

## **Executive Summary:**

The primary objective of this study is to design and develop a relational database that is industry ready for application in sports club management from a management perspective which is used by sports club organizations who are constantly complaining about the process of reviewing the players, maintaining a hectic schedule of matches, looking over medical and coaching staff, sponsorships and advertisements. There were a lot of data duplications and data integrity problems.

This relational database reduces the data input process by 50% and results in huge cost-saving benefits across the industry. The database also implements a central analytics platform that has immense potential for analytics on the coaching and medical teams, studies players' performance preferred selections of players, and also maintains a tab on the sponsorships in these clubs.

The database was modeled taking the requirements of data fields required by the Sports club's organizers and their sectors. The EER and UML diagrams, followed by the mapping of the conceptual model to a relational model with the required primary and foreign keys. This database was then implemented fully using MySQL in the SQL workbench application. We have also developed its application in Python and the Big data tool Mongo DB.

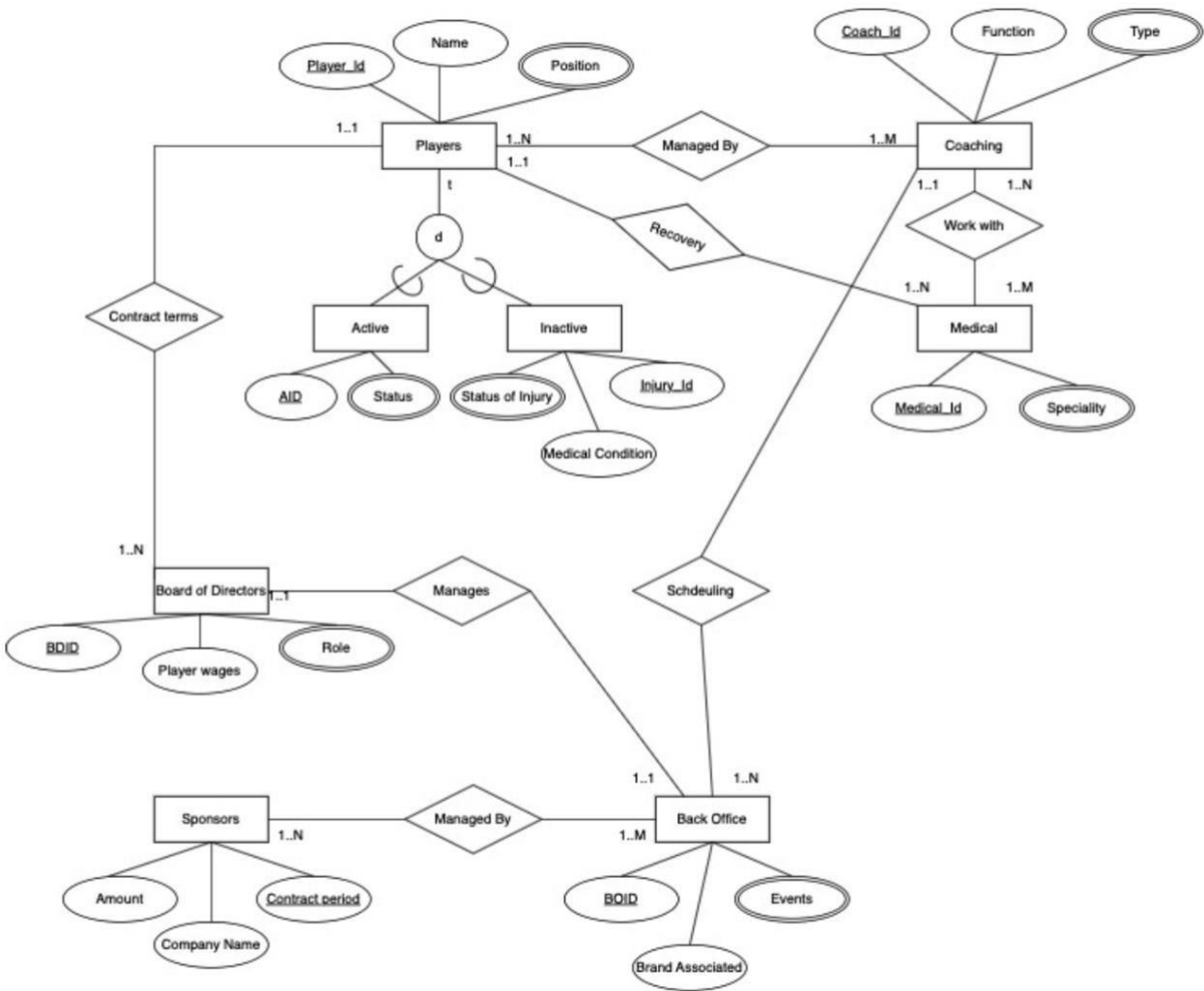
## **I. Introduction**

In this project, we will design and implement a soccer club management system from a management perspective which will store the information used by soccer club management and organizers. The purpose of our project is to provide an efficient and robust way of storing information in a well-defined and structured database system that can be used by people of various levels in the organization like the Board of Directors, Players, Coaches, Medics, and Sponsors.

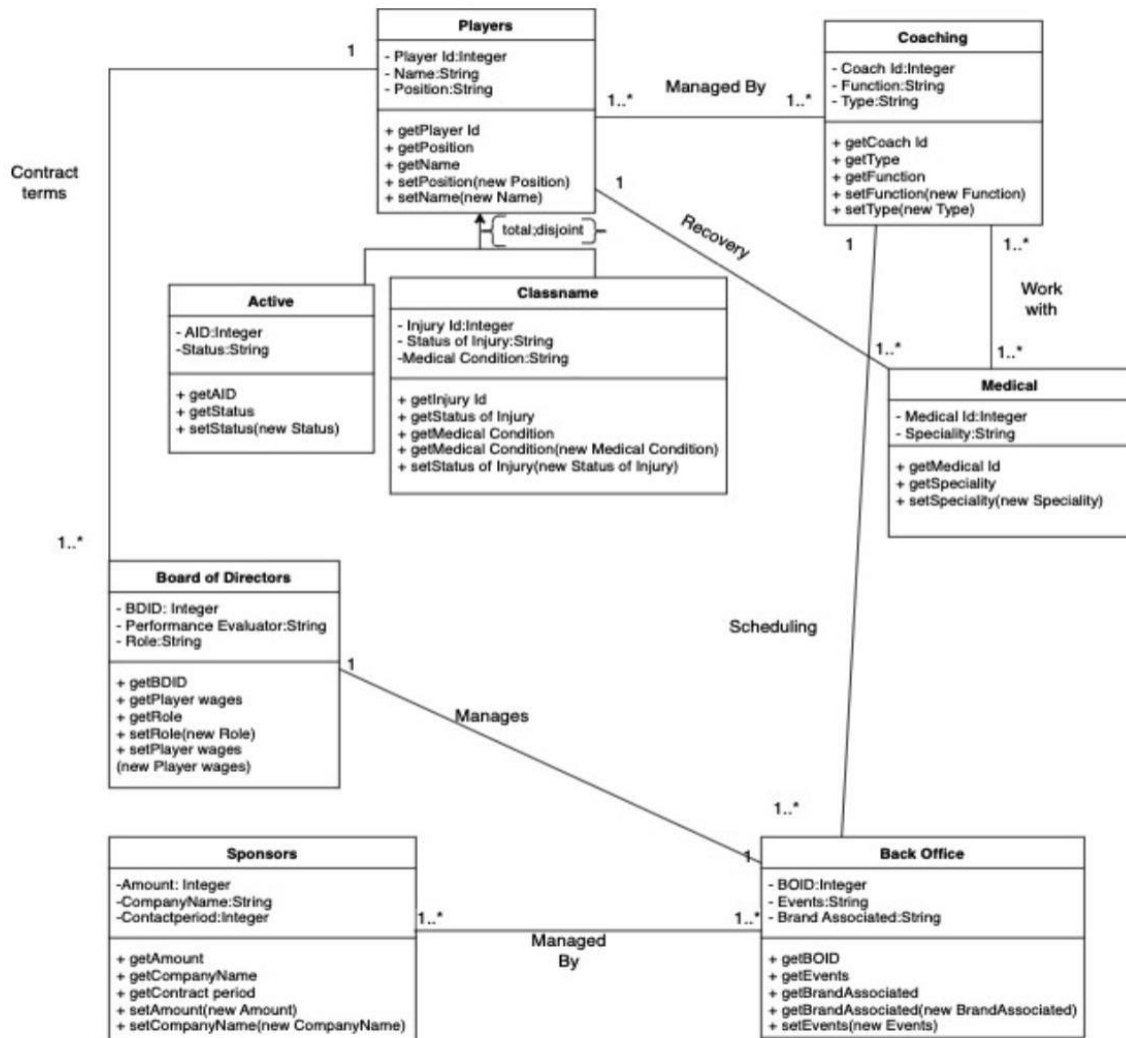
In this database system, the performance of players is recorded, which can be analyzed and used by directors of teams, coaches, and agents. This application is also useful for the coaches since it can project some important statistics about the teams and players. They can also find the physical status of a player. Considering the stats and status, the management and coaches can take valuable decisions about the players throughout the league.

