KATHMANDU UNIVERSITY

SCHOOL OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MINI PROJECT

REPORT ON

Premier League Statistics 2016/17 DBMS

A second year mini project report submitted in partial fulfillment

of the requirements for COMP 232

by:

Sunil Prajapati (37)

Ayush Kumar Shah (44)

Bibash Shrestha (48)

CE 2nd year

Date of Submission : 10th August 2017

Premier League Teams Statistics of 2016/17

Description:

Objective:

1. To understand the concept and utilization of database management system.
2. To develop a database regarding the game statistics of Premier League teams and players
3. To create a simple java application to perform and demonstrate the basic operations on the created database including insert, delete, update and search.

System Design:

1. Programming Language: Java
2. Database Management System: MySQL
3. Tools used: phpMyAdmin, Netbeans

Scenario:

This project has been introduced primarily to the fans of the beautiful game of football. The system is useful for the dedicated fans of the English Premier League who wish to view the different form of statistics from their favorite **teams** and **players**. The process of acquiring various interesting data related to respective parts of the game is quite tiring and usually involves switching between tabs in browser surfing the numerous sports websites. Therefore, this project aims to help the fellow football fans to acquire the statistics about on and off the pitch. The system will be built on Java UI Platform and have a user friendly interface that can be used and navigated without any problems. The user will be able to navigate through the different options to view the statistics like **goals**, **passing**, **shooting**, **transfer activity** and many others as well. The users can also view the **revenues** of a club as well as **standings** and **points** earned by the particular club throughout the season.

Schemes:

The user will be able to search through the statistics of teams and players using either the player name or club name and retrieve the information they need. They can also insert new records into the database or delete a record from the database. The users also have the option to update the values of some of the attributes as per requirement.

Entities and their attributes

1. **Team**: consists information about different football teams
    - Club_id
    - Club_name
    - Stadium
    - Manager

2. **Player**: consists information about different football players
    - Player_id
    - Club_id
    - Player_name
    - Jersey_no
    - Position

3. **Goals_Team**: consists information about number of goals scored by the team
    - Goals_Team_id
    - Club_id
    - Goals_for
    - Goals_conceded
    - Goal_difference
    - Home_Goals
    - Away_Goals

4. **Attacking_Output_Player**: consists information about attacking output of the player
    - Attacking_id
    - Player_id
    - Goals
    - Assists
    - Attacking_Output

5. **Fouls_Team**: consists information about fouls conceded by the team
    - Fouls_Team_id
    - Club_id
    - Yellow Cards
    - Red Cards
    - Total Cards

6. **Fouls_Player:** consists information about fouls conceded by the player
    - Fouls_Player_id
    - Player_id
    - Yellow Cards
    - Red Cards

- Total Cards

7. **Points_Home:** consists information about points earned by the team at home stadium
    - Points_home_id
    - Club_id
    - Points_won
    - Points_dropped
    - Percentage

8. **Points_Away**: consists information about points earned by the team at away stadium.
    - Points_Away_id
    - Club_id
    - Points_won
    - Points_dropped
    - Percentage

9. **Passing_Team**: consists information about passing statistics of the team
    - Passing_Team_id
    - Club_id
    - Pass_completion
    - Pass_accuracy

10. **Passing_Player:** consists information about passing statistics of the player
    - Passing_Player_id
    - Player_id
    - Pass_completion
    - Pass_accuracy

11. **Set_Pieces_Team**: consists information about set pieces scored by the team
    - Set_pieces_id
    - Club_id
    - Free_kicks
    - Penalties
    - Corners
    - Percentage

12. **Shots_Team:** consists information about shooting statistics of the team
    - Shots_id
    - Club_id
    - Total_shots
    - Shots_on_target
    - Shot_accuracy

13. **Revenues:** consists information about revenues earned by the team
    - Revenue_id
    - Club_id
    - Match_day
    - Broadcast
    - Commercial
    - Total

14. **Standings:** consists information about the standings (rank) of the team in the league.
    - Standings_id
    - Club_id
    - Points
    - Standings

Functional dependencies(F) and Closure of F (F^+)

