Database Design Project – Players Data Collection and Analysis

By – Olamilekan Razak Elegbede

1 Introduction

1.1 Background

Statistical data has rapidly become a key aspect in decision making in sports, at the club and fan level. At a club level, statistical data helps determine which player is best suited for a position, how clubs recruit players, help scouting teams to uncover and sign hidden players and has allowed teams to enjoy great success on a shoe-string budget. A perfect example of a sport club is *Brentford Football Club*. For years, this club who are backed by various companies like *Smartodd* and *Opta Analyst* using the data to sign hidden talents, moving from a second division club to a successful first division Premier League side in less than five years.

As a football fan, data has become an important on how I watch the game and in choosing my favorite players. With sports data, it helps me better analyze my favorite team and players, compare player stats to identify their strength and weaknesses, track players progression, understand why a coach has decided to move player to and from the bench, make data-driven decisions when picking players for my fantasy team rather than picking off performance and sentiment, and lastly helps my overall sports-watching experience.

1.2 Problem Statement

A lot of data sources are either paid for or do not provide the information for a football fan to make analysis on. So my aim for this project is to create a database that collects and uses statistical data from an open-sourced data website - *FBref* to track and analyze my favorite football players, compare how they performed based on playing positions, age, and make data-driven decisions.

1.3 Basis

These are the various factors put into consideration when working on the database. For this project, I performed my analysis on the following criteria:

- English Premier League clubs
- At least 5 players of high interest based on each playing position.

2 Data

2.1 Data Description

For this project, all my datasets were sourced from *FBref*. This website provides basic statistical data for over 5 leagues and 100 clubs in and out of Europe. The data meets my current demand and my future demands, which I will explain in greater detail in the Future Improvements section of this report. The data collected for this project were based on the *2021/22* and *2022/23* season, and were divided into two sections:

1. Basic information:
   a) Player information - first and last name, date of birth, nationality, and strong foot.
   b) Club information – club name, club code (gotten from first three letters of name) and league.
   c) Football information – club code, player's shirt number, last name, and club shirt (club code and shirt number).
   d) Season information – current season, number of matches played, minutes played and club code.
2. Statistical Data:
   a) Standard Data: the number of goals scored, assists provided, penalty goals scored, it also provides us with a more advanced stat, goal per 90 minutes played and the expected goal.
   b) Passing Data: all attempted passes, completed passes, key passes, shot creating action, goal creating action, number of touches and number of touches in the opponent attacking third.
   c) Defensive Data: number of yellow and red cards, ball recoveries made, tackles made, tackles won and number of interceptions.
   d) Expected Data: number of expected goals and expected data throughout various seasons.
   e) Shooting Data: total number of shots taken, shots on target, shots per 90 minutes, number of goals per shot, attempted take on and successful take on.

2.2 Database Creation

A database is like the foundation of a house. It is where all data and information about a company or business is stored. So, the creation of a database must be done with precise accuracy and be maintained with high regard to keeping a business or project going.

For this project I decided to use SQL to create my database. This tool is perfect to store large amounts of players' information and quickly extract desired data for analysis.

```
1  -- Drop any existing DATABASE
2  DROP DATABASE IF EXISTS players;
3
4  -- Create A Database (Player)
5  CREATE DATABASE IF NOT EXISTS players;
6
7  -- Use the Database
8  USE players;
```

Figure 1.0 Creating the Database

Figure 1.0 shows the SQL script we used to create our 'Player' database. This process involves dropping any previous database with the same name, creating a new one and initializing the created database.

