

Firas Hachem  
Karim Kenaan

Mohamad AlKadri

Mohamad kazem abdel-sater

Spring 2016-2017

CMPS 277 database systems

American University of Beirut

Department of Computer Science

Spring 2017

Table of Contents

[1- Project Description 2](#_Toc449726973)

[A- Entities: 2](#_Toc449726974)

[B- Relationships: 2](#_Toc449726975)

[C- Constraints: 2](#_Toc449726976)

[D- Sample Queries: 2](#_Toc449726977)

[2- EER MODEL 3](#_Toc449726978)

[3- Relational Model 4](#_Toc449726979)

[4- Views and indexes 4](#_Toc449726980)

[a- Views: 4](#_Toc449726981)

[b- Indexes: 5](#_Toc449726982)

[5- Interface 6](#_Toc449726983)

[6- Conclusion 12](#_Toc449726984)

# Project Description

This project aims at creating a database to manage a university’s athletic teams

1. **Entities:**

We will maintain information about the following set of entities:

People:

Player: ID, Name, Age, Team Number, Team Position, Team Role (Captain, substitute, reserve,).

Staff: ID, Name, Age, Role (Doctor, Assistant-manager, Coaching, Management), Team.

Club: Name (is a candidate key), Buildings/Locations/Stadiums.

Team: Name, Awards, ID, Type (Soccer, Basketball, Handball etc.)

Tournament: Name (is candidate key), Type (League, Cup etc.), Date started, Data finished, Prize, #of rounds, Sponsors.

Court: Type (Football, Basketball, etc.), Name (is candidate key), Location, Condition (Indoor or Outdoor), timing (opening/Closing), Size.

Record:

Player Record: Number of assists, Number of goals, Number of matches played, years of experience, Date (joined), Height, weight, Achievement/Awards

Team Record: Number of assists, Number of goals, Number of matches played, Achievement/Awards, and Number of wins/loses

## Relationships:

        The following relationships will hold between our entities:

1. Teams will have Players, they will also have Coaching Staff and Management,
2. Clubs will have Teams belonging to them, they will have People associated with them as Members or Staff.
3. Players are Members in a Club and play on a Team in that Club
4. Teams will participate in Tournaments
5. Teams will train/play in courts.
6. Tournaments will take place in courts
7. Team will have team record
8. Player will have player record

## Constraints:

1. A Person cannot belong to more than one Club.
2. A Club can have Several Teams but a Team can only belong to one Club.
3. No two Teams can have the same court.
4. Each club must own court.
5. No team can participate in a tournament without the minimum number of players.
6. No two team can win the same award for the same year.
7. A club can have several staffs but a staff can only belong to one club.
8. A team can have only one team record
9. A player can have only one player record