1. Team (Club_id, Club_name, Stadium, Manager)

   Keys→ Club_id, Club_name

   F={Club_id→ Stadium, Manager, Club_name
   Club_name→ Stadium, Manager}

   F^+={Club_id→ Stadium, Manager, Club_name
   Club_name→ Stadium, Manager
   Club_id→ Stadium
   Club_id→ Manager
   Club_id→ Club_name
   Club_name→ Stadium
   Club_name→ Manager}

2. Player (Player_id, Club_id, Player_name, Jersey_no, Position)

   Keys→ Player_id,Player_name

   F={Player_id→ Player_name, Jersey_no, Position, Club_id
   Player_name→ Jersey_no, Position, Club_id}

   F^+={Player_id→ Player_name, Jersey_no, Position, Club_id
   Player_name→ Jersey_no, Position, Club_id
   Player_id→ Player_name
   Player_id→ Jersey_no
   Player_id→ Position
   Player_id→ Club_id
   Player_name→ Jersey_no
   Player_name→ Position
   Player_name→ Club_id}

3. Goals_Team (Goals_Team_id, Club_id, Goals_for, Goals_conceded, Goal_difference, Home_Goals, Away_Goals)

   Keys→ Goals_Team_id, Goals_for, Goals_conceded
   F={Goals_Team_id→ Club_id, Goals_for, Goals_conceded, Goal_difference,
   Home_Goals, Away_Goals
   Goals_for, Goals_conceded → Goal_difference}

F^+={Goals_Team_id→ Club_id, Goals_for, Goals_conceded, Goal_difference, Home_Goals, Away_Goals
Goals_for, Goals_conceded → Goal_difference
Goals_Team_id→ Club_id
Goals_Team_id→ Goals_for
Goals_Team_id→ Goals_conceded
Goals_Team_id→ Goal_difference
Goals_Team_id→ Home_Goals
Goals_Team_id→ Away_Goals
Goals_Team_id, Goals_conceded → Goal_difference
Goals_for, Goals_Team_id → Goal_difference}

4. Attacking_Output_Player (Attacking_Player_id, Player_id Goals, Assists, Attacking_Output)

Keys→ Attacking_Player_id, Goals, Assists

F={Attacking_Player_id→ Player_id, Goals, Assists, Attacking_Output
Goals, Assists → Attacking_Output}

F^+={Attacking_Player_id→ Player_id, Goals, Assists, Attacking_Output
Goals, Assists → Attacking_Output
Attacking_Player_id→ Player_id
Attacking_Player_id→ Goals
Attacking_Player_id→ Assists
Attacking_Player_id→ Attacking_Output
Attacking_Player_id, Assists → Attacking_Output
Goals, Attacking_Player_id → Attacking_Output}

5. Fouls_Team (Fouls_Team_id, Club_id, Yellow Cards, Red Cards, Total Cards)

Keys→Fouls_Team_id, Yellow Cards, Red Cards

F={Fouls_Team_id→ Club_id, Yellow Cards, Red Cards, Total Cards
Yellow Cards, Red Cards → Total Cards}

F^+= {Fouls_Team_id→ Club_id, Yellow Cards, Red Cards, Total Cards
Yellow Cards, Red Cards → Total Cards
Fouls_Team_id→ Club_id
Fouls_Team_id→ Yellow Cards
Fouls_Team_id→ Red Cards
Fouls_Team_id→ Total Cards
Fouls_Team_id, Red Cards → Total Cards
Yellow Cards, Fouls_Team_id → Total Cards}

6. Fouls_Player (Fouls_Player_id, Player_id,, Yellow Cards, Red Cards, Total Cards)

Keys→Fouls_Player_id, Yellow Cards, Red Cards

F= {Fouls_Player_id→ Player_id , Yellow Cards, Red Cards, Total Cards
Yellow Cards, Red Cards → Total Cards}

F^+= {Fouls_Player_id→ Player_id, Yellow Cards, Red Cards, Total Cards
Yellow Cards, Red Cards → Total Cards
Fouls_Player_id→ Player_id
Fouls_Player_id→ Yellow Cards
Fouls_Player_id→ Red Cards
Fouls_Player_id→ Total Cards
Fouls_Player_id, Red Cards → Total Cards
Yellow Cards, Fouls_Player_id → Total Cards}