Now that we have created our database, we first must drop any existing tables as we did for the database. Figure 2.0 below shows the SQL code used for dropping existing tables with similar names. After that go ahead to write SQL codes to the tables, assign column names and corresponding data types and assign constraints where applicable.

```sql
-- Drop any previously existing tables
DROP TABLE IF EXISTS players_info,
                     club,
                     football_info,
                     seasons,
                     shooting,
                     passing,
                     defence;
```

```sql
-- Create tables in the database

-- Table: player_info
CREATE TABLE players_info
(
    first_name   VARCHAR(30)          NOT NULL,
    last_name    VARCHAR(30)          NOT NULL PRIMARY KEY,
    birth_date   DATE,
    nationality  VARCHAR(14),
    strong_foot  ENUM('Left','Right')
);

-- Table: club
CREATE TABLE club
(
    club_code    VARCHAR(3)           NOT NULL PRIMARY KEY,
    club_name    VARCHAR(20),
    league       VARCHAR(40)
);

-- Table: football_info
CREATE TABLE football_info
(
    club_code    VARCHAR(3)           NOT NULL,
    shirt_number INT                  NOT NULL,
    position     VARCHAR(14)
);
```

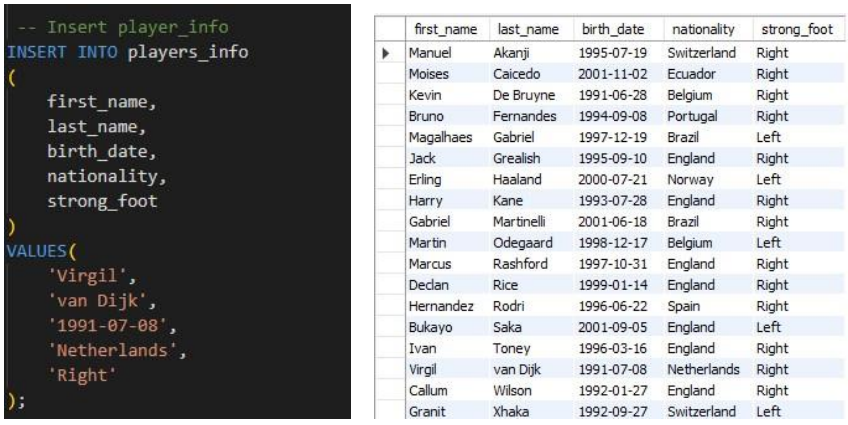
Figures 2.0. Creating Table and Fields

Now we have created the database (Players), all required tables alongside the column names. So, we go ahead to collect and load data into the tables of the database.

2.3 Data Collection and Cleaning

My data collection was all done by manually inserting the data into tables. This process was based on the two sections described in Section 2.1 *Data Description*, players basic information and statistical data.

For the basic information tables, I used MySQL in MySQL Workbench to create a table with the players and club data. I decided to go with this approach because the basic data about the player and club do not need to be updated frequently so hence, I went with that approach.



```sql
-- Insert player_info
INSERT INTO players_info
(
    first_name,
    last_name,
    birth_date,
    nationality,
    strong_foot
)
VALUES(
    'Virgil',
    'van Dijk',
    '1991-07-08',
    'Netherlands',
    'Right'
);
```

first_name	last_name	birth_date	nationality	strong_foot
Manuel	Akanji	1995-07-19	Switzerland	Right
Moises	Caicedo	2001-11-02	Ecuador	Right
Kevin	De Bruyne	1991-06-28	Belgium	Right
Bruno	Fernandes	1994-09-08	Portugal	Right
Magalhaes	Gabriel	1997-12-19	Brazil	Left
Jack	Grealish	1995-09-10	England	Right
Erling	Haaland	2000-07-21	Norway	Left
Harry	Kane	1993-07-28	England	Right
Gabriel	Martinelli	2001-06-18	Brazil	Right
Martin	Odegaard	1998-12-17	Belgium	Left
Marcus	Rashford	1997-10-31	England	Right
Declan	Rice	1999-01-14	England	Right
Hernandez	Rodri	1996-06-22	Spain	Right
Bukayo	Saka	2001-09-05	England	Left
Ivan	Toney	1996-03-16	England	Right
Virgil	van Dijk	1991-07-08	Netherlands	Right
Callum	Wilson	1992-01-27	England	Right
Granit	Xhaka	1992-09-27	Switzerland	Left

Figure 3.0 Player Information Data

Figure 3.0 shows an example of how the data was inserted into our created tables. I repeated this process for all interested players and the outcome is shown on the right-hand side.

