



```
--Project Title: IPL_Database Project
--Created by: Abhishek Kumar Upadhyay
--Date of Creation: 26/02/2023
--Tools Used: PostgreSQL

/* About The Dataset- */

-- Question(1)- Create a table named EPL_Analysis.

CREATE TABLE EPL_Analysis (
    Players_Name VARCHAR,
    CLUB VARCHAR,
    Nationality VARCHAR,
    Position VARCHAR,
    Age INT,
    Matches INT,
    Starts INT,
    Mins INT,
    Goals INT,
    Assists INT,
    Passes_Attempted INT,
    Perc_Passes_Completed DECIMAL,
    Penalty_Goals INT,
    Penalty_Attempted INT,
    xG DECIMAL,
    xA DECIMAL,
    Yellow_Cards INT,
    Red_Cards INT,
    MinsPerMatch INT);

/* Here I am importing a "csv" file from my system. The process for importing is that

1. Create a database
2. Create a table in the database to store the data from the.csv file.
3. After that, go to the table and click right to refresh if it is not showing.
4. After that, click right on the mouse. Here, you have seen the option "Import/Export Data." Click on this.
5. Choose the file path from the file name option, then check the parameters to see that the delimiter, quote, and escape are properly entered.
6. Next, select the "columns" option to verify each column.
7. Click OK to import once again.
*/
-- Question-(2)- Write a query to show the table name. EPL_Analysis

SELECT * FROM EPL_Analysis;

---- Question-(3)-Write a query to calculate the total row from both tables.

SELECT COUNT(*) AS Total_row
FROM EPL_Analysis;

---- Question-(4)-Write a query to calculate the total column, column_name, and data_type from both tables.

-- To Show only Total_Column Count
SELECT COUNT(*) AS Total_Columns
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'epl_analysis';

--To Show the Columns_Name and their Data_Type
SELECT COLUMN_NAME, DATA_TYPE
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'epl_analysis';

/* Question-(5)- Write a query to create a new table named "EPL_Data_Analysis" with all the columns of table "EPL_Analysis." a new column, "Goals_per_Match," with a value Goals are divided by matches. */

CREATE TABLE EPL_Data_Analysis AS
SELECT *,
    ROUND(Goals::NUMERIC / Matches, 6) AS Goals_per_match
FROM EPL_Analysis;
```

SQL Queries- EPL Data Analysis

```
/* Question-(5)- Write a query to create a new table named "EPL_Data_Analysis" with all the columns of table "EPL_Analysis." a new column, "Goals_per_Match," with a value Goals are divided by matches. */

CREATE TABLE EPL_Data_Analysis AS
SELECT *,
    ROUND(Goals::NUMERIC / Matches, 6) AS Goals_per_match
FROM EPL_Analysis;

-- Question-(6)- Write a query to show the new table name. EPL_Data_Analysis

SELECT * FROM EPL_Data_Analysis;

-- Question-(7)- Write a query to calculate the total matches.

SELECT SUM(Matches) AS Total_matches
FROM EPL_Analysis;

-- Question-(8)- Write a query to calculate the total goals.

SELECT SUM(Goals) AS Total_Goals
FROM EPL_Analysis;

-- Question-(9)- Write a query to count the unique players in EPL_Analysis.

SELECT COUNT(DISTINCT Players_Name) AS Total_Players
FROM EPL_Analysis;

-- Question-(10)- Write a query to count the unique clubs in EPL_Analysis.

SELECT COUNT(DISTINCT Club)AS Total_Clubs
FROM EPL_Analysis;

-- Question-(11)- Write a query to calculate the total assists.

SELECT SUM(Assists) AS Total_Assists
FROM EPL_Analysis;

-- Question-(12)- Write a query to count the total yellow cards in EPL_Analysis.

SELECT SUM(Yellow_cards) AS Total_Yellow_Cards
FROM EPL_Analysis;

-- Question-(13)- Write a query to count the total red cards in EPL_Analysis.

SELECT SUM(Red_Cards) AS Total_Red_Cards
FROM EPL_Analysis;

-- Question-(14)- Write a query to display the five players with the highest number of goals in EPL_Analysis.

SELECT Players_Name AS "Player", SUM(Goals) AS "Total_Goals"
FROM EPL_Analysis
GROUP BY "Player"
ORDER BY "Total_Goals" DESC
LIMIT 5;

-- Question-(15)- Write a query to display the top 10 players with the highest no-issue yellow card in EPL_Analysis.

SELECT Players_Name AS "Players", SUM(Yellow_cards) AS "Total_Issue_Yellow_Cards"
FROM EPL_Analysis
GROUP BY "Players"
ORDER BY "Total_Issue_Yellow_Cards" DESC
LIMIT 10;

-- Question-(16)- Write a query to display the top 10 players with the highest number of no-issue red cards in EPL_Analysis.

SELECT Players_Name AS "Players", SUM (Red_cards) AS "Total_Red_cards"
FROM EPL_Analysis
GROUP BY "Players"
ORDER BY "Total_Red_cards" DESC
LIMIT 10;
```