7. Points_Home (Points_Home_id, Club_id, Points_won, Points_dropped, Percentage)

Keys→Points_Home_id, Points_won, Points_dropped
F={Points_Home_id→Club_id, Points_won, Points_dropped, Percentage
Points_won, Points_dropped → Percentage}

F^+ = { Points_Home_id→Club_id, Points_won, Points_dropped, Percentage
Points_won, Points_dropped → Percentage
Points_Home_id→Club_id
Points_Home_id→Points_won
Points_Home_id→Points_dropped
Points_Home_id→ Percentage
Points_Home_id, Points_dropped → Percentage
Points_won, Points_Home_id → Percentage}

8. Points_Away (Points_Away_id, Club_id, Points_won, Points_dropped, Percentage)

Keys→Points_Away_id , Points_won, Points_dropped

F={Points_Away_id→Club_id, Points_won, Points_dropped, Percentage
Points_won, Points_dropped → Percentage}

F^+ = { Points_Away_id→Club_id, Points_won, Points_dropped, Percentage
Points_won, Points_dropped → Percentage
Points_Away_id→Club_id
Points_Away_id→Points_won
Points_Away_id→Points_dropped

Points_Away_id→ Percentage
Points_Away_id, Points_dropped → Percentage
Points_won, Points_Away_id → Percentage}

9. Passing_Team (Passing_Team_id, Club_id, Pass_completion, Pass_accuracy)

Keys→Passing_Team_id, Pass_completion

F={Passing_Team_id→Club_id, Pass_completion, Pass_accuracy}

F^+ = {Passing_Team_id→Club_id, Pass_completion, Pass_accuracy
Passing_Team_id→Club_id
Passing_Team_id→Pass_completion
Passing_Team_id→Pass_accuracy}

10. Passing_Player (Passing_Player_id,Player_id, Pass_completion, Pass_accuracy)

Keys→Passing_Player_id, Pass_completion

F={Passing_Player_id→Player_id, Pass_completion, Pass_accuracy

F^+ = {Passing_Player_id→Player_id, Pass_completion, Pass_accuracy
Passing_Player_id→Player_id
Passing_Player_id→Pass_completion
Passing_Player_id→Pass_accuracy}

11. Set_Pieces_Team (Set_Pieces_id, Club_id, Free_kicks, Penalties, Corners, Percentage)

Keys→Set_Pieces_id
F={Set_Pieces_id→ Club_id, Free_kicks, Penalties, Corners, Percentage}
F={Set_Pieces_id→ Club_id, Free_kicks, Penalties, Corners, Percentage}

12. Shots_Team (Shots_id, Club_id, Total_shots, Shots_on_target, Shot_accuracy)
Keys→Shots_id, Total_shots, Shots_on_target

F={Shots_id→Club_id, Total_shots, Shots_on_target, Shot_accuracy
Total_shots, Shots_on_target→Shot_accuracy}

F^+ = {Shots_id→Club_id, Total_shots, Shots_on_target, Shot_accuracy
Total_shots, Shots_on_target→Shot_accuracy
Shots_id→Club_id
Shots_id→Total_shots
Shots_id→Shots_on_target

Shots_id→Shot_accuracy
Shots_id, Shots_on_target→Shot_accuracy
Total_shots, Shots_id→Shot_accuracy}

13. Revenues (Revenues_id, Club_id, Match_day, Broadcast, Commercial, Total)

Keys→Revenues_id, Match_day, Broadcast, Commercial

F={Revenues_id→Club_id, Match_day, Broadcast, Commercial, Total
Match_day, Broadcast, Commercial→Total}

F^+={Revenues_id→Club_id, Match_day, Broadcast, Commercial, Total
Match_day, Broadcast, Commercial→Total
Revenues_id→Club_id
Revenues_id→Match_day
Revenues_id→ Broadcast
Revenues_id→Commercial
Revenues_id→Total
Revenues_id, Broadcast, Commercial→Total
Match_day, Revenues_id, Commercial→Total
Match_day, Broadcast, Revenues_id→Total}

