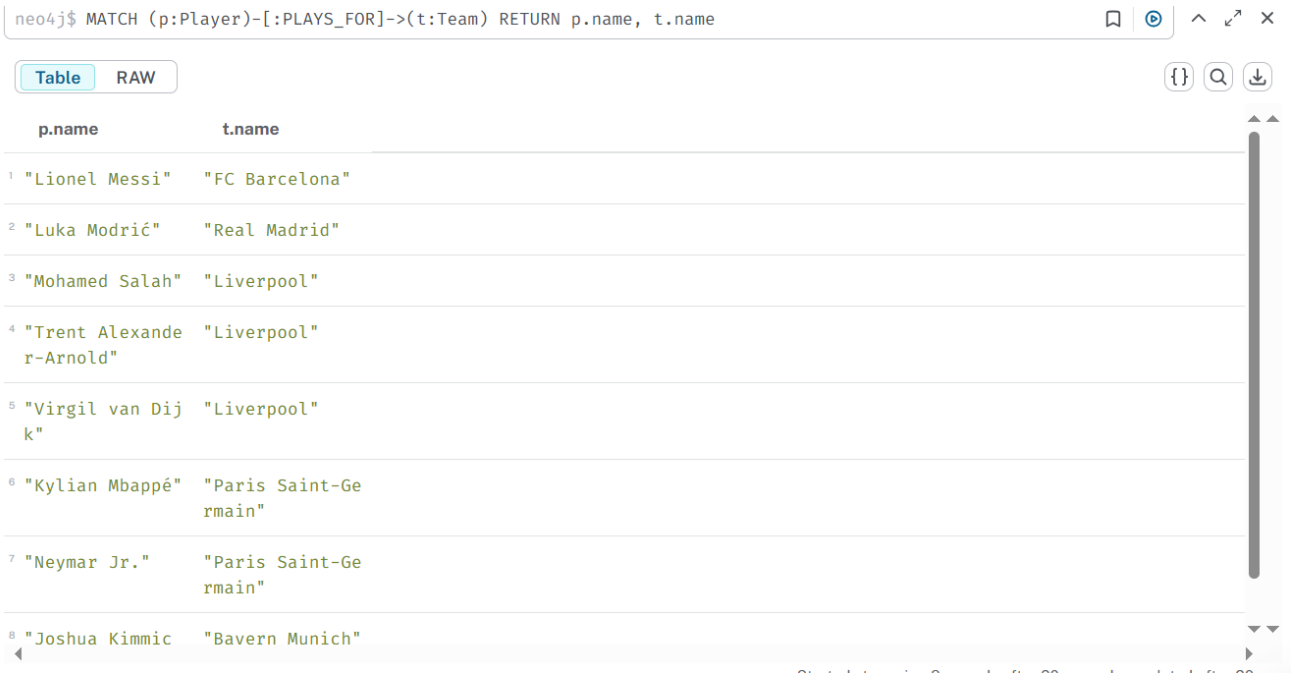
6. Find all matches hosted at a specific stadium

```
MATCH (m:Match)-[:HOSTED_AT]->(s:Stadium {name: "Lusail Stadium"})
RETURN m.id, m.date, m.competition
```



7. List of all players and their respective teams

```
MATCH (p:Player)-[:PLAYS_FOR]->(t:Team)
RETURN p.name, t.name
```

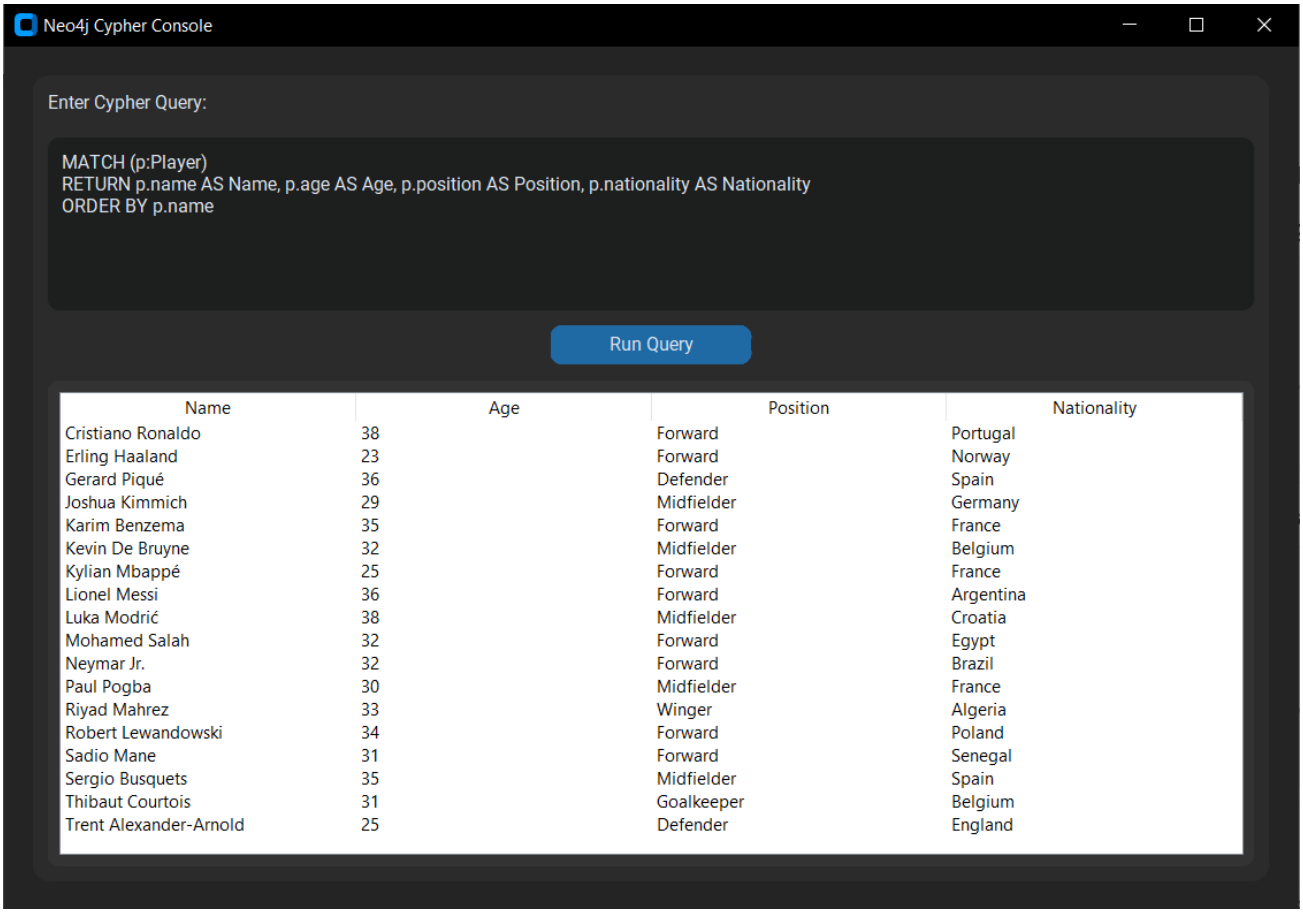


```
neo4j$ MATCH (p:Player)-[:PLAYS_FOR]->(t:Team) RETURN p.name, t.name
```