# II. Conceptual Data Modeling

## 1. EER Diagram



## 2. UML Diagram



### III. Mapping Conceptual Model to Relational Model

- For the following relation, the Primary key is denoted as the Primary key, and the foreign key is denoted as **Foreign Key**.
- Players (PlayerID, Name, Age)  
 Coaching (CoachID, Name)  
 Medical (MedicalID, **PlayerID (Not Null)**)  
 Board of Directors (BDID, Player\_wages, Role)

- Sponsors (Period, Amount, SName)  
 Back Office (BOID, Events, Brand\_Assocaited, **CoachID (NotNull)**)
- Active (PlayerID, Status)  
 Inactive (PlayerID, Status\_of\_Injury, Type\_of\_Injury)  
 Work\_with (**CoachID (Not Null)**, **MedicalID (Not Null)**)
  - Manages\_coaching (**PlayerID (Not Null)**, **CoachID (Not Null)**)
  - Managed\_By (**Period (Not Null)**, **BOID (Not Null)**)
  - Player-Position (PlayerID, Position)
  - For Player-Position, the combination of PlayerID and Position form the primary key.

## IV. Implementation of Relation Model via MySQL and NoSQL

### MySQL Implementation:

The database was created in MySQL and the following queries were performed:

**Query 1: Find number of 'Reserve' players in the squad**

***SELECT count(\*) AS COUNT FROM Active where Status = 'Reserves';***

	COUNT
▶	16

**Query 2: Find number of players in a position having more than seven players in that position(Query returns count of all position except GK) select Position, count(Position) as Number from Player\_Position group by Position having count(Position) >= 7;**

	Position	Number	
►	Midfielder	10	
	Forward	8	
	Defender	13	

**Query 3: Find Name of Sponssor with amount and id of people from Back Office associated with the sponsor**

```
select r.Amount, r.SName, s.BOID from Sponsors
as r left outer join Managed_By as s on
(r.Period=s.Period);
```

	Amount	SName	BOID	
►	282657	Adidas	106	
	282657	Adidas	109	
	282657	Adidas	110	
	292536	Nike	101	
	292536	Nike	102	
	292536	Nike	104	
	292536	Nike	106	
	292536	Nike	108	
	511776	Adidas	103	
	511776	Adidas	108	
	511776	Adidas	109	
	1650907	Under...	101	
	1650907	Under...	102	
	1650907	Under...	105	
	1650907	Under...	108	
	1650907	Under...	109	
	1234541	Puma	102	
	1206075	Nike	103	
	1206075	Nike	108	
	1206075	Nike	110	

**Query 4: Find Type of Injury using player id of the injured player. SELECT**

```
Type_of_injury FROM Inactive
where PlayerID =
(select PlayerID from
Players where
PlayerID ='40');
```

	Type_of_injury	
▶	Knee Sprain	

**Query 5: Find name of medical staff working with players who are not part of the First Team.(The first eleven records in the database are of the first team players)**

Select p.Name from Medical p where 11 <

(select PlayerID

from Players s

where p.PlayerID = s.PlayerID);

	Name	
▶	Kimberley Rojas	
	Flynn Schroeder	
	Forrest Blankenship	

**Query 6: Find Status of players playing in the 'Forward' position from Active table.(Active table contains records of players who are not injured)**

Select Status from Active p

where exists (select Position

from Player\_Position s where

p.PlayerID = s.PlayerID and

Position = 'Forward');

	Status	
▶	First Team	
	Reserves	
	Reserves	
	Substitutes	
	Substitutes	
	First Team	
	First Team	
	First Team	

**Query 7: Find all players playing in forward position and players from Reserves.**

```

select PlayerID, Position
from Player_Position
where Position = 'Forward'
UNION
select s.PlayerID, s.Status from
Player_position p,Active s
where p.PlayerID = s.PlayerID
and Status ='Reserves';