14. Standings (Standings_id, Club_id, Points, Standings)

Keys→Standings_id, Points

F={Standings_id→Club_id, Points, Standings
Points→Standings}

F^+={Standings_id→Club_id, Points, Standings
Points→Standings
Standings_id→Club_id
Standings_id→Points
Standings_id→Standings}

Normalization

1NF

Since the attribute ‘position’ in player table is multi valued, we decompose the relation Player into

Player (Player_id, Club_id, Player_name, Jersey_no)
Position (Player_id, Position)

All other attributes are single valued. Hence the database is reduced to 1NF form.

2NF

1. Team (Club_id, Club_name, Stadium, Manager)

   Keys→ Club_id, Club_name

   F={Club_id→ Stadium, Manager, Club_name
   Club_name→ Stadium, Manager}

   There exists partial key dependency since the non key attributes Stadium, Manager depend on partial key Club_name and partial key Club_id.
   So we decompose relation Team into

   Team (Club_id, Club_name, Stadium, Manager)
   Team_info (Club_name, Stadium, Manager)

2. Player (Player_id, Club_id, Player_name, Jersey_no)

   Keys→ Player_id,Player_name

   F={Player_id→ Player_name, Jersey_no, Club_id
   Player_name→ Jersey_no, Club_id}

   Similar to above, we decompose Player into

   Player (Player_id, Club_id, Player_name, Jersey_no)
   Player_info (Player_name , Club_id, Jersey_no)

Proceeding in similar way for the remaining relations, we obtain the following schemas after decomposition into 2NF form:

1. Team (Club_id, Club_name, Stadium, Manager)
2. Team_info (Club_name, Stadium, Manager)

3. Player (Player_id, Club_id, Player_name, Jersey_no)
4. Position (Player_id, Position)
5. Player_info (Player_name , Club_id, Jersey_no)
6. Goals_Team (Goals_Team_id, Club_id, Goals_for, Goals_conceded, Goal_difference, Home_Goals, Away_Goals)
7. Goals (Goals_for, Goals_conceded, Goal_difference)
8. Attacking_Output_Player (Attacking_Player_id, Player_id, Goals, Assists, Attacking_Output)
9. Attacking_Output (Goals, Assists, Attacking_Output)
10. Fouls_Team (Fouls_Team_id, Club_id, Yellow Cards, Red Cards, Total Cards)
11. Fouls_Player (Fouls_Player_id, Player_id, Yellow Cards, Red Cards, Total Cards)
12. Fouls (Yellow Cards, Red Cards, Total Cards)
13. Points_Home (Points_Home_id, Club_id, Points_won, Points_dropped, Percentage)
14. Points_Away (Points_Away_id, Club_id, Points_won, Points_dropped, Percentage)
15. Points (Points_won, Points_dropped, Percentage)
16. Passing_Team (Passing_Team_id, Club_id, Pass_completion, Pass_accuracy)
17. Passing_Player (Passing_Player_id,Player_id, Pass_completion, Pass_accuracy)
18. Set_Pieces_Team (Set_Pieces_id, Club_id, Free_kicks, Penalties, Corners, Percentage)
19. Shots_Team (Shots_id, Club_id, Total_shots, Shots_on_target, Shot_accuracy)
20. Shots_Accuracy (Total_shots, Shots_on_target, Shot_accuracy)
21. Revenues (Revenues_id, Club_id, Match_day, Broadcast, Commercial, Total)
22. Total_Revenues (Match_day, Broadcast, Commercial, Total)
23. Standings (Standings_id, Club_id, Points, Standings)
24. StandingsPoints (Points, Standings)

3NF

There shouldn’t be any interdependencies among the non-key attributes i.e. transitive dependencies in 3NF form.

1. Team (Club_id, Club_name, Stadium, Manager)
Keys→ Club_id
Club_id→ Club_name, Stadium, Manager
Club_name→ Stadium, Manager

So there is transitive dependencies among Club_name, Stadium and Manager. So we decompose Team into

Team (Club_id, Club_name)
Team_info (Club_name, Stadium, Manager)

Team_info already exists with the same attributes. So only the attributes of Team are reduced.