## Sample Queries:

1.    List all Players belong to a specific team

2.    List all Players according to their (achievements, weight, height, age, or team position)

3.    List all team that won a particular award

4.    List all teams that belong to a specific club.

5.  List all tournaments according to the value prize

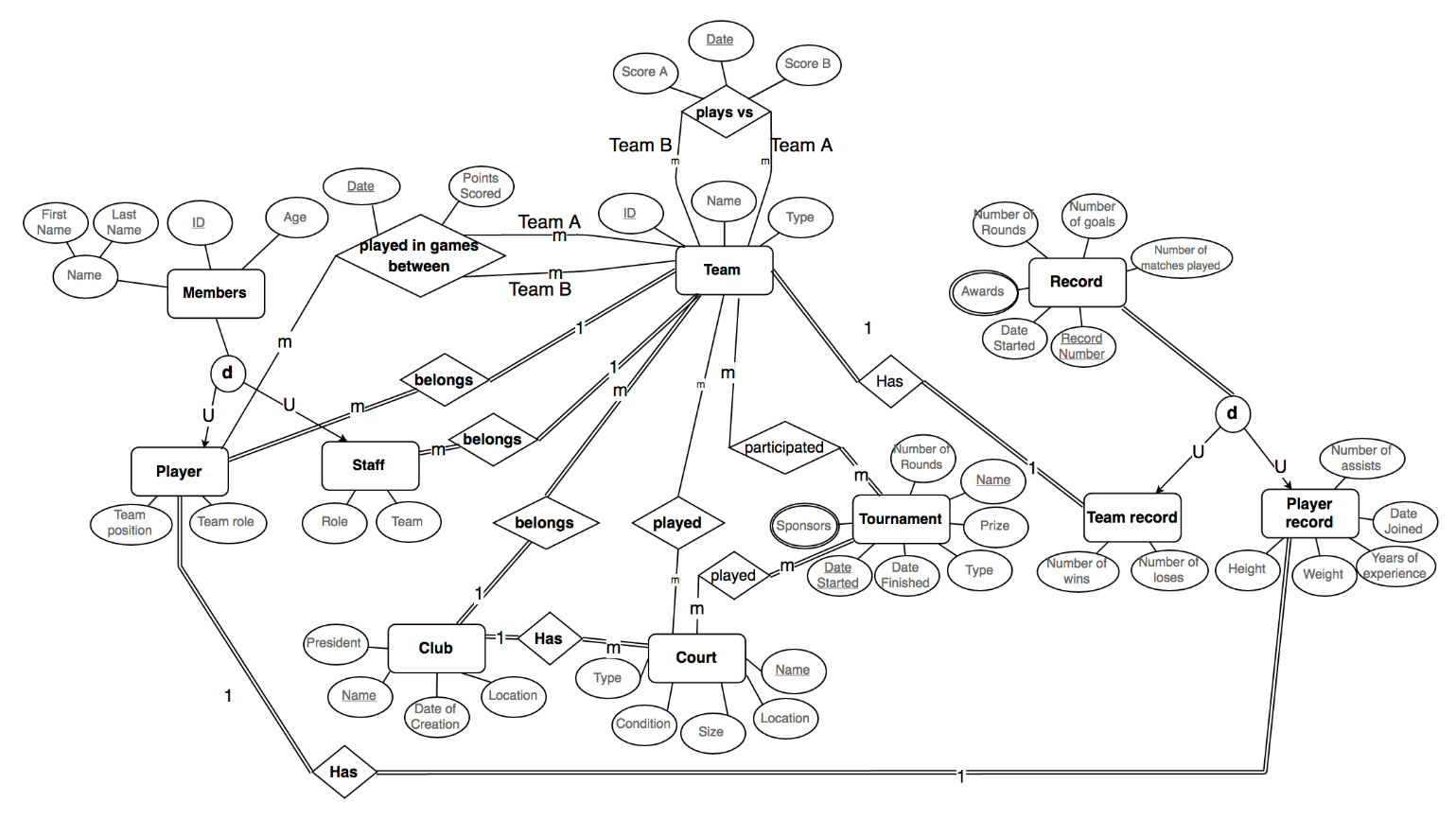
6.  List all tournaments the have been participated by a specific team

7. List all staffs belong to a specific team.

9. List all courts that are owned by a specific team

10. List all players according to a specific criteria (Height, weight and so on from the record)

# EER MODEL



# Relational Model

## C:\Users\kak17\Desktop\Relational Model.png

# Views and indexes

**Views**:

We have created Six views in our database

1. Staffinfo : This view does join between: staff , team , and members.
2. Clubteams : This view does join between: club , team , and player.
3. Clubcourts : This view does join between: club, and court.
4. Playerinfo : This view does join between: player , members , team , playerRecord , and records.
5. Playerawards : This view does join between: records , playerRecord , awards , player , and members.
6. PlayerMatches: This view does join between: team, team, player, members, and playersMatchHistory

The reason why we have created these views is that many queries in our interface tend to use these views; therefore, instead of repeating calling on the complete query statement, we just call on these views (Virtual tables).

**Indexes :**

An index is used in order to speed up the process of searching, and the performance of the queries. It is done by reducing of number of blocks scanned. It is used mainly for the columns that are most frequently used, searched. In our case, we did indexing for all of the primary and foreign keys, because these are the most queried columns, especially in joining. By default MYSQL creates indexes on primary and foreign keys, which is so recommended. In addition to the indexes of primary and foreign keys we added the following secondary indexes:

1.