For statistical data, because these types of data change frequently and need to be updated as soon as a player takes part in a game – usually weekly I decided to use Microsoft Excel for easier data loading. Below in Table 1.0 shows loaded data that were manually inputted in Excel with the column names corresponding to those of tables created in MySQL.

	A	B	C	D	E	F	G	H
1	club_shirt	season	total_shots	shots_on_target	shots_90	goals_shot	attempted_takeon	successful_takeon
2	ARS7	2022-2023	86	29	2.43	0.14	159	61
3	ARS7	2021-2022	95	29	2.87	0.09	102	43
4	MUN10	2021-2022	22	12	1.6	0.18	53	22
5	MUN10	2022-2023	108	49	3.38	0.35	138	54
6	TOT10	2022-2023	124	54	3.28	0.2	108	40
7	TOT10	2021-2022	129	49	3.59	0.1	100	54
8	NEW9	2021-2022	33	12	2.14	0.18	21	10
9	NEW9	2022-2023	70	31	3.36	0.21	36	12
10	MCI9	2022-2023	116	53	3.77	0.25	32	11
11	MCI9	2021-2022	74	31	3.49	0.22	24	13
12	BRE17	2021-2022	83	24	2.57	0.08	50	24
13	BRE17	2022-2023	87	33	2.65	0.16	54	16
14	MCI10	2021-2022	45	16	2.12	0.07	68	40
15	MCI10	2022-2023	41	15	1.8	0.12	110	45
16	ARS11	2022-2023	79	30	2.55	0.19	132	59
17	ARS11	2021-2022	51	15	2.47	0.1	79	45
18	MCI17	2021-2022	78	31	3.19	0.19	55	31
19	MCI17	2022-2023	65	21	2.42	0.11	61	32

Table 1.0 Shooting Data in Excel

With data collection we also must ensure our data is also clean for analysis. The process of cleaning my data was very important especially in my Excel data, it involves:

- Ensuring the data types were accurate.
- Decimal places were all set to 2 decimal places.
- Ensuring column were appropriately named.

3. Methodology

Now all data has been gathered, cleaned, and collected into the database, we proceed with analysis and getting some insights from the players data.

My goals for analysis are:

- Create a view that shows the player's areas of improvement and weakness across different seasons.
- Create a rank, ranking players based on a specific stat e.g., Ranking based on goal involvements.
- Build visualizations to better understand the data.
-

For Task 1: Creating a View

To create a view that shows the players areas of improvement and weakness across different. To achieve that, we use the standard procedure to create a view, apply mathematical calculations getting the difference (%), and use joins to apply data from various tables and seasons.

```
/*Difference Between Seasons*/
CREATE OR REPLACE VIEW attacking_season_comparison AS
SELECT
        fi.last_name,
        (s2.total_shots - s1.total_shots) as difference_shots,
        CONCAT(ROUND(((s2.shots_on_target/s2.total_shots)-(s1.shots_on_target/s1.total_shots))*100,1),'%') as difference_shot_on_target,
        ROUND((s2.goals_shot - s1.goals_shot),2) as difference_goal_shot,
        ROUND((s2.shots_90 - s1.shots_90),2) as difference_shots_90,
        (s2.attempted_takeon - s1.attempted_takeon) as difference_attempted_takeon,
        CONCAT(ROUND(((s2.successful_takeon/s2.attempted_takeon)-(s1.successful_takeon/s1.attempted_takeon))*100,1),'%') as difference_successful_takeon
FROM shooting s1
JOIN shooting s2 on s2.club_shirt = s1.club_shirt
JOIN football_info fi ON fi.club_shirt = s1.CLUB_SHIRT
WHERE s1.season = '2021-2022' AND s2.season = '2022-2023'
GROUP BY fi.club_shirt;
```

Figure 4.0 Creating a View for Attacking Data

The output of the view is seen below.

