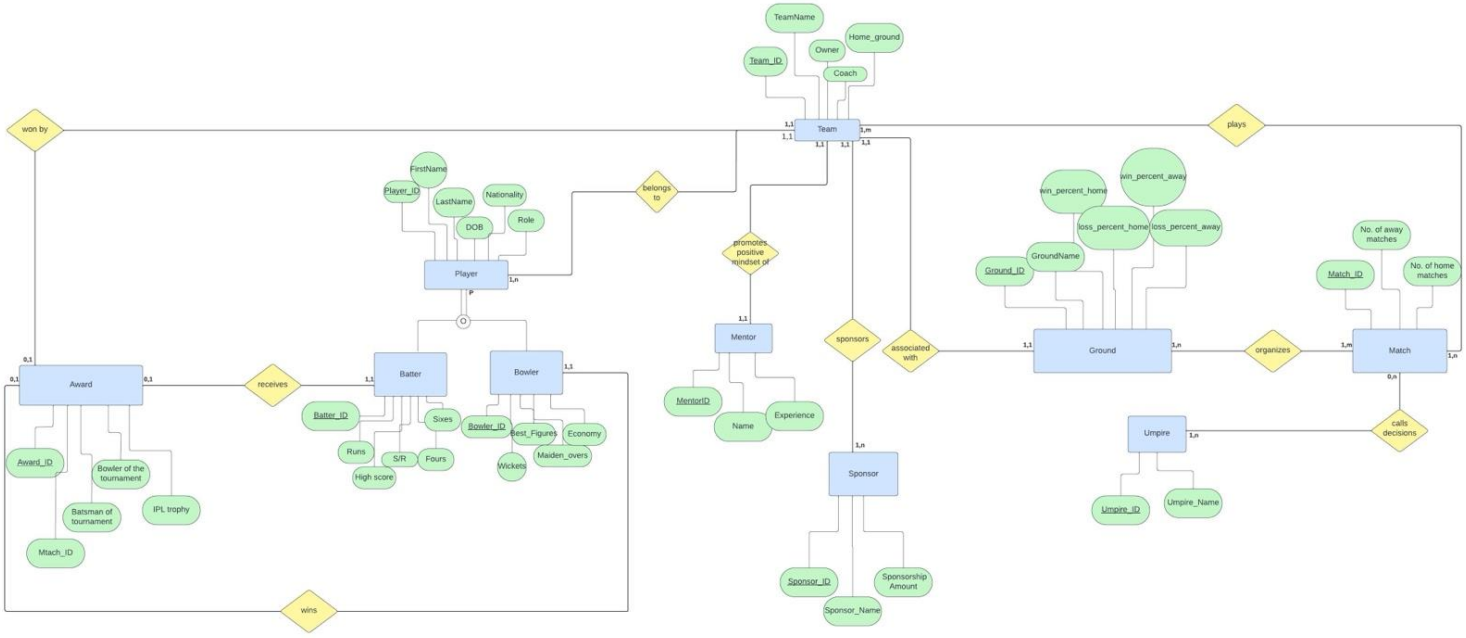
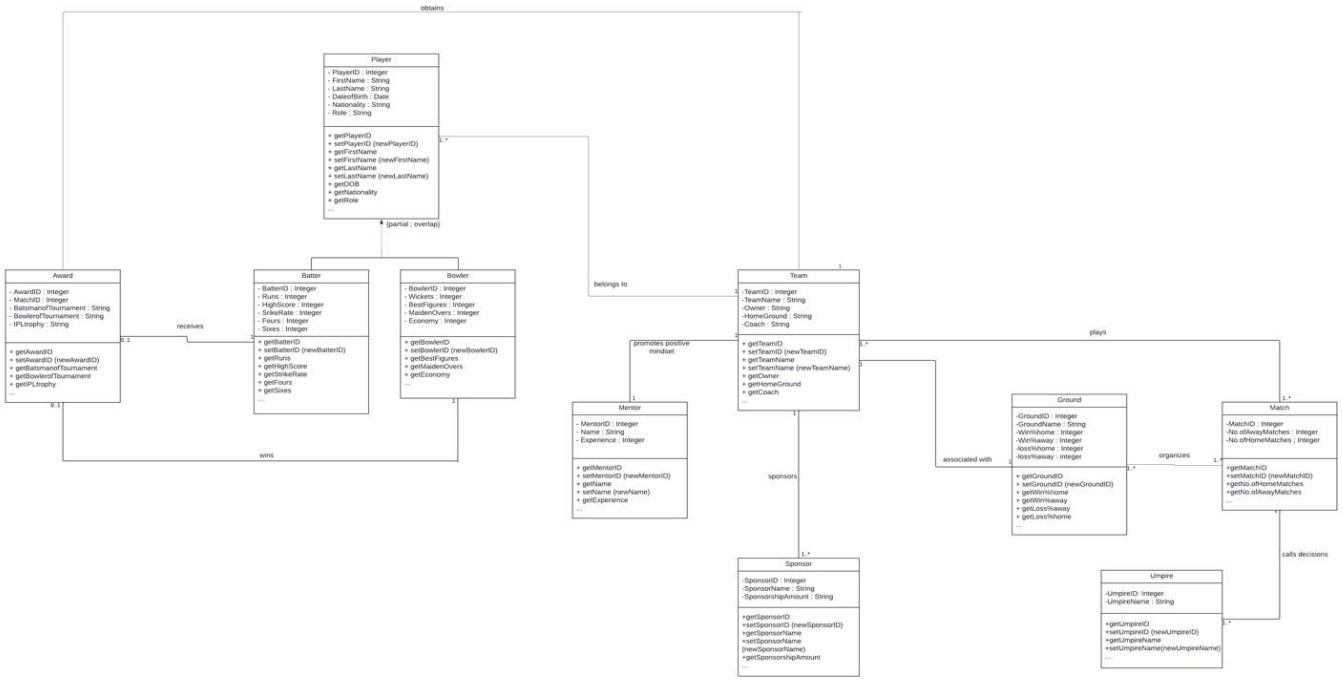
II. CONCEPTUAL MODEL

