

Final Project Report

Name: Jakub Szumny

NetID: jszum3

Purpose and Scope

The purpose and scope of my database project is to create a database containing soccer information. This includes the players, their leagues, their teams, their statistics, their coaches, and the matches they have played.

Business Objective

I have created this database with the business objective of providing more statistics for soccer fans in major leagues, and to provide sports betters with more statistics to make more informed decisions on their bets.

Stakeholder Analysis

There are many roles in this database, and how these roles can use the information in this database. Soccer players have all their information and statistics in the database, so they can see how well they perform. Coaches can also keep track of their players' statistics to ensure they play to a certain standard and track how well their team is performing. Soccer fans can use this database to keep up with all current events and statistics of players and teams in major soccer leagues. Finally, sports betters can use the information to make informed decisions on bets.

User Requirements

The users of this database have specific requirements and/or preferences of what they can do with this database. Soccer fans expect a user-friendly process that allows them to learn more about their favorite players and teams. Coaches expect an informational database that will allow them to keep track of their team's and players' performances and progress. Players expect an informational database that will allow them to keep track of their performances and progress. Sports betters expect an informational and heavily statistical database with they can use to make better, more informed decisions on future bets.

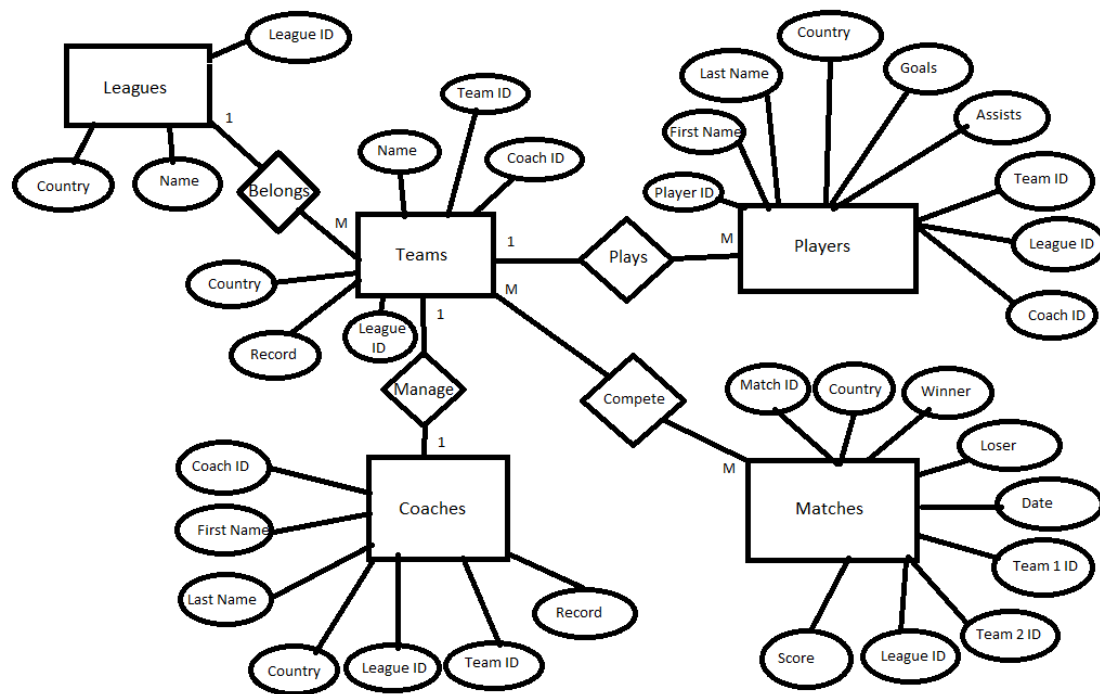
Functional Scope

The functional scope of this database system allows for all users to have a great place to search, make decisions, and see information about current players, teams, coaches, matches, and leagues.

Data Scope

The types of data that will be managed by the system are player data which will capture and store information about players and their statistics/information, league data which will capture and store information about leagues and their teams, team data which will capture and store information about specific teams and their games played, coach data that will capture and store information about coaches and their performances as coaches, and match data that will capture and store information about matches.

ERD for the Database



I made some changes to the ERD, by adding attributes to each table, and also added a completely new table, Matches, for more information in the entire Database. Also fixed how each table was related to the other tables.

Database Schema Based on ERD

Part 1: Entity-Relationship Diagram (ERD) to Database Schema Translation

Tables Defined in SQL:

Leagues - Stores League Information

- leagueID (primary key)
- Name
- Country

Teams - Stores Team Information and Statistics

- teamID (primary key)
- Name
- Country
- Record
- leagueID (foreign key)
- coachID (foreign key)

Coaches - Stores Coach Information and Statistics

- coachID (primary key)
- firstName
- lastName
- Country
- record
- teamID (foreign key)
- leagueID (foreign key)

Players - Stores Player Information and Statistics

- playerID (primary key)
- firstName
- lastName
- Country
- Goals
- Assists
- teamID (foreign key)
- leagueID (foreign key)
- coachID (foreign key)

Matches - Stores information on specific matchups

- matchID (primary key)
- Country
- Score
- Winner
- Loser
- Date
- team1ID (foreign key)
- team2ID (foreign key)
- leagueID (foreign key)

Part 2: Explanation of Key Choices

Primary Keys: Used for unique identification among the parts of the soccer world. LeagueID, TeamID, CoachID, PlayerID, and MatchID serve as the primary keys in their respective tables.