Proceeding in similar way,we obtain the following schemas after decomposition into 3NF form:

1. Team (Club_id, Club_name)
2. Team_info (Club_name, Stadium, Manager)
3. Player (Player_id, Player_name)
4. Position (Player_id, Position)
5. Player_info (Player_name , Club_id, Jersey_no)
6. Goals_Team (Goals_Team_id, Club_id, Goals_for, Goals_conceded, Home_Goals, Away_Goals)
7. Goals (Goals_for, Goals_conceded, Goal_difference)
8. Attacking_Output_Player (Attacking_Player_id, Player_id, Goals, Assists)
9. Attacking_Output (Goals, Assists, Attacking_Output)
10. Fouls_Team (Fouls_Team_id, Club_id, Yellow Cards, Red Cards)
11. Fouls_Player (Fouls_Player_id, Player_id, Yellow Cards, Red Cards)
12. Fouls (Yellow Cards, Red Cards, Total Cards)
13. Points_Home (Points_Home_id, Club_id, Points_won, Points_dropped)
14. Points_Away (Points_Away_id, Club_id, Points_won, Points_dropped)
15. Points (Points_won, Points_dropped, Percentage)
16. Passing_Team (Passing_Team_id, Club_id, Pass_completion, Pass_accuracy)
17. Passing_Player (Passing_Player_id,Player_id, Pass_completion, Pass_accuracy)
18. Set_Pieces_Team (Set_Pieces_id, Club_id, Free_kicks, Penalties, Corners, Percentage)
19. Shots_Team (Shots_id, Club_id, Total_shots, Shots_on_target)
20. Shots_Accuracy (Total_shots, Shots_on_target, Shot_accuracy)
21. Revenues (Revenues_id, Club_id, Match_day, Broadcast, Commercial)
22. Total_Revenues (Match_day, Broadcast, Commercial, Total)
23. Standings (Standings_id, Club_id, Points, Standings)
24. StandingsPoints (Points, Standings)

BCNF form:

In the above schemas, all the determinants are candidate keys which uniquely identifies each record. So the above schema is already in BCNF form.
Example:

Team_info (Club_name, Stadium, Manager)
Keys→ Club_name
Club_name→ Stadium, Manager

determinants→ Club_name
Here Club_name is candidate key. So it is in BCNF form.
Similarly, all other schemas are in BCNF form already.
Hence, our final normalized schema is:

1. Team (Club_id, Club_name)
2. Team_info (Club_name, Stadium, Manager)
3. Player (Player_id, Player_name)
4. Position (Player_id, Position)
5. Player_info (Player_name, Club_id, Jersey_no)
6. Goals_Team (Goals_Team_id, Club_id, Goals_for, Goals_conceded, Home_Goals, Away_Goals)
7. Goals (Goals_for, Goals_conceded, Goal_difference)
8. Attacking_Output_Player (Attacking_Player_id, Player_id, Goals, Assists)
9. Attacking_Output (Goals, Assists, Attacking_Output)
10. Fouls_Team (Fouls_Team_id, Club_id, Yellow Cards, Red Cards)
11. Fouls_Player (Fouls_Player_id, Player_id, Yellow Cards, Red Cards)
12. Fouls (Yellow Cards, Red Cards, Total Cards)
13. Points_Home (Points_Home_id, Club_id, Points_won, Points_dropped)
14. Points_Away (Points_Away_id, Club_id, Points_won, Points_dropped)
15. Points (Points_won, Points_dropped, Percentage)
16. Passing_Team (Passing_Team_id, Club_id, Pass_completion, Pass_accuracy)
17. Passing_Player (Passing_Player_id,Player_id, Pass_completion, Pass_accuracy)
18. Set_Pieces_Team (Set_Pieces_id, Club_id, Free_kicks, Penalties, Corners, Percentage)
19. Shots_Team (Shots_id, Club_id, Total_shots, Shots_on_target)
20. Shots_Accuracy (Total_shots, Shots_on_target, Shot_accuracy)
21. Revenues (Revenues_id, Club_id, Match_day, Broadcast, Commercial)
22. Total_Revenues (Match_day, Broadcast, Commercial, Total)
23. Standings (Standings_id, Club_id, Points, Standings)
24. StandingsPoints (Points, Standings)

Primary Keys

The underlined attributes in each entity are the primary keys of the corresponding entities because it uniquely identifies each entity.