CREATE INDEX teamName  
ON team (teamName);

This index is so essential because most of our quires search for a team according to its name. For example find the team who has the given name “AUB varsity Team”.

2.

CREATE INDEX name  
ON members (firstName,lastName);

This index is so essential because most of our quires search for a member according to its first and last name. For example find a member who has the given firstName= “Mohamad” and lastName=”Alkadri”.

3.

CREATE INDEX sportName  
ON team (sportName);

This index is required because some quires search for teams according to their sport Name. For example find all teams who practiced “soccer”.

4.

CREATE INDEX location  
ON club (location);

This index is required because some quires search for clubs according to their Location. For example find all clubs who are located in “Beirut”.

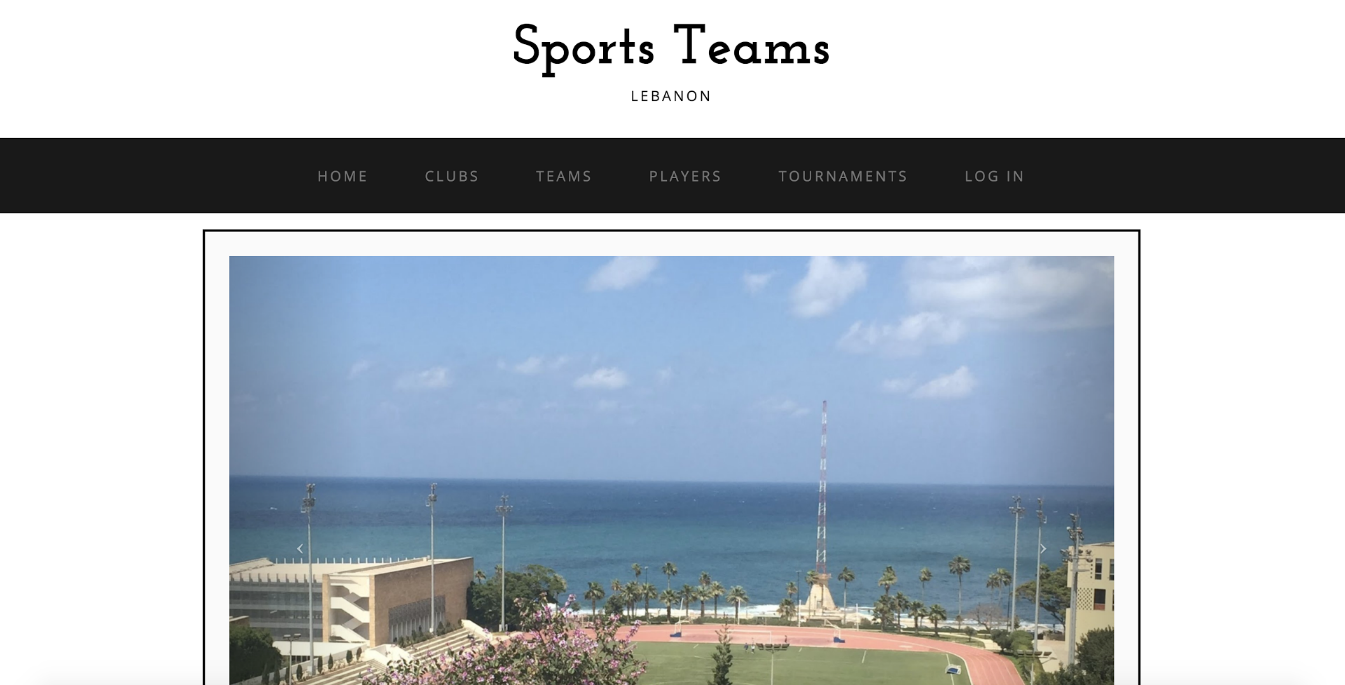
5.

CREATE INDEX teamPosition  
ON player (teamPosition);

This index is required because some quires search for player according to its team position. For example find all players who played as “attackers”.