SQL Queries- EPL Data Analysis

```
-- Question-(17)- In EPL_Analysis, create a query to display players who have received no red cards.Â

-- There are four methods to solve this query.

--Method-1
SELECT Players_Name AS "Players", SUM(Red_Cards) AS "Total_Issue_Red_Cards"
FROM EPL_Analysis
WHERE Red_Cards = 0
GROUP BY Players_Name
ORDER BY "Total_Issue_Red_Cards";

--Method-2
SELECT Players_Name, Red_cards
FROM EPL_Analysis
WHERE Red_Cards = 0;

/*We obtained different results from both methods. We got 477 from the first method and 485 from the second method. The variation between Both outcomes have the duplicate players name. The second method counts the duplicate value, means the original value and duplicate value count 1+1.Also, as in the first method, we group the players by name so that duplicate and original values count only once, and the variation between both results is 7 players, whose names are repeated.

In short- The difference in the count between the two queries is most likely due to the fact that the first query is grouping the players by name, and hence any duplicate player names are being counted only once, while the second query is simply checking for the presence of a zero value in the Red_Cards column, which could potentially result in counting duplicate player names more than once.

To avoid this issue and get an accurate count of unique players with zero red cards, you can modify the second query to use the DISTINCT keyword to ensure that each player name is only counted once: See the below query.*/

--Method-3
SELECT COUNT(DISTINCT Players_Name) AS Players
FROM EPL_Analysis
WHERE Red_Cards = 0

--Method-4
SELECT COUNT(DISTINCT Players_Name) AS Players, Red_cards
FROM EPL_Analysis
WHERE Red_Cards = 0
GROUP BY Red_Cards

--Note- And finally, we got the result: 477.

-- Question-(18)- In EPL_Analysis, create a query to display players who have received no yellow cards.Â

--Method-1
SELECT Players_Name as "Players", SUM(Yellow_Cards) AS "Total_Yellow_Cards"
FROM EPL_Analysis
WHERE Yellow_Cards = 0
GROUP BY "Players"
ORDER BY "Total_Yellow_Cards";

--Method-2
SELECT Players_Name as Players, Yellow_Cards
FROM EPL_Analysis
WHERE Yellow_Cards = 0;

/*We obtained different results from both methods. We got 172 from the first method and 174 from the second method. The variation between Both outcomes have the duplicate players name. The second method counts the duplicate value, means the original value and duplicate value count 1+1.Also, as in the first method, we group the players by name so that duplicate and original values count only once, and the variation between both results is 2 players, whose names are repeated.

In short- The difference in the count between the two queries is most likely due to the fact that the first query is grouping the players by name, and hence any duplicate player names are being counted only once, while the second query is simply checking for the presence of a zero value in the Yellow_Cards column, which could potentially result in counting duplicate player names more than once.

To avoid this issue and get an accurate count of unique players with zero red cards, you can modify the second query to use the DISTINCT keyword to ensure that each player name is only counted once: See the below query.*/

--Method-3
SELECT COUNT(DISTINCT Players_Name) as "Players", Yellow_Cards
FROM EPL_Analysis
WHERE Yellow_Cards = 0
GROUP BY Yellow_Cards;

--Method-4
SELECT COUNT(DISTINCT Players_Name) AS "Players"
FROM EPL_Analysis
WHERE Yellow_Cards = 0;

--Note- And finally, we got the result: 172.
```



```
-- Question-(19)- Write a query to display the total position in the EPL analysis.

SELECT DISTINCT(Position)
FROM EPL_Analysis;

-- Question-(20)- Write a query to display the total position in the EPL analysis.

SELECT Players_Name AS "Players", Position
FROM EPL_Analysis
GROUP BY Players_Name, Position;

/* -- Question-(21)- Write a query to display the total number of unique players and their total goals and matches, average minutes per match,
and average goals per match. For this query, you need to join two tables, EPL_Analysis and EPL_Data_Analysis, because the
The "goals per match" column is in the second table.*/

SELECT A.Players_Name, SUM(A.Goals) AS Total_Goals, COUNT(A.Matches) AS Total_Matches,
       AVG(B.Minspermatch) AS Avg_Minspermatch, AVG(B.goals_per_match) AS Avg_goals_per_match