last_name	difference_shots	difference_shot_on_target	difference_goal_shot	difference_shots_90	difference_attempted_takeon	difference_successful_takeon
Saka	-9	3.2%	0.05	-0.44	57	-3.8%
Rashford	86	-9.2%	0.17	1.78	85	-2.4%
Kane	-5	5.6%	0.1	-0.31	8	-17.0%
Wilson	37	7.9%	0.03	1.22	15	-14.3%
Haaland	42	3.8%	0.03	0.28	8	-19.8%
Toney	4	9.0%	0.08	0.08	4	-18.4%
Grealish	-4	1.0%	0.05	-0.32	42	-17.9%
Martinelli	28	8.6%	0.09	0.08	53	-12.3%
De Bruyne	-13	-7.4%	-0.08	-0.77	6	-3.9%
Odegaard	40	-6.3%	0.03	0.97	37	3.5%
Fernandes	3	2.2%	-0.04	-0.08	14	-1.5%
Xhaka	13	16.5%	0.14	0.16	2	-19.5%
Rodri	5	-3.1%	-0.12	0.15	15	13.0%
Caicedo	18	-11.4%	-0.06	-0.56	44	-18.1%
Rice	9	-7.9%	0.07	0.22	-9	-13.9%
Gabriel	3	-13.3%	-0.08	-0.01	-1	-33.3%
Akanji	2	-9.7%	-0.08	0.07	3	-38.9%
van Dijk	-5	-10.0%	0.01	-0.08	-7	25.0%

Table 2.0 Difference Query Output

To understand the output, here's an example.
Saka has a '-9' in *difference_shots*. This simply means he took 9 less shots in the '2022-2023 season' as compared to the '2021-2022 season' but a *difference_shot_on_target* of 3.2% means he had a higher percentage of shots on target in '2022-2023', so with less shots taken and more of those shots being on target, we can conclude he had an improved shooting efficiency rate.

For Task 2: Create a Rank

To create a rank on the players that ranks according to a statistical value and used to measure how a player ranks compared to other players on the database. This gives more meaning to the data, and it helps with determining player preference and ranking performance.

Below is a code that we use to rank players based on the goals involvements, which is the sum of goals and assists throughout the season.

```
-- Create a rank for goals involvements
SELECT fi.last_name,
       st.club_shirt,
       se.season,
       se.matches_played,
       se.minutes_played,
       (st.goals+st.assists) as goal_involvments,
       RANK() OVER w AS ga_ranking
FROM standard st
JOIN seasons se ON se.club_shirt = st.club_shirt AND se.season = st.season
JOIN football_info fi ON fi.club_shirt = st.CLUB_SHIRT
WHERE se.season = '2022-2023'
WINDOW w AS(order by (st.goals+st.assists) desc);
```

Figure 5. 0 Creating a Rank for Goal Involvements

The output of the rank code is below:

last_name	club_shirt	season	matches_played	minutes_played	goal_involvments	ga_ranking
Haaland	MCI9	2022-2023	35	2769	44	1
Kane	TOT10	2022-2023	38	3405	33	2
Saka	ARS7	2022-2023	38	3181	25	3
Toney	BRE17	2022-2023	33	2951	24	4
Wilson	NEW9	2022-2023	31	1877	23	5
De Bruyne	MCI17	2022-2023	32	2417	23	5
Rashford	MUN10	2022-2023	35	2879	22	7
Odegaard	ARS8	2022-2023	37	3127	22	7
Martinelli	ARS11	2022-2023	36	2789	20	9
Fernandes	MUN8	2022-2023	37	3316	16	10
Xhaka	ARS34	2022-2023	37	2993	14	11
Grealish	MCI10	2022-2023	28	2055	12	12
Rodri	MCI16	2022-2023	36	2911	8	13
Rice	WHU41	2022-2023	37	3273	5	14
van Dijk	LIV4	2022-2023	32	2835	4	15
Gabriel	ARS6	2022-2023	38	3409	3	16
Caicedo	BRI25	2022-2023	37	3139	2	17
Akanji	MCI25	2022-2023	29	2287	1	18

Table 3.0 Rank Query Output

4. Results and Analysis

The goal of this project was to create a database where we store players data. From this data we can make further analysis. Below is a couple analysis we made from the dataset.