```

	PlayerID	Position
▶	3	Forward
	15	Forward
	20	Forward
	23	Forward
	26	Forward
	31	Forward
	34	Forward
	35	Forward
	1	Reserves
	2	Reserves
	8	Reserves
	10	Reserves
	11	Reserves
	15	Reserves
	20	Reserves
	21	Reserves
	24	Reserves
	25	Reserves
	27	Reserves
	28	Reserves
	29	Reserves
	30	Reserves
	32	Reserves
	33	Reserves

**Query 8: Find ID, Name and Age of all players whose age is more than 20 years**

**old** select s.PlayerID, s.Status, p.Age from Active s, Players p

where s.PlayerID = p.PlayerID and p.Age >

(select Age from

Players

where Age >= 20);

	PlayerID	Status	Age	
▶	3	First Team	31	
	4	First Team	25	
	5	First Team	31	
	6	Substitutes	36	
	7	Substitutes	26	
	8	Reserves	24	
	9	First Team	26	
	10	Reserves	28	
	11	Reserves	23	
	12	First Team	29	
	13	Substitutes	35	
	14	First Team	30	
	15	Reserves	24	
	16	Substitutes	25	
	17	First Team	35	
	18	First Team	30	
	19	Substitutes	26	
	20	Reserves	22	
	21	Reserves	30	
	23	Substitutes	33	
	24	Reserves	24	
	25	Reserves	21	
	27	Reserves	35	
	28	Reserves	33	
	29	Reserves	26	
	30	Reserves	26	
	31	First Team	26	
	32	Reserves	23	
	34	First Team	32	
	35	First Team	35	



## NoSQL Implementation:

All tables have been created in MongoDB playground. The following queries were done:

### Query 1: Find number of players with respective to their age.

```
db.Players.aggregate([{$group: {_id:"$Age", PlayerID:{$sum:1}}}]])
```

Players > PlayerID	
_id	PlayerID
35	4.0
30	5.0
28	1.0
29	1.0
21	1.0
26	8.0
33	4.0
20	1.0
19	1.0
18	3.0
24	3.0
22	1.0
17	1.0
13	1.0
27	1.0
31	2.0
16	3.0
25	2.0

**Query 2: Find all records from Players table who are 18 years old.**

`db.Players.find({ Age:18 })`

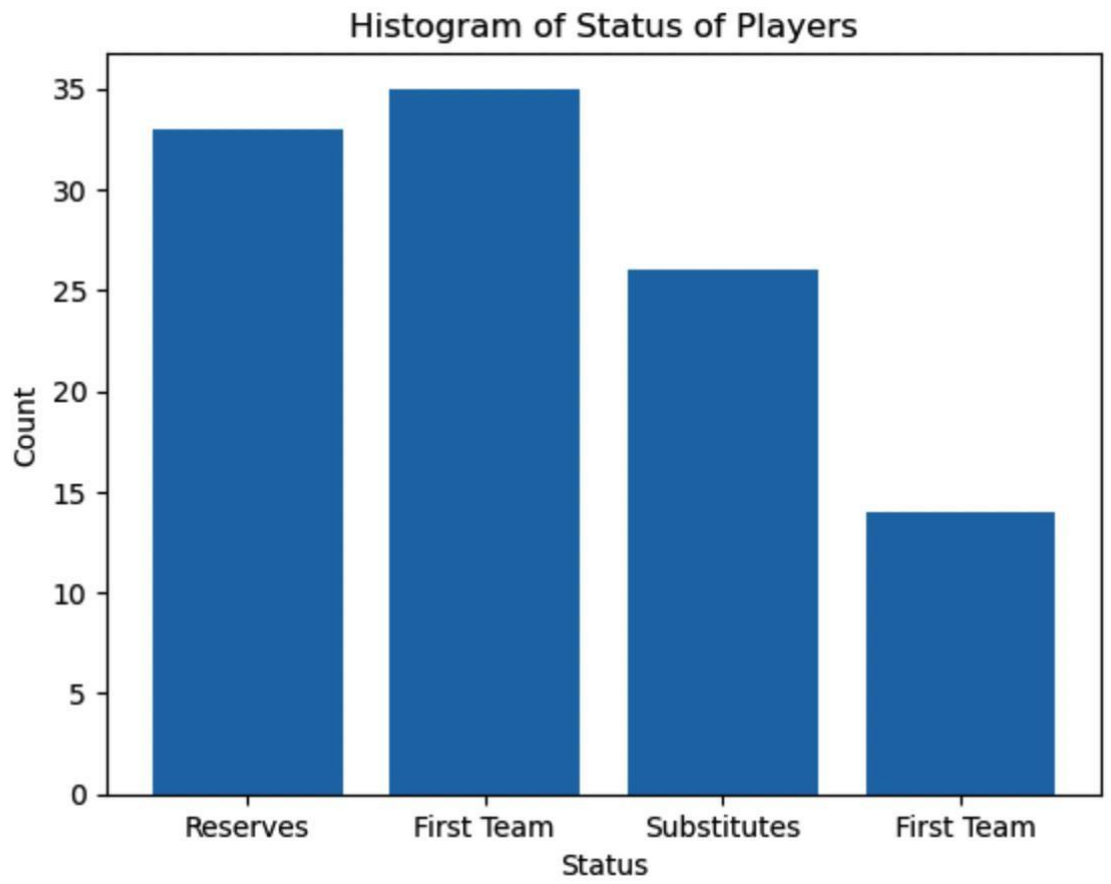
Players > PlayerID		
PlayerID	Name	Age
1	Nero	18
26	Ethan	18
46	Quinlan	18