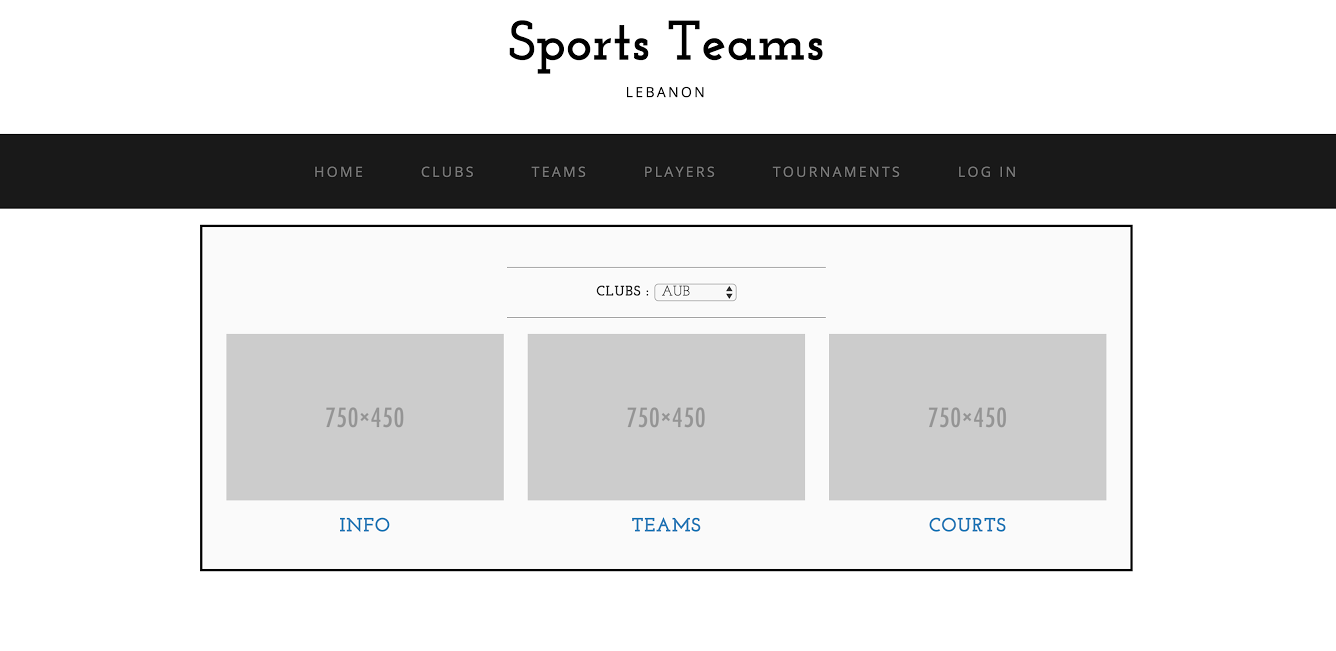
# Interface

Since we are implementing the database of sports teams, we have created an interface that starts with a welcome screen. If you scroll down, you can see some general information about the website, and the Lebanese Universities Sports teams. In addition in the navigation bar you can navigate to different screens which are Clubs, Teams, Players, and Registration.



As a user or a guest you can only see the info about the teams, clubs, and players.

In Clubs screen you can see a table of records of the Club (query 1) and teams in the club (query 2) and courts that the team plays in (query 3).