1. An interesting analysis we can perform with the dataset is plotting the goals per 90 and shots per 90 data against each other. This stat means on average the number of goals for each shot a player takes. From this graph we can conclude that:

- Erling Haaland is a clinical striker who on average scores from 1 of 4 shots he takes.
- Wilson and Harry Kane both take a good number of shots and score from them, on average 1 goal for every 6 shots.
- Ivan Toney is a striker who scores goals with little chances, so he is a good finisher.

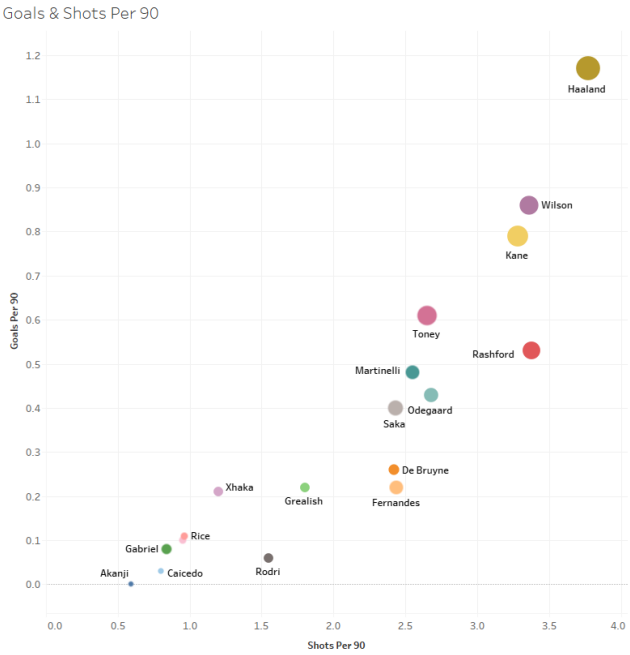


Figure Plot of Goals per 90 vs Shots per 90

We can also use the number of goals scored shown in Figure 8.0 in that season to back up our analysis.

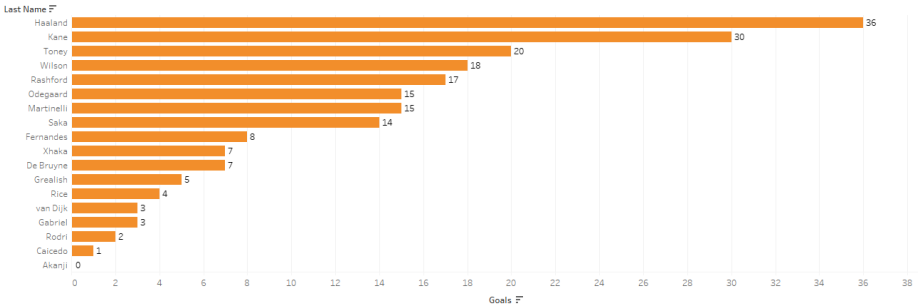


Figure 8.0 Goals Scored by each Player.

2. To better understand the comparison made in Table 2.0, here we see that Saka had a higher shot on target (%) in '2022-2023' season, leading to a greater number of goals scored. So, we can conclude taking better shots that tend to be more on target increases a players' chances of scoring more goals.

last_name	difference_shots	difference_shot_on_target	difference_goal_shot	difference_shots_90	difference_attempted_takeon	difference_successful_takeon
Saka	-9	3.2%	0.05	-0.44	57	-3.8%
Rashford	86	-9.2%	0.17	1.78	85	-2.4%
Kane	-5	5.6%	0.1	-0.31	8	-17.0%