**Query 3: Find all records from Medical Table.**

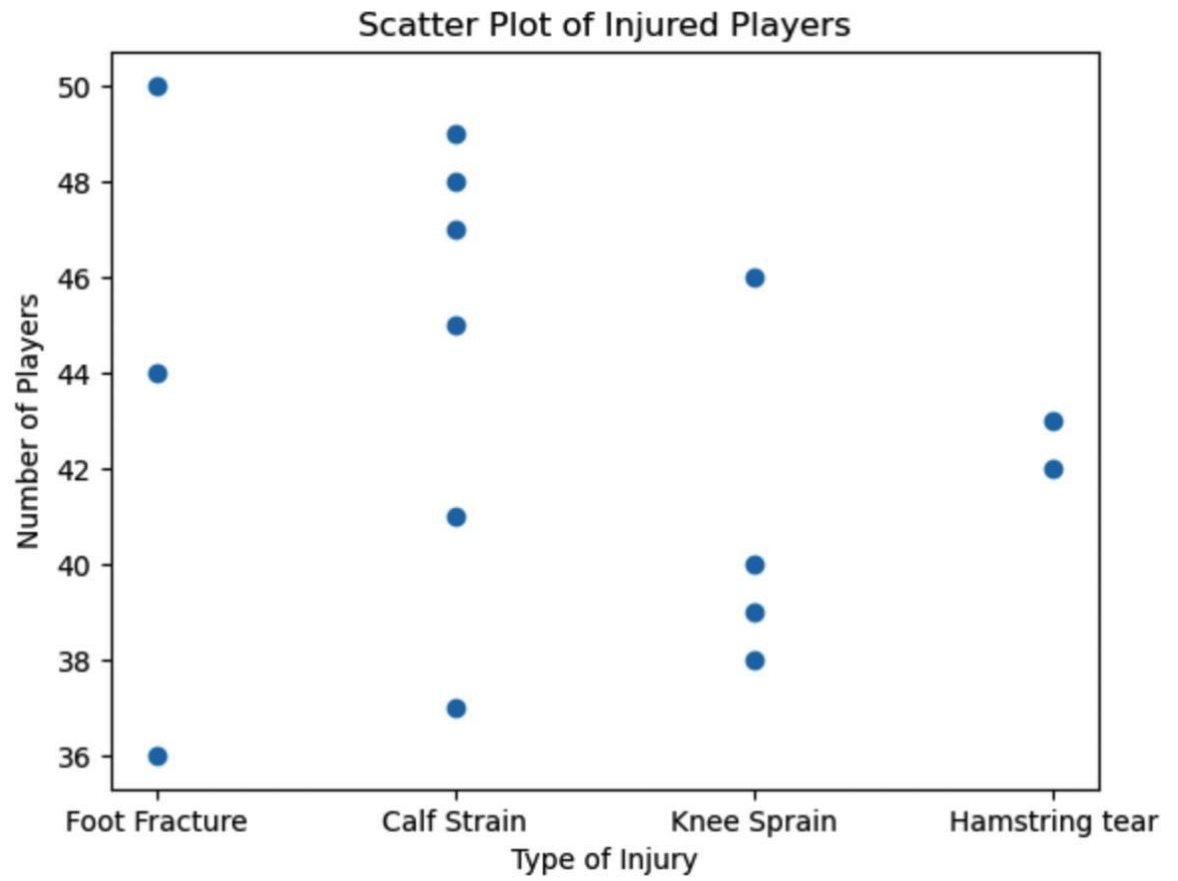
Medical > MedicalID		
MedicalID	Name	PlayerID
17	Kimberley Rojas	32
20	Catherine Herma	1
39	Chadwick Gray	6
60	Flynn Schroeder	37
74	Forrest Blankens	22
99	Hilel Wiley	2