Table | RAW

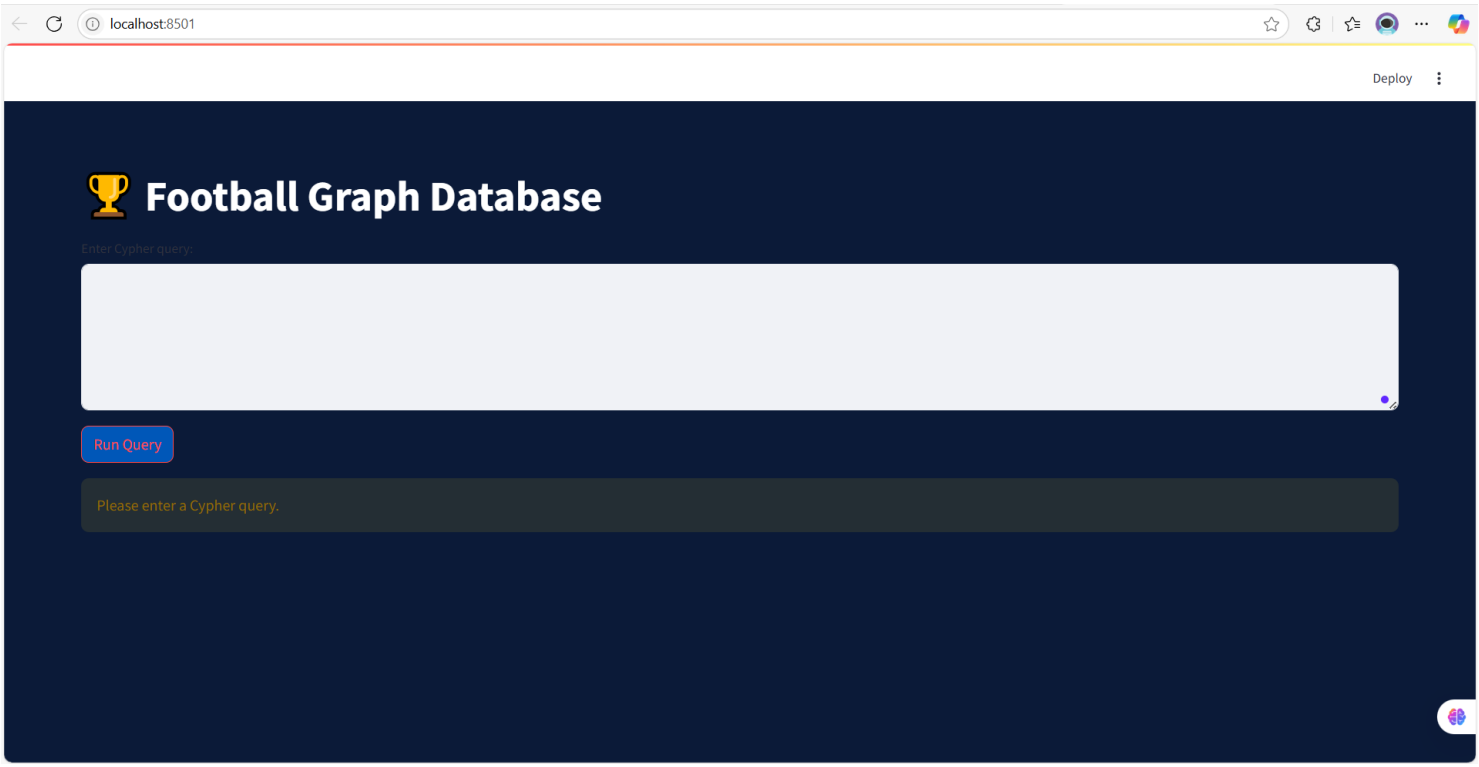
	p.name	t.name
1	"Lionel Messi"	"FC Barcelona"
2	"Luka Modrić"	"Real Madrid"
3	"Mohamed Salah"	"Liverpool"
4	"Trent Alexander-Arnold"	"Liverpool"
5	"Virgil van Dijk"	"Liverpool"
6	"Kylian Mbappé"	"Paris Saint-Germain"
7	"Neymar Jr."	"Paris Saint-Germain"
8	"Joshua Kimmic	"Bavern Munich"