1. EER Diagram



2. UML Diagram



III. MAPPING CONCEPTUAL MODEL TO RELATIONAL MODEL

Match(Match_ID, no._of_away_matches, no_of_home_matches)

Ground(Ground_ID, GroundName, win_percent_home, loss_percent_home, win_percent_away, loss_percent_away)

Organises(*Match_ID, Ground_ID*)
- *Match_ID* is a foreign key of Match relation and NOT NULL
- *Ground_ID* is a foreign key of Ground relation and NOT NULL

Umpire(Umpire_ID, UmpireName)

Calls_decisions(*Umpire_ID*, *Match_ID*)
- *Umpire_ID* is a foreign key of Umpire relation and NOT NULL
- *Match_ID* is a foreign key of Match relation and NOT NULL

Mentor(Mentor_ID, Mentor_Name, Experience)

Team(Team_ID, TeamName, Owner, Coach, Homeground)

Plays(*Team_ID*, *Match_ID)*
- *Team_ID* is a foreign key of Team relation and NOT NULL
- *Match_ID* is a foreign key of Match relation and NOT NULL

Sponsor(Sponsor_ID, SponsorName, SponsorshipAmount, *Team_ID*)
- *Team_ID* is a foreign key of Team relation and NOT NULL

Player(Player_ID, FirstName, LastName, DOB, Nationality, Role, *Team_ID*)
- *Team_ID* is a foreign key of Team relation and NOT NULL

Batter(Batter_ID, Runs, High_score, S/R, Fours, Sixes, *Player_ID*)
- *Player_ID* is a foreign key of Player relation and NOT NULL

Bowler(Bowler_ID, Wickets, Economy, Maiden_overs, Best_figures, *Player_ID*)
- *Player_ID* is a foreign key of Player relation and NOT NULL

Award(Award_ID, batsman_of_the_tournament, bowler_of_the_tournament, IPL_trophy, *batter_ID*, *bowler_ID*, *Team_ID*)
- *Batter_ID* is a foreign key of Batter relation and NOT NULL
- *Bowler_ID* is a foreign key of Bowler relation and NOT NULL
- *Team_ID* is a foreign key of Team relation and NOT NULL

The primary keys are **Bold and underlined** and the foreign keys are in *Italics*.