    FROM EPL_Analysis AS A
    LEFT JOIN EPL_Data_Analysis AS B
      ON A.Players_Name = B.Players_Name
   GROUP BY A.Players_Name
ORDER BY Avg_goals_per_match DESC;

-- Question-(22)-Write a query to display the top 5 players with highest no of assists.

SELECT Players_Name AS Players, SUM(Assists) AS Total_Assists
FROM EPL_Analysis
GROUP BY Players
ORDER BY Total_Assists DESC
LIMIT 5;

-- Question-(23)-Write a query to display the top 5 clubs with the highest number of assists.

SELECT Club, SUM(Goals) AS Total_Goals
FROM EPL_Analysis
GROUP BY Club
ORDER BY Total_Goals DESC
LIMIT 5;

--Question-(24)- Create a query to display players' positions.

SELECT DISTINCT Players_Name, Position
FROM EPL_Analysis
ORDER BY Players_Name ASC;

--Question-(25)- Write a query to display the unique positions from the EPL_Data_Analysis table.

SELECT DISTINCT(Position) AS Players_Position_IN_EPL
FROM EPL_Data_Analysis;

--Question-(26)- Write a query to display the total number of players whose position is only "FW."

SELECT DISTINCT(Players_Name) AS Players, Position
FROM EPL_Analysis
WHERE Position IN ('FW');

--Question-(27)- Write a query to provide the ranking by total goals to clubs.

SELECT Club, SUM(Goals) AS Total_Goals, RANK() OVER (ORDER BY SUM(Goals) DESC) AS Rank
FROM EPL_Analysis
GROUP BY Club
ORDER BY Total_Goals DESC;

--We can use the above query by DENSE_RANK()
SELECT Club, SUM(Goals) AS Total_Goals, DENSE_RANK() OVER (ORDER BY SUM(Goals) DESC) AS Rank
FROM EPL_Analysis
GROUP BY CLUB
ORDER BY Total_Goals DESC;

/*The Difference between both the query is that RANK() function provide the 1 rank to 1st row and rows having same value assigned
same rank and one rank value is skipped after associating the same rank to the row, and DENSE_RANL() does not skip it; it provides the next
rank after two same-ranked assignments.*/
```



```
/*Question-(28) Write a query to provide the total number of goals scored by players of each nationality. Where the position is "DF", "DF,FW", "DF,MF", "FW" */

SELECT Nationality, SUM(Goals) AS Total_Goals
FROM EPL_Analysis
WHERE Position IN ('DF', 'DF,FW', 'DF,MF', 'FW')
GROUP BY Nationality
ORDER BY Total_Goals DESC;

/*Question-(29) Write a query that retrieves the club and the maximum number of goals scored by a player for that club, taking into account only players with the positions "DF", "DF,FW", "DF,MF", or "FW". */

SELECT Club, MAX(Goals) AS Top_Scorer
FROM EPL_Analysis
WHERE Position IN ('DF', 'DF,FW', 'DF,MF', 'FW')
GROUP BY Club
ORDER BY Top_Scorer DESC;

/*Question-(29) Write a query to retrieves the club, nationality, and the maximum number of goals scored by a player for each combination of club and nationality in the EPL_Analysis table, taking into account only players with the positions 'DF', 'DF,FW', 'DF,MF', or 'FW'. */

SELECT
    Club, Nationality,
    MAX(Goals) AS Top_Scorer
FROM
    EPL_Analysis
WHERE Position IN ('DF', 'DF,FW', 'DF,MF', 'FW')
GROUP BY Club, Nationality
ORDER BY Top_Scorer DESC;

/*Question-(30)- Write a query to find the correlation between goals and xG. */

SELECT CORR (Goals,xG) AS Correlation_Between_xG_VS_Goals
FROM EPL_Analysis;

/*Question-(30)- Write a query to find the correlation between xG and xA */

SELECT CORR(xG,xA) AS Correlation_Between_xG_VS_xA
FROM EPL_Analysis;

--Question-(31)- Write a query to display the total players by age group.

SELECT CASE
WHEN Age >=16 AND Age <=20 THEN '16-20'
WHEN Age >=21 AND Age <=25 THEN '21-25'
WHEN Age >= 26 AND Age <= 30 THEN '26-30'
WHEN Age >= 31 AND Age <= 35 THEN '31-35'
ELSE '35+' END AS Age_Group,
COUNT(DISTINCT Players_Name) AS Total_Unique_Players
FROM EPL_Analysis
GROUP BY Age_Group
ORDER BY Age_Group;

--Question-(32)- Create a query to show Total_penalty_goals and Penalty_Not_Goals.

SELECT SUM(Penalty_Goals) AS Total_penalty_Goals ,
SUM(Penalty_Attempted) - SUM(Penalty_Goals) AS Penalty_Not_Goals
FROM EPL_Analysis;

--Question-(33)- Write a query to display the total penalty goals and penalty attempts.

SELECT SUM(Penalty_Goals) AS Total_Penalty_Goals,
SUM(Penalty_Attempted) AS Total_Penalty_Attempted
FROM EPL_Analysis;

--Question-(34)- Create a query to display the Total_Penalty_Attempted and Penalty_Not_Goals statistics.

SELECT SUM(Penalty_Attempted) AS Total_Penalty_Attempted,
SUM(Penalty_Attempted) - SUM(Penalty_Goals) AS Penalty_Not_Goals
FROM EPL_Analysis;
```