The Tkinter GUI:



Name	Age	Position	Nationality
Cristiano Ronaldo	38	Forward	Portugal
Erling Haaland	23	Forward	Norway
Gerard Piqué	36	Defender	Spain
Joshua Kimmich	29	Midfielder	Germany
Karim Benzema	35	Forward	France
Kevin De Bruyne	32	Midfielder	Belgium
Kylian Mbappé	25	Forward	France
Lionel Messi	36	Forward	Argentina
Luka Modrić	38	Midfielder	Croatia
Mohamed Salah	32	Forward	Egypt
Neymar Jr.	32	Forward	Brazil
Paul Pogba	30	Midfielder	France
Riyad Mahrez	33	Winger	Algeria
Robert Lewandowski	34	Forward	Poland
Sadio Mane	31	Forward	Senegal
Sergio Busquets	35	Midfielder	Spain
Thibaut Courtois	31	Goalkeeper	Belgium
Trent Alexander-Arnold	25	Defender	England

The Streamlit GUI:



Querying the Graph Database in the GUI (Tables & Graph):

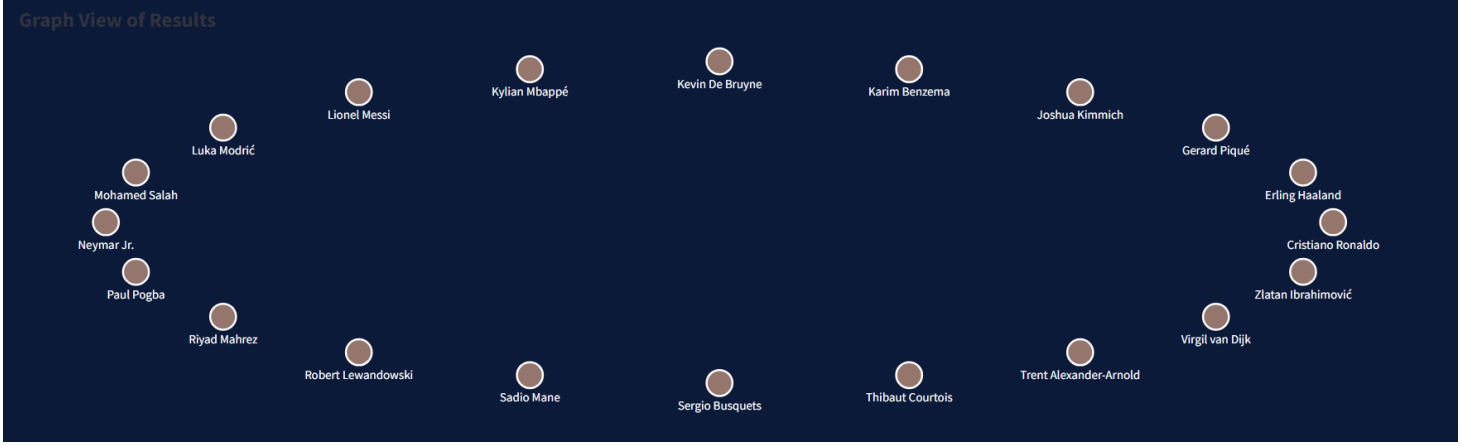
1. List all players (name, age, position, nationality):

```
MATCH (p:Player)
RETURN p, p.name AS Name, p.age AS Age, p.position AS Position, p.nationality AS
Nationality
ORDER BY p.name
```

Query Results (Table)

	p	Name	Age	Position	Nationality
0	Player: Cristiano Ronaldo	Cristiano Ronaldo	38	Forward	Portugal
1	Player: Erling Haaland	Erling Haaland	23	Forward	Norway
2	Player: Gerard Piqué	Gerard Piqué	36	Defender	Spain
3	Player: Joshua Kimmich	Joshua Kimmich	29	Midfielder	Germany
4	Player: Karim Benzema	Karim Benzema	35	Forward	France
5	Player: Kevin De Bruyne	Kevin De Bruyne	32	Midfielder	Belgium
6	Player: Kylian Mbappé	Kylian Mbappé	25	Forward	France
7	Player: Lionel Messi	Lionel Messi	36	Forward	Argentina
8	Player: Luka Modrić	Luka Modrić	38	Midfielder	Croatia
9	Player: Mohamed Salah	Mohamed Salah	32	Forward	Egypt

Query Results (Graph Visualization)