IV. IMPLEMENTATION OF RELATIONAL MODEL VIA MYSQL AND NOSQL

- **MySQL Implementation:**

1. **Write a query to select all records from the player table.**

   SELECT * FROM player;

Player_ID	FirstName	LastName	Date_Of_Birth	Nationality	Player_Type	Team_ID
0	Abel	Spencer	1994-08-15	Solomon Islands	Batter	5
1	Erick	Leonard	1991-04-01	Tunisia	All Rounder	2
2	Janice	Montes	1981-06-02	Saint Pierre and Miquelon	All Rounder	4
3	Gretchen	Proctor	1996-02-08	Guadeloupe	All Rounder	1
4	Lawanda	Velazquez	1981-05-05	Uganda	Batter	6
5	Robbie	Wilkins	1991-08-21	Palau	Bowler	6
6	Carla	Randall	1995-05-13	South Africa	All Rounder	0
7	Heath	Dickson	1991-07-03	Cyprus	Bowler	0
8	Kendra	Rodgers	1984-07-18	Aruba	Batter	2
9	Brandie	Finley	1997-01-17	Slovenia	All Rounder	5
10	Rose	Escobar	1993-03-16	Saudi Arabia	All Rounder	5
11	Ernest	Robinson	1987-01-21	Guam	Bowler	0
12	Randal	Larsen	1996-02-18	Sint Maarten (Dutch part)	All Rounder	4
13	Ismael	Dickerson	1981-04-30	Zimbabwe	All Rounder	5
14	Keri	Brennan	1980-08-05	Greenland	Batter	7
15	Cameron	Bass	1988-03-25	Saint Pierre and Miquelon	Bowler	1

2. **Write a query to select the names of all IPL teams from the Team table.**

   SELECT TeamName FROM team;

TeamName
Lucknow Giants
Mumbai Indians
Kolkata Knight Riders
Delhi Capitals
Mumbai Indians
Punjab Kings
Chennai Super Kings
Punjab Kings

3. **Write a query to select the total average of maiden overs using AVG operator.**

   SELECT AVG(Maiden_Overs)
   AS Maiden_Over FROM Bowler;

Maiden_Over
31.3467

4. **Write a query to count the number of players who scored more than 10,000 runs.**

   SELECT COUNT(Player_ID) AS Players
   FROM Batter
   WHERE Runs > 10000;

Players
95

5. **Write a query to retrieve the concatenated name, nationality, runs, high score, and wickets of players who are both batters and bowlers, ordered by nationality.**

```
SELECT CONCAT(P.FirstName," ",P.LastName) AS Name, P.Nationality, B.Runs,
B.High_Score, Bo.Wickets
FROM Player P
JOIN Batter B ON P.Player_ID = B.Player_ID
JOIN Bowler Bo ON P.Player_ID = Bo.Player_ID
ORDER BY P.Nationality;
```

Name	Nationality	Runs	High_Score	Wickets
Bobby Harrison	Antarctica	14133	75	95
Guadalupe Weber	Antarctica	10874	54	64
Mario Pham	Antigua and Barbuda	19741	94	147
Austin Briggs	Argentina	8430	98	42
Roberta Morse	Argentina	11395	119	69
Anne Lester	Armenia	5228	106	12
Elisabeth Cross	Barbados	10179	105	58
Nina Vaughn	Barbados	12373	137	78
Neil Chavez	Belgium	6809	61	26
Tamiko Church	Bulgaria	18093	117	20
Bennie Munoz	Bulgaria	5014	124	104

6. **Write a query to retrieve all the players who are all-rounders (batter and bowler)**

```
SELECT *
FROM Batter
WHERE player_id IN (
   SELECT Player_ID
        FROM Player
        WHERE Player_Type = "All Rounder"
       );
```

Player_ID	FirstName	LastName	Date_Of_Birth	Nationality	Player_Type	Team_ID
0	Abel	Spencer	1994-08-15	Solomon Islands	Batter	5
4	Lawanda	Velazquez	1981-05-05	Uganda	Batter	6
5	Robbie	Wilkins	1991-08-21	Palau	Bowler	6
7	Heath	Dickson	1991-07-03	Cyprus	Bowler	0
8	Kendra	Rodgers	1984-07-18	Aruba	Batter	2
11	Ernest	Robinson	1987-01-21	Guam	Bowler	0
14	Keri	Brennan	1980-08-05	Greenland	Batter	7
15	Cameron	Bass	1988-03-25	Saint Pierre and Miquelon	Bowler	1
18	Cornelius	Herring	1998-11-28	Montenegro	Bowler	7
19	Teddy	Martin	1997-04-18	Bermuda	Bowler	6
21	Dianna	Chapman	1994-03-28	Svalbard and Jan Mayen	Bowler	4