## V. Database Access via Python

The database is accessed using Python and visualization of analyzed data is shown below. The connection of MySQL to Python is done using `mysql.connector`, followed by `cursor.execute` to run and `fetchall` from query, followed by converting the list into a dataframe using `pandas` library and using `matplotlib` to plot the graphs for the analytics.



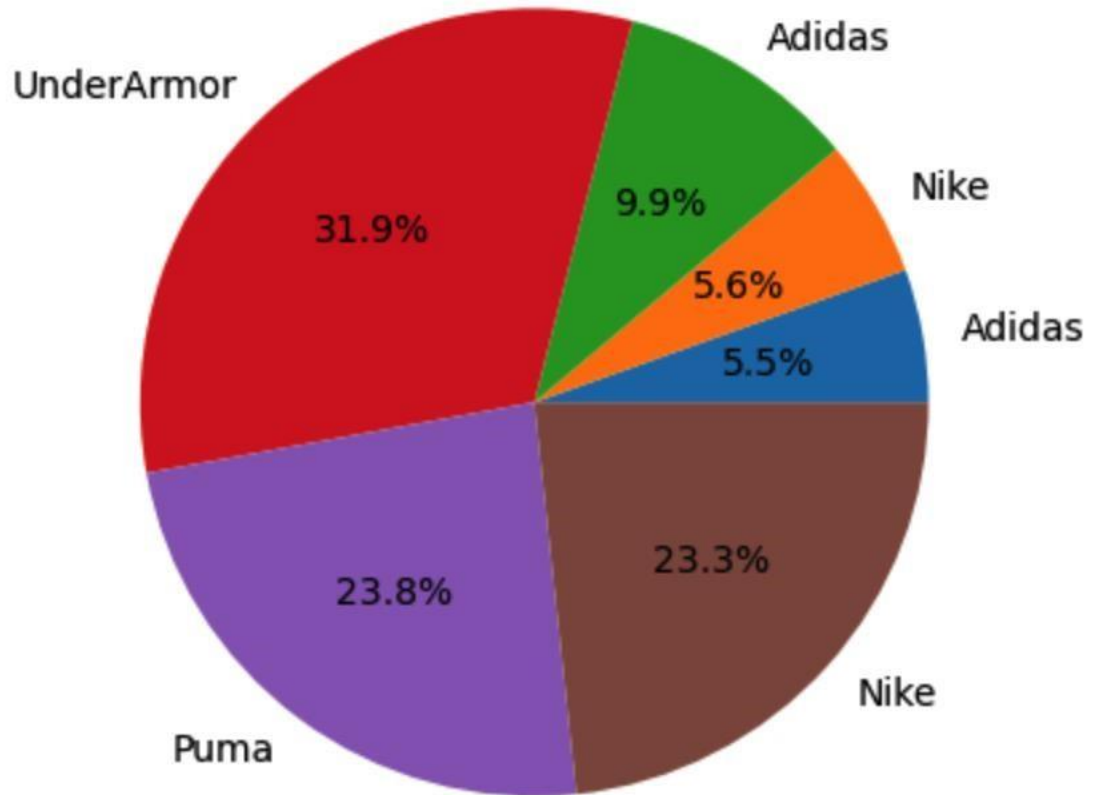
**Graph1: Histogram of Status of Players**



**Graph2: Scatter Plot of Injured Players**

---

## Sponsorships



**Graph 3: Pie Chart of Sponsorship distribution**

## **VII. Summary and Recommendation**

To sum up, this database system will present a large amount of data in a very easy way so that it is easy to search through the website for particular matches, players, coaches, medics, Board of Directors, and Admin Staff. It will result in great time saving and provides easy access to information for all required.