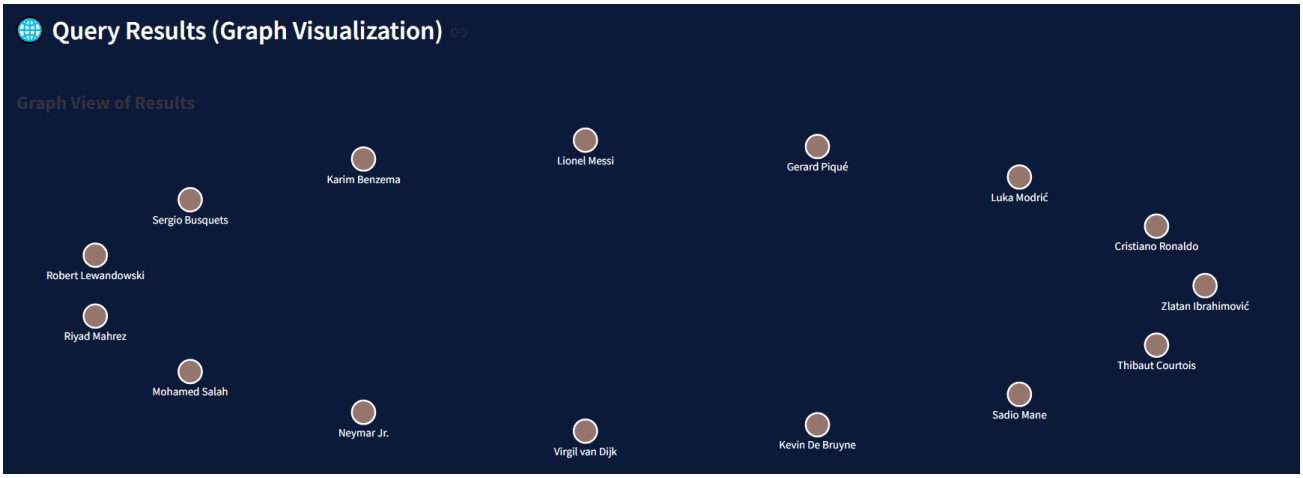


2. Find all players older than 30:

```
MATCH (p:Player)
WHERE p.age > 30
RETURN p, p.name AS Name, p.age AS Age
ORDER BY p.age DESC
```

Query Results (Table)

	p	Name	Age
0	Player: Zlatan Ibrahimović	Zlatan Ibrahimović	43
1	Player: Cristiano Ronaldo	Cristiano Ronaldo	38
2	Player: Luka Modrić	Luka Modrić	38
3	Player: Gerard Piqué	Gerard Piqué	36
4	Player: Lionel Messi	Lionel Messi	36
5	Player: Karim Benzema	Karim Benzema	35
6	Player: Sergio Busquets	Sergio Busquets	35
7	Player: Robert Lewandowski	Robert Lewandowski	34
8	Player: Riyad Mahrez	Riyad Mahrez	33
9	Player: Mohamed Salah	Mohamed Salah	32



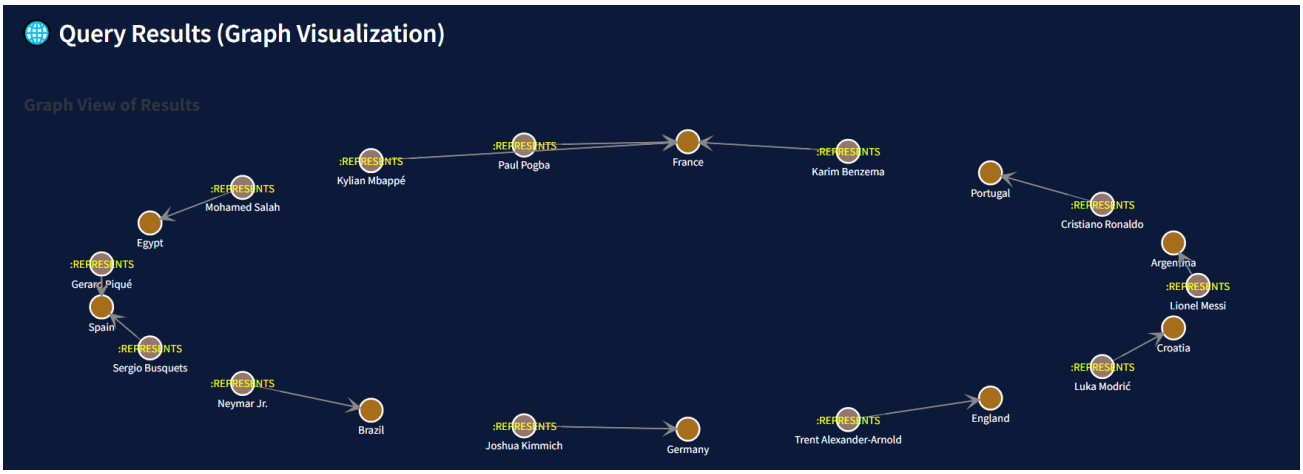
3. Count number of players per country:

```
MATCH (p:Player)-[r:REPRESENTS]->(c:Country)
RETURN p, c, r, c.name AS Country, COUNT(p) AS PlayerCount
ORDER BY PlayerCount DESC
```

Query Results (Table)

	p	c	r	Country	PlayerCount
0	Player: Lionel Messi	Country: Argentina	:REPRESENTS (56 → 36)	Argentina	1
1	Player: Cristiano Ronaldo	Country: Portugal	:REPRESENTS (57 → 37)	Portugal	1
2	Player: Karim Benzema	Country: France	:REPRESENTS (0 → 38)	France	1
3	Player: Paul Pogba	Country: France	:REPRESENTS (3 → 38)	France	1
4	Player: Kylian Mbappé	Country: France	:REPRESENTS (58 → 38)	France	1
5	Player: Mohamed Salah	Country: Egypt	:REPRESENTS (59 → 39)	Egypt	1
6	Player: Gerard Piqué	Country: Spain	:REPRESENTS (4 → 40)	Spain	1
7	Player: Sergio Busquets	Country: Spain	:REPRESENTS (60 → 40)	Spain	1
8	Player: Neymar Jr.	Country: Brazil	:REPRESENTS (61 → 41)	Brazil	1
9	Player: Joshua Kimmich	Country: Germany	:REPRESENTS (63 → 42)	Germany	1

Query Results (Graph Visualization)



4. Show each team with its home stadium with its capacity:

```
MATCH (t:Team)-[r:HAS_HOME]->(s:Stadium)
RETURN t, s, r, t.name AS Team, s.name AS HomeStadium, s.capacity AS Capacity
ORDER BY t.name
```

