**Introduction:**

The English Premier League (EPL) is one of the most popular leagues of soccer, also known as football, in the world. It was founded in 1992 and is still hosting seasons to this day, where each of their 20 clubs plays 38 matches a season. Our team has created a database that tracks information about the EPL seasons from the last five years (2015-2019), including the matches from each season, what clubs and players were involved, the goals scored from the home and away teams, the players’ statistics, and more.

Each of our team members were interested in soccer and were motivated to build a detailed dataset that could draw interesting analyses from EPL’s matches and statistics. Although the Premier League’s data is readily available and accessible across websites such as Kaggle and the official EPL website, many sources were too complex, outdated, and tedious to understand and work with. That’s why our team aimed to provide an organized, neat, and normalized dataset that included recent and relevant information for our target audience.

Our database can benefit multiple people interested in the league, since our data can be used for making predictions and discoveries for future seasons by analyzing our matches, clubs, player\_goals, and other columns. This is especially useful for statisticians and reporters looking for patterns in past games for their predictions. Coaches, teams, and players can use this dataset to look for their strengths and weaknesses from past results in order to improve for the next upcoming season. Fans and even sponsors of the Premier League can also make analyses to prepare for future matches since they can find out who will be most likely to win, who they should support, and which teams and members they can rely on to succeed.

**Database Description:**

We have created a database that collects information about the Premier League, an English football championship that occurs every year. As such, this database can answer sports analysts’ and fans’ questions about teams’ performance and individual players. The database includes seven tables of several parts of the Premier League: Players, Player Goals, Clubs, Winners, Player Cards, Penalties and Matches.

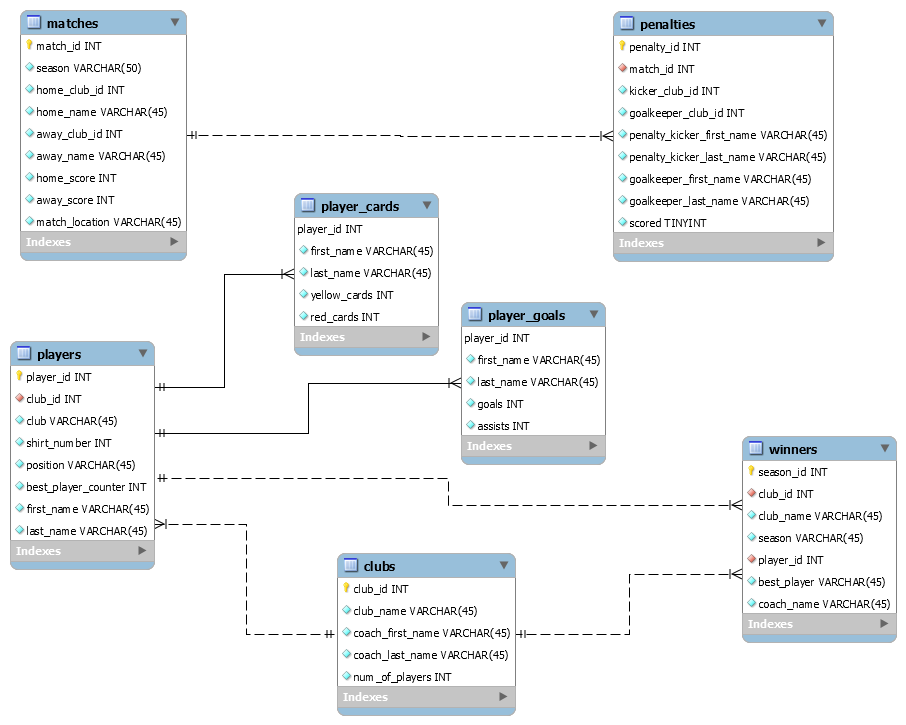
**Logical Design:**

Image 1. Entity Relationship Diagram for the English Premier League.

We designed the database tables and their relationship in a logical way that reflect the connections they have with each other in real life. First, we identified the entities and parts of the Premier League. We found that it mainly consists of: matches, clubs, and players. We added more tables to answer interesting questions about the league such as players statistics, matches statistics, and overall seasons statistics.

As our database’s ERD shows above, we established the relationship among our tables by looking at the relationship among them in real life. Our database has two main tables: matches and clubs. These are the encompassing entities of a match in the Premier League. After that, we connected other tables based on the relationship between teams, their players and their achievements. Clubs have players and stats. Individual players have their own stats separate from their club.

The last few tables that are very specific follow the same rules. Matches that result in a draw settle the winner with penalties, therefore the penalties table is connected with the matches table. On the players’ side, their own stats, such as number of cards they received and number of goals they scored, is independent from the club they are associated with. Finally, we found that overall winners of the league involves information about the team that won the championship, and a player who was awarded the best player of the season.

**Physical Database:**

Our database includes seven tables—matches, players, player\_cards, player\_goals, clubs, penalties, and winners. In terms of referencing, the players table can be seen as the focal table of our database, referencing to the most tables. It references four tables, including player\_cards, player\_goals, winners, and clubs. These references are all one-to-many, with the exception of the clubs table which serves as a linking table. The match table also shares a one-to-many relationship with the penalties table. Our approach for structuring and developing our database was through assessing how best effectively we could represent key information of the Premier League that would also allow for answering interesting questions.