7. **Write a query that retrieves the teams whose home ground has a win percentage higher than the average win percentage of all home grounds.**

TeamName
Lucknow Giants
Mumbai Indians
Kolkata Knight Riders
Delhi Capitals

```
SELECT TeamName
FROM Team t
WHERE EXISTS (
   SELECT 1
   FROM Ground g
   WHERE TRIM(g.GroundName) = TRIM(t.Homeground)
   AND g.Win_Percent_Home > (SELECT AVG(Win_Percent_Home) FROM Ground)
   );
```

8. **Write a query to select all the players who are not classified as all-rounders.**

```
SELECT *
FROM Player as p
WHERE NOT EXISTS(
        SELECT 1
   FROM Player as p1
   WHERE Player_Type = 'All Rounder' AND
   p1.Player_ID = p.Player_ID
);
```

Batter_ID	Player_ID	Runs	High_score	Strike_Rate	Fours	Sixes
0	34	15967	123	204.204	221	221
1	140	18521	57	236.181	35	178
2	142	12452	56	236.163	151	235
3	142	19788	126	195.525	143	43
4	84	17856	72	93.3259	140	108
7	108	6121	52	216.839	230	269
8	120	16147	131	105.709	73	14
9	13	19243	144	237.179	276	280
10	6	5457	65	209.852	266	160
13	41	5851	103	242.557	137	297
14	80	11124	147	220.02	123	208
18	80	12139	94	127.671	81	110

9. **Write a query to retrieve players who are either from New Zealand or Zimbabwe.**

```
SELECT Player_ID, FirstName, LastName, Nationality
FROM Player
WHERE Nationality = "New Zealand"
UNION
SELECT Player_ID, FirstName, LastName,
Nationality
FROM Player
WHERE Nationality = "Zimbabwe";
```

Player_ID	FirstName	LastName	Nationality
47	Caroline	Snyder	New Zealand
13	Ismael	Dickerson	Zimbabwe
128	Aisha	Valdez	Zimbabwe

10. **Write a query to retrieve the first names of players along with their corresponding average runs.**

```
SELECT FirstName, batting_stats.average_runs
FROM Player,
    (SELECT Player_ID, AVG(Runs) AS average_runs
    FROM batter
    GROUP BY player_id) AS batting_stats
 WHERE player.player_id = batting_stats.player_id;
```

FirstName	average_runs
Lucas	15967.0000
Valerie	18521.0000
Brandy	16120.0000
Regina	12875.0000
Jami	16395.0000
Herman	8422.0000
Bridgett	6534.3333
Pablo	16147.0000
Ismael	19243.0000
Carla	10426.2500
Bobbi	15374.0000
Sonya	14826.0000
Luz	6213.0000
Judith	11631.5000
Dianna	15084.5000
Jasmine	14570.5000

- **NoSQL Implementation:**

1. **Write a query to select players who scored more than 10,000 runs.**

```
Indian_Premier_League> db.Batter.find({
...   Runs: { $gt: 10000 }
... }).count()
```

Output:

```
95
```

2. **Write a query to select players who are either from New Zealand or Zimbabwe (using the aggregate method).**

```
Indian_Premier_League> db.getCollection("Player").aggregate([
...   {
...     $match: {
...       $or: [
...         { Nationality: "New Zealand" },
...         { Nationality: "Zimbabwe" }
...       ]
...     }
...   },
...   {
...     $project: {
...       Player_ID: 1,
...       FirstName: 1,
...       LastName: 1,
...       Nationality: 1,
...       _id: 0
...     }
...   }
... ])
```

Output:

```
[
  {
    Player_ID: 13,
    FirstName: 'Ismael',
    LastName: 'Dickerson',
    Nationality: 'Zimbabwe'
  },
  {
    Player_ID: 47,
    FirstName: 'Caroline',
    LastName: 'Snyder',
    Nationality: 'New Zealand'
  },
  {
    Player_ID: 128,
    FirstName: 'Aisha',
    LastName: 'Valdez',
    Nationality: 'Zimbabwe'
  }
]
```