Query Results (Table)

	t	s	r	Team	HomeStadium	Capacity
0	Team: FC Barcelona	Stadium: Camp Nou	:HAS_HOME (46 → 11)	FC Barcelona	Camp Nou	99,354
1	Team: Liverpool	Stadium: Anfield	:HAS_HOME (48 → 16)	Liverpool	Anfield	54,074
2	Team: Paris Saint-Germain	Stadium: Parc des Princes	:HAS_HOME (49 → 14)	Paris Saint-Germain	Parc des Princes	47,929
3	Team: Real Madrid	Stadium: Santiago Bernabéu	:HAS_HOME (47 → 15)	Real Madrid	Santiago Bernabéu	81,000

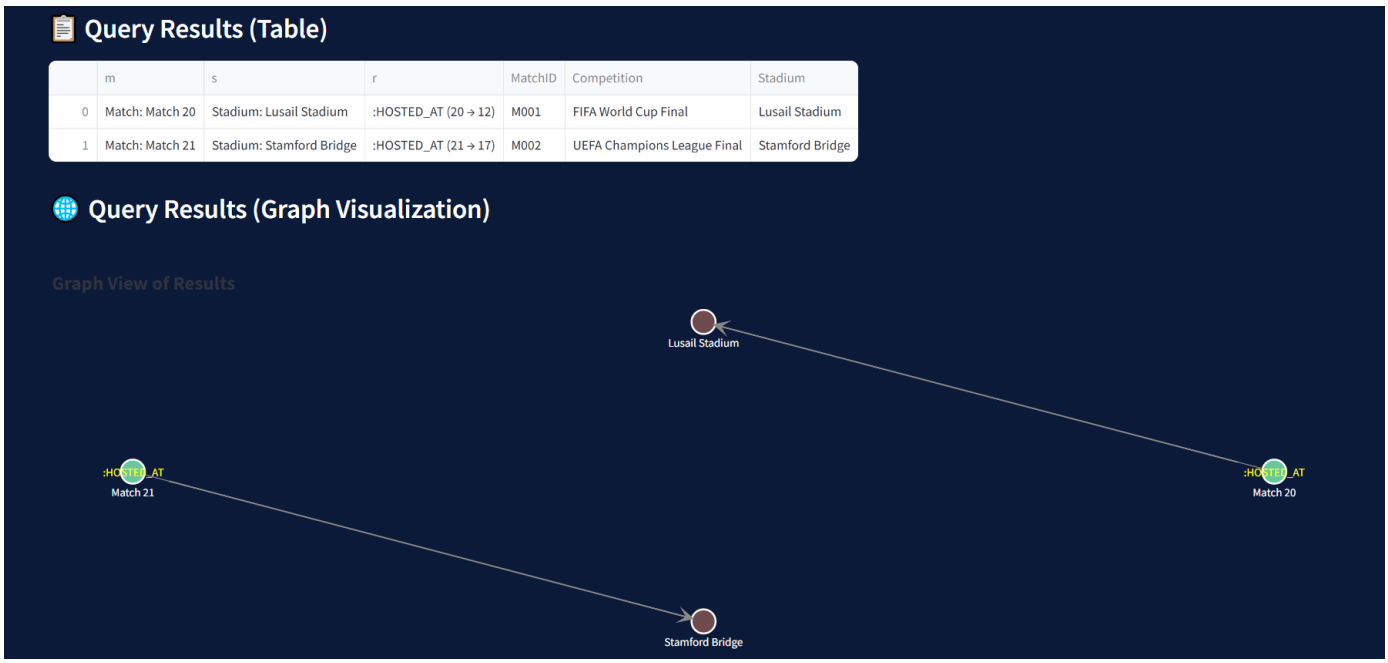
Query Results (Graph Visualization)

5. Who manages each team?

```
MATCH (c:Coach)-[r:MANAGES]->(t:Team)
RETURN c, t, r, c.name AS Coach, t.name AS Team
ORDER BY c.name
```

6. Matches and where they were hosted:

```
MATCH (m:Match)-[r:HOSTED_AT]->(s:Stadium)
RETURN m, s, r, m.id AS MatchID, m.competition AS Competition, s.name AS Stadium
ORDER BY m.date DESC
```