1. Team (Club_id, Club_name)
2. Team_info (Club_name, Stadium, Manager)
3. Player (Player_id, Player_name)
4. Position (Player_id, Position)
5. Player_info (Player_name , Club_id, Jersey_no)
6. Goals_Team (Goals_Team_id, Club_id, Goals_id)
7. Goals (Goals_id, Goals_for, Goals_conceded, Goal_difference,Home_Goals, Away_Goals)
8. Attacking_Output_Player (Attacking_Player_id, Attack_id, Player_id)
9. Attacking_Output (Attack_id, Goals, Assists, Attacking_Output)
10. Fouls_Team (Fouls_Team_id, Club_id, Fouls_id)
11. Fouls_Player (Fouls_Player_id, Player_id, Fouls_id)
12. Fouls (Fouls_id, Yellow Cards, Red Cards, Total Cards)
13. Points_Home (Points_Home_id, Club_id, Points_id)
14. Points_Away (Points_Away_id, Club_id, Points_id)
15. Points (Points_id, Points_won, Points_dropped, Percentage)
16. Passing_Team (Passing_Team_id, Club_id, Pass_completion, Pass_accuracy)
17. Passing_Player (Passing_Player_id,Player_id, Pass_completion, Pass_accuracy)
18. Set_Pieces_Team (Set_Pieces_id, Club_id, Free_kicks, Penalties, Corners, Percentage)
19. Shots_Team (Shots_id, Club_id,Accuracy_id)
20. Shots_Accuracy (Accuracy_id, Total_shots, Shots_on_target, Shot_accuracy)
21. Revenues (Revenues_id, Club_id, Total_id)
22. Total_Revenues (Total_id, Match_day, Broadcast, Commercial, Total)
23. Standings (Standings_id, Club_id, Standings)
24. StandingsPoints (Points, Standings)

Some composite primary keys in different relations are:

1. Club_id

The primary key Club_id is generated using the first three letters of the Club_name followed by 2 digit number which is the serial number corresponding to the particular club after sorting the records in ascending alphabetical order by Club_name.

Example: If the Club_name is **Ars**enal which is the **2nd** record as per its alphabetical order, then it would have the Club_id '**ARS02'**.

2. Player_id

The primary key Player_id is generated using the first letters of the first name and last name followed by first three letters of the Club_name. Then it is followed by a 2 digit number which is the jersey number of the corresponding player.

Example: If the Player_info is **A**lexis **S**anchez and Club_name is **Ars**enal i.e. the player plays for Arsenal and wears jersey no **07**, then it would have the Player_id ‘**ASARS07**’.

Foreign Keys

The dotted line attributes are the foreign keys used to link the parent and child tables. Some of the foreign keys are:

1. Club_id is a foreign key in table ‘Player_info’ and is a primary key in table ‘Team’. So Player_info is child table for parent table ‘Team’ .

2. Player_info is a foreign key in table ‘Player’ and primary key in table ‘Player_info’. So ‘Player’ is child table for parent table ‘Player_info.’

Attributes (Foreign Key)	**Child tables**	**Parent tables**
Club_name	Team	Team_info
Player_name	Player	Player_info
Player_id	Player, Fouls_Player, Passing_Player, Attacking_Output_Player	Player
Club_id	Player_info, Fouls_team, Points_Home, Points_Away, Passing_Team, Set_Pieces_Team, Shots_Team, Revenues, Standings	Team
Goals_id	Goals_Team	Goals
Attack_id	Attacking_Output_Player	Attacking_Output
Fouls_id	Fouls_Player, Fouls_Team	Fouls
Points_id	Points_Home, Points_Away	Points
Accuracy_id	Shots_team	Shots_Accuracy
Total_id	Revenues	Total_Revenues
Standings	Standings	StandingsPoints

Referential Integrity

We used referential integrity constraints primary key and foreign key.