3. **Write a query to select the mentors who have an experience greater than 5 years (using MapReduce).**

```
Indian_Premier_League> var mapFunction = function() {
...     if (this.Experience > 5) {
...         emit(this.Mentor_Name, this.Experience);
...     }
... };

Indian_Premier_League>

Indian_Premier_League> var reduceFunction = function(key, values) {
...     // For simplicity, assuming there's only one document per key
...     return values[0];
... };

Indian_Premier_League>

Indian_Premier_League> db.Mentor.mapReduce(
...     mapFunction,
...     reduceFunction,
...     {
...         out: { replace: "filteredMentors" }
...     }
... );
{ result: 'filteredMentors', ok: 1 }
Indian_Premier_League>

Indian_Premier_League> db.filteredMentors.find();
```

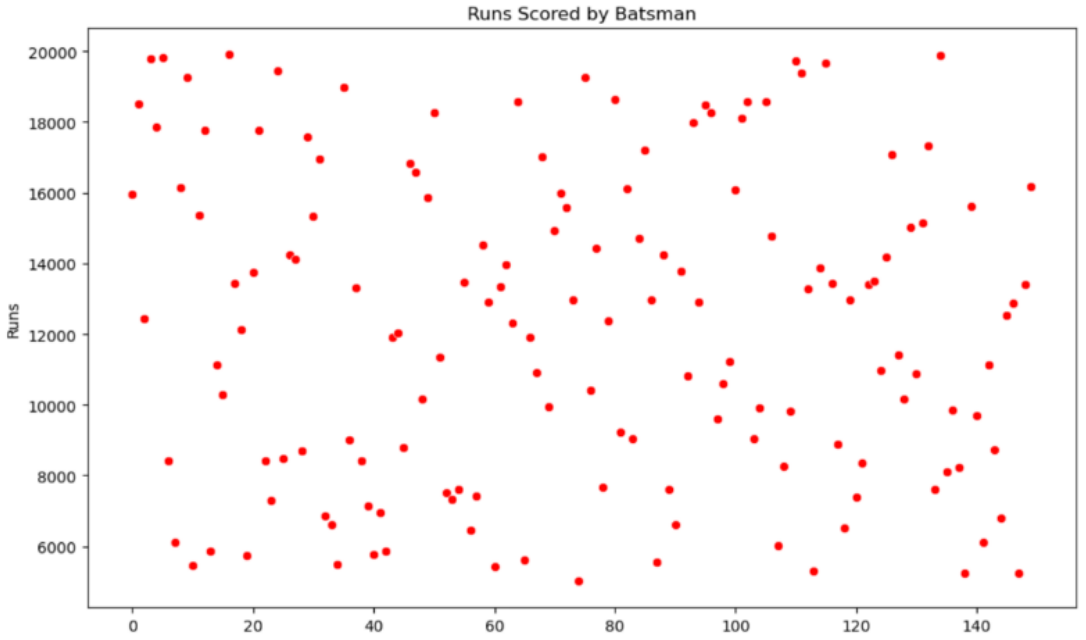
Output:

```
[
  { _id: 'Abel', value: 7 },
  { _id: 'Robbie', value: 9 },
  { _id: 'Erick', value: 9 },
  { _id: 'Randal', value: 8 }
]
```