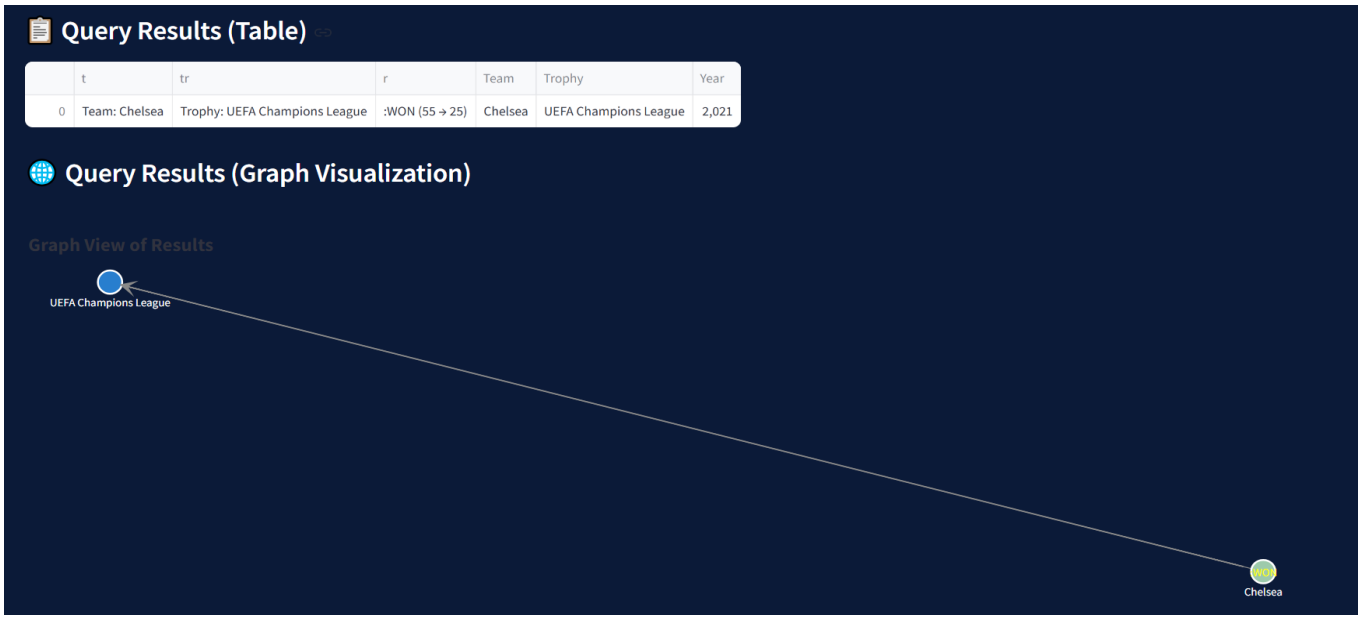


7. Players who played in the World Cup Final (M001)?

```
MATCH (p:Player)-[r:PLAYED_IN]->(m:Match {id:"M001"})
RETURN p, m, r, p.name AS Player
```

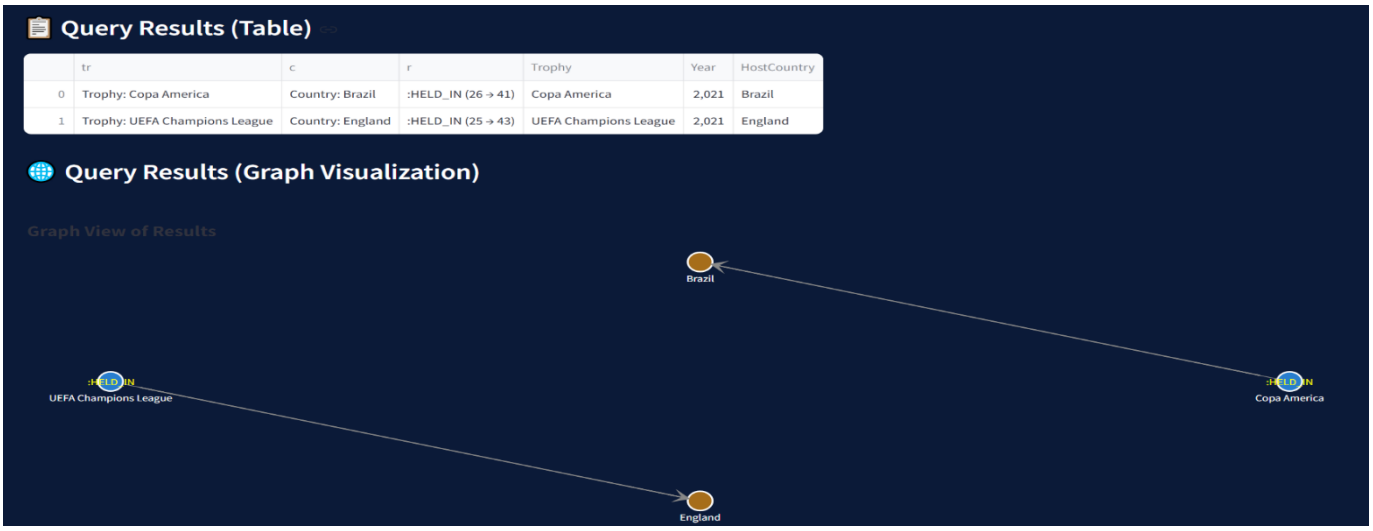
8. Teams that have won trophies and the years:

```
MATCH (t:Team)-[r:WON]->(tr:Trophy)
RETURN t, tr, r, t.name AS Team, tr.name AS Trophy, tr.year AS Year
ORDER BY tr.year DESC
```



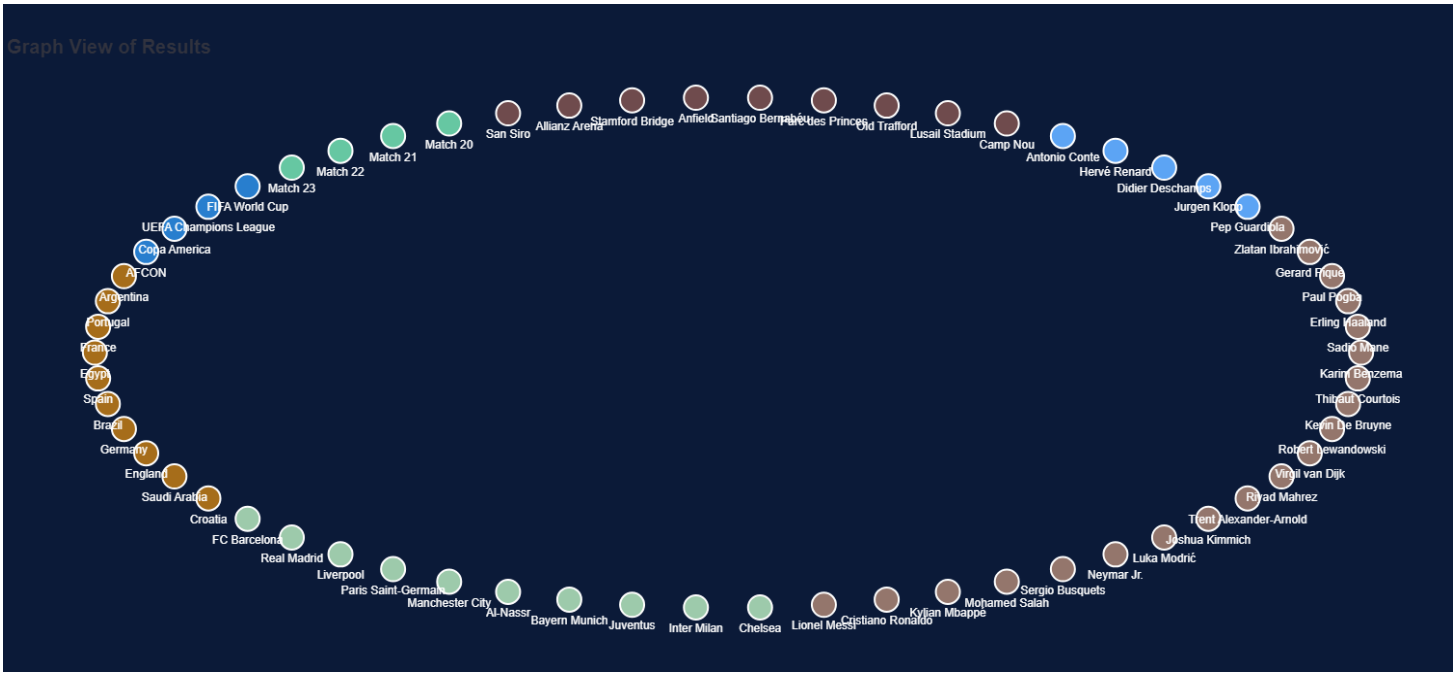
9. Countries that hosted each trophy?

```
MATCH (tr:Trophy)-[r:HELD_IN]->(c:Country)
RETURN tr, c, r, tr.name AS Trophy, tr.year AS Year, c.name AS HostCountry
ORDER BY tr.name, tr.year
```