Examples in our database:

```
ALTER TABLE `Goals_Team` ADD CONSTRAINT `Goals_id`
FOREIGN KEY (`Goals_id`) REFERENCES `Goals`(`Goals_id`)

ALTER TABLE `Fouls_Player` ADD CONSTRAINT `Fouls_id`
FOREIGN KEY (`Fouls_id`) REFERENCES `Fouls`(`Fouls_id`)

ALTER TABLE `Team` ADD CONSTRAINT `Club_id`
PRIMARY KEY (`Club_id`)

ALTER TABLE `Player` ADD CONSTRAINT `Player_id`
PRIMARY KEY (`Player_id`)
```

Cascading actions in foreign key constraint

We used on delete cascade and on update cascade in all our foreign key constraints.

Examples in our database:

```
ALTER TABLE `team` ADD CONSTRAINT `Team_teamname`
FOREIGN KEY (`Club_name`) REFERENCES `team_info`(`Club_name`)
ON DELETE CASCADE
ON UPDATE CASCADE;

ALTER TABLE `player_info` ADD CONSTRAINT `Playername_team`
FOREIGN KEY (`Club_id`) REFERENCES `team`(`Club_id`)
ON DELETE CASCADE
ON UPDATE CASCADE;
```

Check Constraints

The check constraints used in our database are:

```
alter table points add CONSTRAINT Percentage
check (Percentage<100);

alter table standingspoints add CONSTRAINT standings
check (standings BETWEEN 1 and 10);
```

However, MySQL does not support SQL **check constraints**. We can define them in your DDL query for compatibility reasons but they are just ignored.

Weak Entity:

In our database, the entity **Position** is a weak entity set since it doesn't have any primary key but has Player_id as a discriminating key. The identifying entity seat for this weak entity set is entity **Player**. There is many to one relationship from the weak entity set Position to identifying entity set Player since one player can be associated to many positions and Position has total participation since each position must be associated to a player.

Indexing:

In our database, we have implemented indexing on various attributes for faster operations especially for searching of records using the SQL queries:

```
create index player_name on player_info(player_name);
create index Club_name on Team(Club_name);
```

We created an index on each of the foreign keys as well as on Club_name of the table Team and on Player_name of the table Player because we search by name of players and teams in our database.

Join

We used natural join and inner join because each record in a table had an association to the table to be joined. So the multiple tables had only the common records.

Examples used in our database:
Natural join:

1. Team (Club_id, Club_name)
2. Goals_Team (Goals_Team_id, Club_id, Goals_id)
3. Goals (Goals_id, Goals_for, Goals_conceded, Goal_difference,Home_Goals, Away_Goals)

```
SELECT Club_name,Goals_for,Goals_conceeded,Goal_difference,Home_Goals,Away_Goals
FROM team NATURAL join goals_team
NATURAL join goals
order by Goal_difference desc;
```

1. Points_Away (Points_Away_id, Club_id, Points_id)
2. Points (Points_id, Points_won, Points_dropped, Percentage)
3. Team (Club_id, Club_name)

```
SELECT Club_name,Points_won,Points_dropped,Percentage
FROM points natural join points_away natural join team
order by Percentage desc;
```

Inner join:

1. Player (Player_id, Player_name)
2. Attacking_Output_Player (Attacking_Player_id, Attack_id, Player_id)
3. Attacking_Output (Attack_id, Goals, Assists, Attacking_Output)

```
Select
player.Player_name,attacking_output.Goals,attacking_output.Assists,attacking_output.Attacking_Output
 FROM attacking_output inner join attacking_output_player
 ON attacking_output.Attack_id=attacking_output_player.Attack_id
 inner join player
on attacking_output_player.Player_id=player.Player_id
order by Attacking_Output desc;
```

1. Standings (Standings_id, Club_id, Standings)
2. StandingsPoints (Points, Standings)
3. Team (Club_id, Club_name)

```
SELECT team.Club_Name,standingspoints.Points,standingspoints.Standings
FROM standings inner join standingspoints
ON standings.Standings=standingspoints.Standings
inner join team
on standings.Club_id=team.Club_id
order by standings;
```