Overall, our database seems to have fulfilled this, as we were able to answer many of our initially devised questions. The few of our initial questions that couldn’t be answered were due to revisions in a few table columns we’d made, however, we believe we’ve retained the most useful and interesting data. There were a few challenges we faced as well when converting our logical design into our physical database, however, we were able to resolve most of them in a timely manner by going through each step carefully.

**Sample Data:**

Our group pulled data from various csv files since we were not able to find a singular csv file that would contain all the data we needed for our project. We were able to compile the data for most of our tables from Kaggle. This data included player names, team names, players’ position, and the information regarding matches within a given time period. A majority of our data came from the two Kaggle databases that we found; however, for two of our tables we needed other sources. It became difficult to find a database that included the goals, assists, red cards, and yellow cards of a player therefore our group decided to use the official Premier League’s website in order to find this information in order to add to our database.

**Views / Queries:**

| **Query Name** | **Join** | **Filter** | **Aggregate** | **Linking** | **Sub-Query** |
| --- | --- | --- | --- | --- | --- |
| home\_away\_avg | **X** | **X** | **X** |  | **X** |
| matches\_best\_player | **X** | **X** |  |  |  |
| position\_assists | **X** | **X** | **X** |  |  |
| season\_best\_player | **X** |  |  | **X** |  |
| yellow\_cards | **X** | **X** |  |  |  |
| Total | **5** | **4** | **2** | **1** | **1** |

**Description of View / Queries:**

**home\_away\_avg:** Creates a view that compares the average score of the selected team during home and away games.

**matches\_best\_players:** Creates a view of all the players that have been the best player in a match and shows their total goals.

**position\_assists:** Creates a view of all the positions and displays that positions assists.

**season\_best\_player\_stat:** Creates a view that displays all the stats of the season’s best player including total goals, total assist, red cards and yellow cards.

**yellow\_cards:** Creates a view that shows players goals and the amount of yellow cards that player got.

**Changes from Original Design:**

Our team only made minor changes to our final design. We kept our original 7 tables but made changes to some columns to make the tables easier to understand. The first changes we made were to some of the column names. In the winners table we changed the column “year\_won” to “season”. In the matches table we changed the “date” column to “season”. We realized that in soccer, they usually refer to dates and years as season so we thought it would be better if we kept that consistent with our database. Another change we made was adding the players first and last name into our player goals table. In our progress report we decided to take it out because we thought that it was redundant, but when we were building the database we realized it was better to have. We thought that having just the player id was not enough. Since the table's main focus is about the player, a user would get the most information out of it if they knew the player's name. These minor changes allowed our database to be more understandable.

We made the most amount of changes to our penalties table. In our original design it was not clear what team was taking the penalty kick and what team was defending. So instead of having columns that give the home and away team id, we decided to have “kicker\_club\_id” and “goal\_keeper\_club\_id”. This way you are able to tell which team is benefiting from the penalty kick. In this table we also decided to add the kicker first and last name and the goalkeepers first and last name. In the progress report we decided to take the names out, but when we were making the database we realized that it was beneficial to have it in there. Adding the names made the table easier to read on its own. So that means a user wouldn’t have to flip back to the players table to find out what 2 players were involved in the penalty kick. We think that these changes really improved the penalties table.

**Lessons Learned:**

The main lesson that we learned was that our group works better together. When building the database, we realized that it was too hard to divide the work and do it separately. People had too many questions when working on their own and there was a lot of confusion. Also, there were decisions to make every step of the way regarding columns and data input. Since everyone has different skill levels, it was easier to just collaborate. We decided it was better for us to do every step of the database design together on zoom. This allowed us to get our work done more efficiently and quickly. We were able to make decisions quickly and it allowed everyone's voice to be heard. We also learned that doing too much work in one day was not effective. After 3 hours of doing work we would forget to do things like press the apply button when modifying our tables in MySQL, so when we would try to upload data it wouldn't run. This made us split the work into multiple days. We decided to only work on the project for no more than 2 hour time periods. This allowed our brains not to get tired and we were able to finish the project in a timely manner. The final lesson we learned was that it is good to do research about the tool you are using. There were a couple instances that we had to manually input data or create data. Since we had tables with thousands and thousands of rows it would take us all day. After manually inserting data for about an hour on excel, our team realized we needed to find a solution. We did some research and found excel resources such as VLOOKUP() and RAND() that allowed us to fill up our missing rows of data quickly. This is a lesson that we are all probably going to remember and use in the future.

**Potential Future Work:**

One possible expansion and continuation of our project would be to include data from all Premier League matches instead of only from the last five years. A web scraping application could be used in conjunction with this to automate the process of collecting data from older matches as well as newly added ones. While such a database would be monstrous, it would be a much more representative sample, and thus allow for more accurate predictions and analysis. Another possible (albeit audacious) continuation could be to implement an AI program that predicts future Premier League stat results based on previous data.

**References**

Maurya, Shubham. *“English Premier League Players Dataset, 2017/2018”.* Kaggle. https://www.kaggle.com/mauryashubham/english-premier-league-players-dataset. Accessed: October 11.

Nalla, Zaeem. *“Premier League”.* Kaggle. https://www.kaggle.com/zaeemnalla/premier-league. Accessed: October 11.