Foreign Keys: Establish Relationships among the tables. For example, teamID in players references teamID in teams, linking players to their teams. Also, coachID in teams references coachID in coaches, linking coaches to their teams.

Part 3: SQL Code and Table Creation

```
1 CREATE TABLE Leagues (  
2     leagueID int,  
3     name varchar(255),  
4     country varchar(255),  
5     PRIMARY KEY (leagueID)  
6 )
```

```
1 CREATE TABLE Coaches (  
2     coachID int,  
3     firstName varchar(255),  
4     lastName varchar(255),  
5     teamID int,  
6     leagueID int,  
7     country varchar(255),  
8     record varchar(255),  
9  
10    PRIMARY KEY (coachID),  
11    FOREIGN KEY (teamID) REFERENCES teams(teamID)  
12    FOREIGN KEY (leagueID) REFERENCES Leagues(leagueID)  
13 )
```

```
1 CREATE TABLE teams (  
2     teamID int,  
3     name varchar(255),  
4     country varchar(255),  
5     leagueID int,  
6     coachID int, record TEXT,  
7     PRIMARY KEY (teamID),  
8     FOREIGN KEY (leagueID) REFERENCES Leagues(leagueid),  
9     FOREIGN KEY (coachID) REFERENCES Coaches(coachID)  
10 )
```

```
1 CREATE TABLE players (  
2     playerID int,  
3     firstName varchar(255),  
4     lastName varchar(255),  
5     country varchar(255),  
6     teamID int,  
7     goals int,  
8     assists int,  
9     leagueID int,  
10    coachID int,  
11    PRIMARY KEY (playerID),  
12    FOREIGN KEY (leagueID) REFERENCES leagues(leagueid),  
13    FOREIGN KEY (coachID) REFERENCES coaches(coachID),  
14    FOREIGN KEY (teamID) REFERENCES teams(teamID)  
15 )
```

```
1 CREATE TABLE matches (  
2     matchID int,  
3     team1ID varchar(255),  
4     team2ID varchar(255),  
5     leagueID int,  
6     country varchar(255),  
7     winner varchar(255),  
8     loser varchar(255), date NUMERIC, score TEXT,  
9     PRIMARY KEY (matchID),  
10    FOREIGN KEY (team1ID) REFERENCES teams(teamID),  
11    FOREIGN KEY (team2ID) REFERENCES teams(teamID),  
12    FOREIGN KEY (leagueID) REFERENCES Leagues(leaguesID)  
13 )
```

SQL Queries to Demonstrate Functionality

Query 1

```
1 SELECT firstName, lastName, leagues.Name, max(goals), assists FROM players
2 JOIN Leagues
3 WHERE leagues.leagueID = players.leagueid
4 AND leagues.Name = "Premier League"
```

firstName	lastName	name	max(goals)	assists
Erling	Haaland	Premier League	21	5

This query selects the player who has the most amount of goals in the Premier League. It joins two different tables, Players and Leagues, and grabs the full name, the league name, and the statistics of the player who scored the most goals. It is relevant to the schema as it allows a user to find the best-performing players in any specific league or team.

Query 2

```
1 SELECT matchID, country, winner, loser, date, score FROM matches
2 WHERE score > "3-0"
3
```

matchID	country	winner	loser	date	score
200	Germany	Bayern Munich	Dortmund	11-04-23	4-0
600	Spain	Real Madrid	Dortmund	12-6-17	3-2

This query selects the match data of games where the score was greater than “3-0”. It grabs the matchID, the country, the winner, the loser, the date, and the score of the games where the score was greater than “3-0”. It is relevant to the schema as it allows a user to find specific games based on information about the game such as the score.

Query 3

```
1 SELECT players.firstName, players.lastname, teams.name, leagues.name, players.goals FROM players
2 JOIN teams
3 JOIN Leagues
4 WHERE players.teamID = teams.teamID
5 AND teams.leagueID = leagues.leagueID
```

firstName	lastName	name	name	goals
Erling	Haaland	Manchester City	Premier League	21
Harry	Kane	Bayern Munich	Bundesliga	22
Robert	Lewandowski	Barcelona	La Liga Esports	9
Jude	Bellingham	Real Madrid	La Liga Esports	16
Scott	McTominay	Manchester United	Premier League	6
Marco	Reus	Dortmund	Bundesliga	4

This query selects all players and the amount of goals they have scored for their team. It joins three tables, Players, Teams, and Leagues, and grabs important information from each. It is relevant to the schema because it allows a user to compare players from different teams and leagues based on their performances. It also allows a user to see what team players play for, or for what league their teams are in.

Query 4

```
1 SELECT teams.Name, teams.record, coaches.firstName, coaches.lastname FROM teams
2 JOIN coaches
3 WHERE teams.coachID = coaches.coachID
4
```

name	record	firstName	lastName
Manchester City	16-5-4	Pep	Guardiola
Bayern Munich	10-2-1	Thomas	Tuchel
Barcelona	14-4-4	Xavi	Creus
Real Madrid	12-3-1	Carlos	Ancelotti
Manchester United	11-1-12	Erik	Ten Hag
Dortmund	7-4-3	Edin	Terzic

This query selects the current-season records of teams under specific coaches. It joins the tables Teams and Coaches and grabs the team name, their record, and the coach's full name. It is relevant to the schema because it allows a user to see what a team's current record is, and who coaches that team.