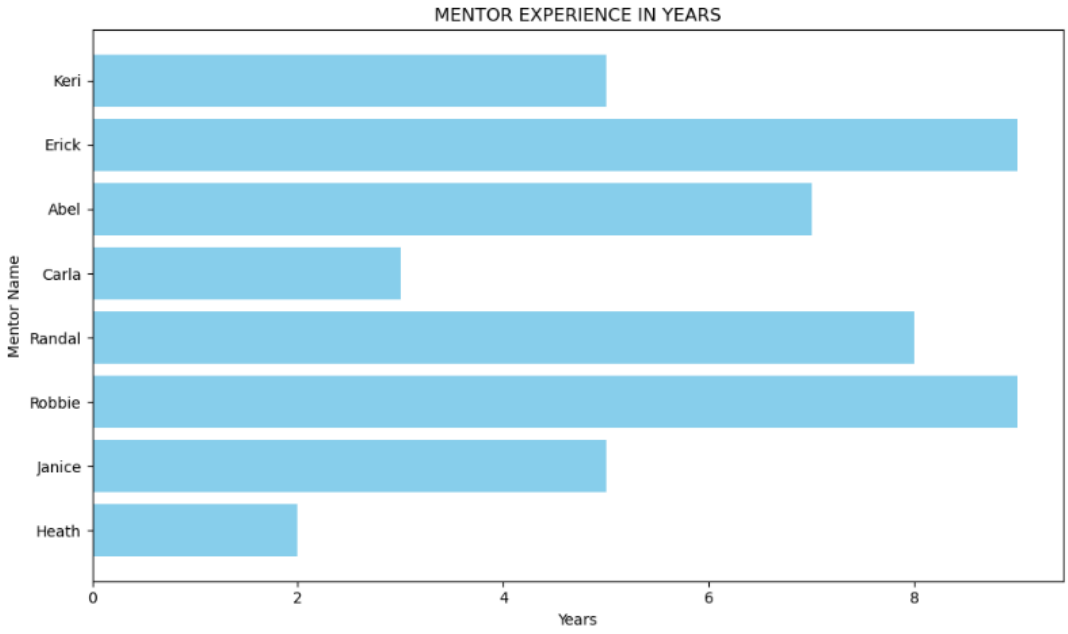
V. Database Access Via Python

In this section, we will demonstrate the successful integration of our database using Python in Jupyter Notebook Environment. There are numerous advantages of connecting the database in Python such as the powerful ecosystem of libraries, including Pandas for data manipulation and Matplotlib for visualization. This facilitates seamless integration and analysis of database data, optimizing data analysis and visualization workflows.

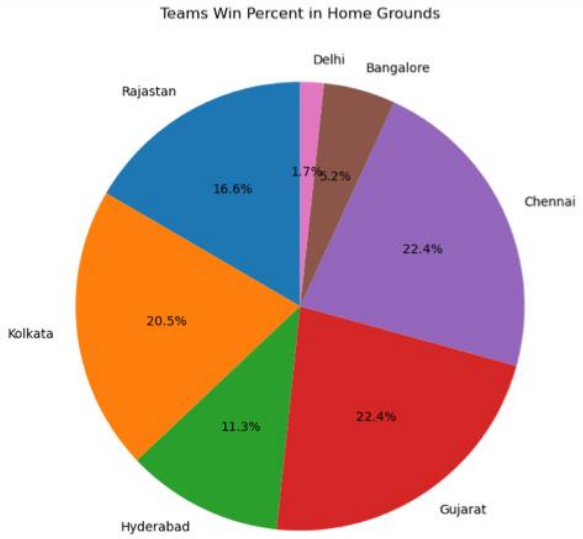
Graph 1: Scatter plot displaying number of runs scored by the batter.



Graph 2: Bar plot displaying the mentor experience



Graph 3: Pie Chart displaying the team's winning percentage in their respective home grounds.



VI. Summary and Recommendation

Based on the key trends we visualized from the historical data collected from multiple IPL seasons, we can conclude that Chennai Super Kings and Mumbai Indians are the two teams which showed comparatively higher consistency rate as both these teams have bagged multiple trophies. Also, it was observed that the more successful teams were able to get considerably higher sponsorship amount which benefitted the team greatly in their success journey. The atmosphere in the dressing room, the rapport of players with the coaching staff, the role of mentor in fostering a positive mindset, were among many other factors contributing to the success of these teams.

The study also highlights the necessity of further development to improve the system's capabilities and open up new doors for success in the constantly changing IPL landscape. These developments include the integration of real-time data, the application of machine learning, and interactive visualizations.

It is advised to give real-time data integration top priority to improve the IPL team performance analysis for the system's efficacy and enable prompt decision-making during games. Furthermore, investigating the incorporation of machine learning algorithms can offer a prospective viewpoint for player performance forecasting and predictive analysis. Interactive visualizations will make it easier to interpret data and encourage more in-depth interaction with the analysis's findings. Putting in place a procedure for ongoing data enrichment guarantees that the system remains up to date and can adjust to shifting team compositions and outside factors. Maximizing the system's relevance and adoption through cooperation with IPL teams to customize it to their unique requirements and offering user training sessions will support the system's ongoing development and success in the competitive cricket scene.