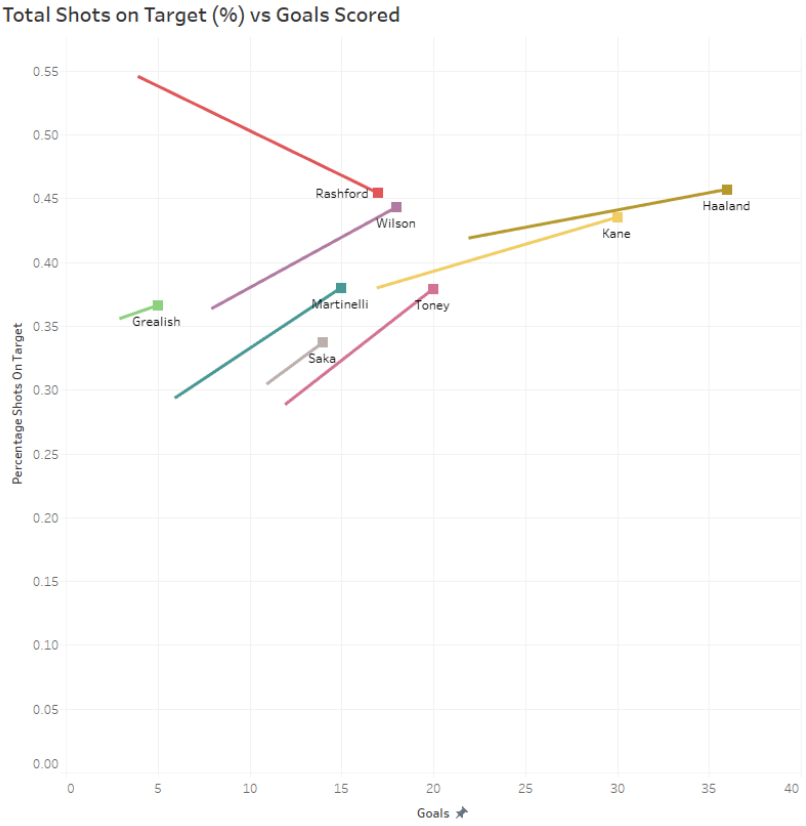


Figure 7.0 Total Shots on Target (%) vs Goals Scored

* Rashford being an outlier because as shown in the output of the SQL rank query, he took 86 more shots in the '2022-2023' season.

5. Database Backups

The primary purpose of backing up a database is to create a duplicate copy of its data and structure and not interfere with the original data. This process involves simply creating a duplicate set of the data and saving it accordingly from the data we explore.

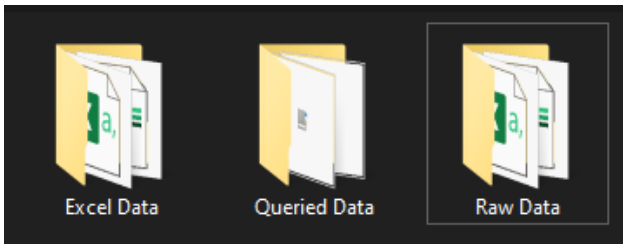


Figure Data Duplicates

My database was stored and backed up on my hard drive using MySQL local instance Data Export feature and was also stored on a cloud storage service, in this case google cloud storage.

Some other locations where a database can be backed up includes:

- On-site Servers
- Network Attached Storage
- External Hard Drives or USB Drives (depending on the size of the files)
- Cloud Storage Services – best for accessibility, data recovery and scalability.

6. Conclusion & Future Improvements

Although the sole goal for this project was to show my understanding on how a database is created and managed, which involves data collection, cleaning, tables creation and alterations, views, and database backup. This process can be considered a beginner level introduction into database creation and management.

The following are ways in which I will be improving the whole process mentioned above and make it more time and user efficient.

1. Automate data collection through web scraping – this process involves the use of Python to collect data from a website, in our case *FBref,* into Microsoft Excel or a spreadsheet. This process can be done automatically which saves more time and human effort of manually getting the data into Excel.
2. Create an interactive dashboard – make a dashboard where a user can filter and choose a specific player of interest based on different data points. To achieve this, we connect Tableau desktop to MySQL workbench where we run our desired queries and it is then inserted as a view and it automatically updates the tables in Tableau.
3. Create a predictive module – with the predictive module that Tableau provides, we can tell and choose how a player is going to perform in the future. Then from this result, we can make decisions like choosing that player into our Fantasy Premier League team as a fan and as a coach we can use that to determine player selection for future games.