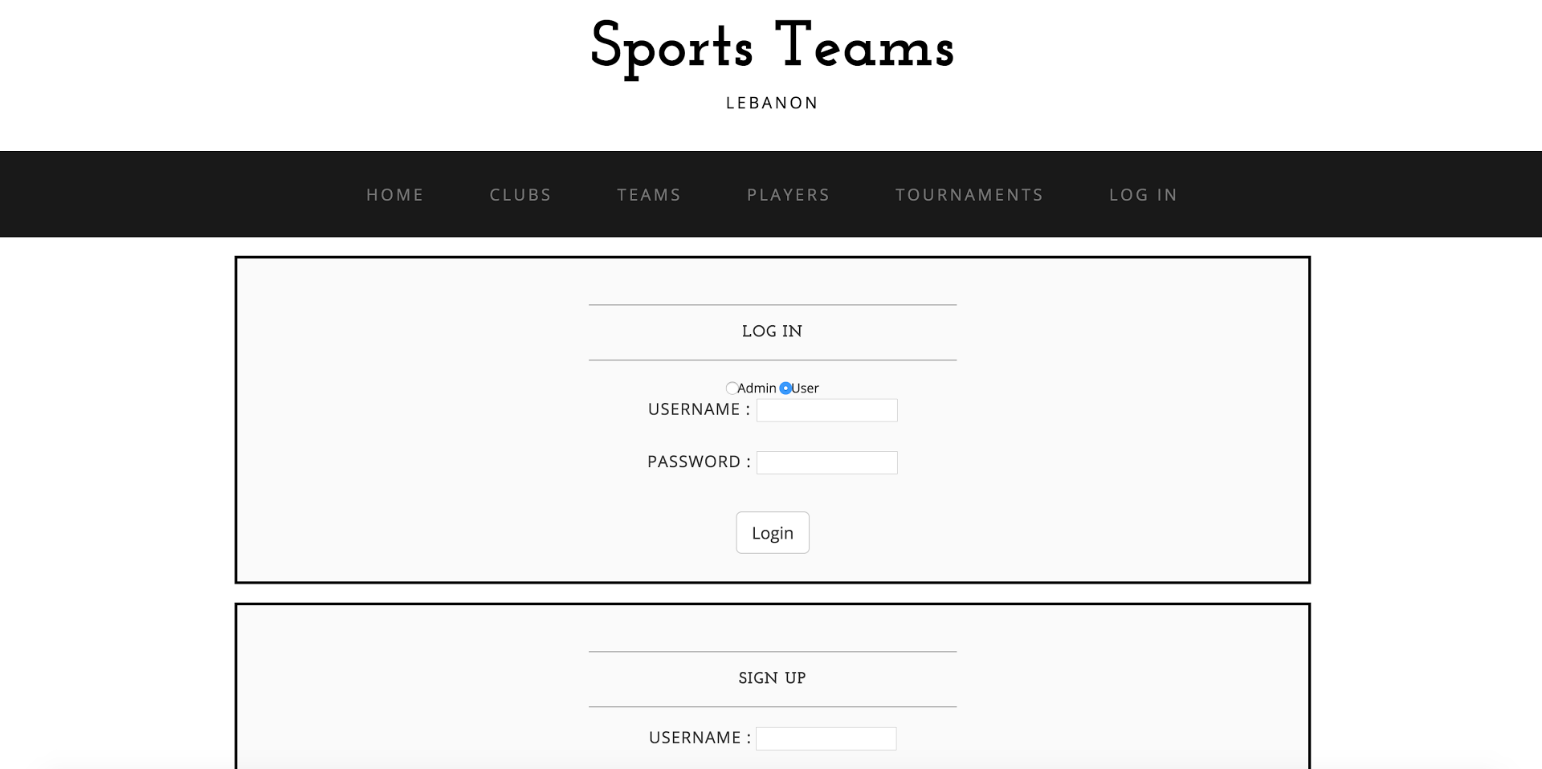


In the Teams Screen, the user can see table of players in the team (query 4), table of records of the team (query5), table of the games’ records played by the team (query 6), table of staff that works with the team (query 7), and a table of tournaments played in by the team (query 8).

In the Players screen, the user can see a table of records of each player (query 9), and a table of awards won by the player (query 10)

In Registration you can log in as a user or as an admin, where the user can only view data, while the admin can add, delete and update data. You can also sign up to the website.

As an admin, you can add a club, team, player, staff, records, awards, and tournament.



Once you press, you will be redirected to a new page, where you can first add, delete and update the information of a staff. You should add the requested information and accordingly the database will be updated.

1. **Conclusion**

As a conclusion*, we can see how important implementing a database is. This is useful for all projects both in the university and in the work place, so it obviously became necessary for all backups in all companies. In addition, we have noticed the significance of using views for easier deployment of table structure changes as well as indexes for quicker search processes. Overall, the experience was very beneficial for all of us, specifically, for our future educational.*