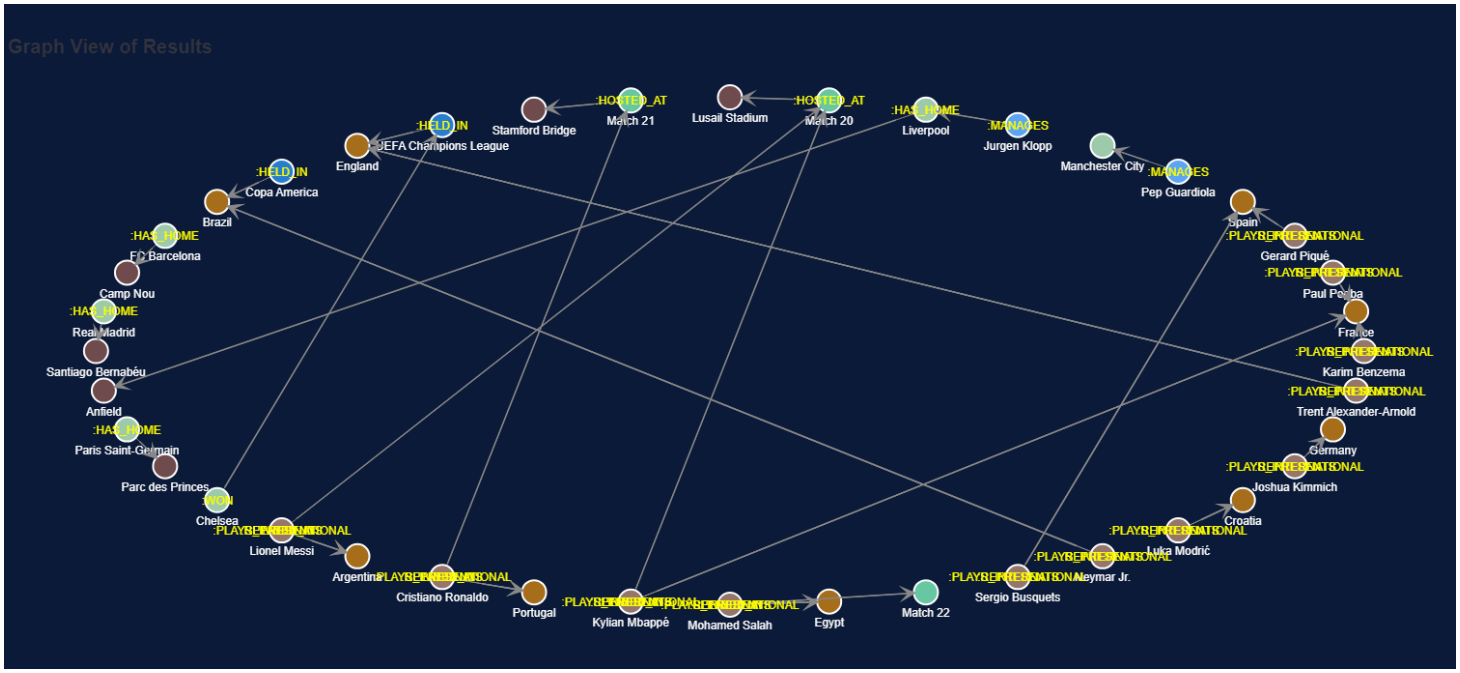
10. All nodes in the graph:

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

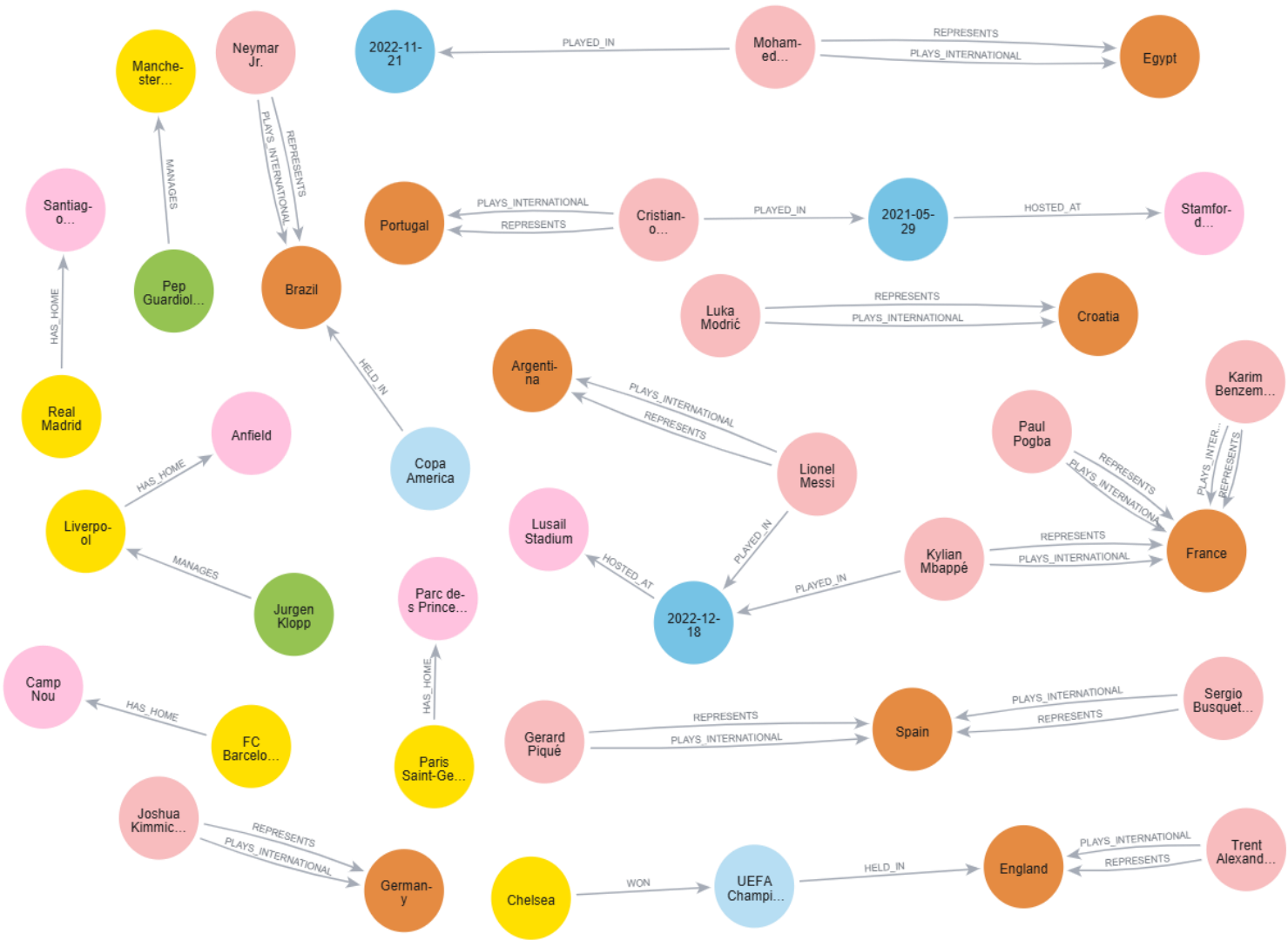


11. All nodes and relationships (full graph):

```
MATCH (n)-[r]->(m)
RETURN n, r, m
```



Visualization of the Graph Database (in neo4j browser):



7. Results and Discussion

The final result of this project is a fully-populated football database with nodes representing players, teams, coaches, trophies, stadiums, and matches. The relationships between these entities provide a comprehensive view of the football world, allowing for complex queries to uncover hidden insights. The project demonstrates how graph databases can be used to effectively model sports-related data and derive insights from the relationships between entities. This structure supports advanced analytics, such as identifying key players in tournaments or analyzing coaching impacts on team performance.

7.1. Use Cases

- **Player Transfer Analysis**: Analyzing the transfer patterns of football players across teams.
- **Team Performance Insights**: Understanding team performance through the relationship between teams and the trophies they have won.
- **Match Prediction**: Predicting match outcomes based on player and team statistics.

7.2. Limitations and Future Work

While this database is rich with relationships, there are limitations in terms of real-time data updates and dynamic querying. Future work could include integrating real-time match data, player performance metrics, and more detailed historical data for predictive analytics.

---

8. Conclusion

The Neo4j Football Graph Database provides a robust framework for modeling and analyzing football data. Its graph structure captures the complexity of relationships inherent in the sport, offering valuable insights for analysts, enthusiasts, and professionals. Future enhancements could include integrating real-time match data or expanding to include women's football leagues.

References

[1] Needham, Mark. "Analysing Football Transfers with Neo4j." *Neo4j GraphGists*. https://neo4j.com/graphgists/cd8868d1-da9a-44ad-a221-baab3086c902/Graph Database & Analytics

[2] Needham, Mark. "Football Graph - Neo4j and the Premier League." *SlideShare*. https://www.slideshare.net/markhneedham/football-graph-neo4j-and-the-premier-leaguewww.slideshare.net+1www.slideshare.net+1

[3] Fjelstul, Joshua C. "Euro 2024 Special: Exploring World Cup History with Neo4j." *Neo4j Videos*. https://neo4j.com/videos/euro-2024-special-exploring-world-cup-history-with-neo4j/Graph Database & Analytics

[4] "Example Datasets - Getting Started." *Neo4j Documentation*. https://neo4j.com/docs/getting-started/appendix/example-data/Graph Database & Analytics

[5] Graph Data Science by Andrés Taylor & Michael Hunger (Neo4j)
Techniques for community detection, centrality, link prediction using Neo4j's GDS library.
Online: https://neo4j.com/graph-data-science-book/