The English Premier League (EPL) 2020–2021 season was the 29th season of the Premier League since its establishment in 1992. The season started on September 12, 2020, and ended on May 23, 2021.

Manchester City were crowned champions for the fifth time, finishing 12 points clear of second-placed Manchester United. They won 27 of their 38 league matches, scoring 83 goals and conceding only 32.

The top four teams in the league qualified for the UEFA Champions League, which included Manchester City, Manchester United, Liverpool, and Chelsea. Leicester City, West Ham United, and Tottenham Hotspur secured the Europa League places by finishing in fifth, sixth, and seventh places, respectively.

The three teams that were relegated to the Championship were Fulham, West Bromwich Albion, and Sheffield United. The three teams promoted from the Championship to the Premier League for the 2021–2022 season were Norwich City, Watford, and Brentford.

The top goal scorer of the season was Harry Kane of Tottenham Hotspur with 23 goals, followed by Mohamed Salah of Liverpool with 22 goals. The player with the most assists was Harry Kane with 14 assists. The Premier League Player of the Season award was won by Manchester City's Ruben Dias.

477 different players have not been issued a red card, while 172 players have not been issued a yellow card.

In the analysis we got there are 10 unique position for the players.

There are 81 players whose position is only "FW."

"Harry Kane" and "Mohamed Salah" are in first and second place, respectively, after players were ranked by total goals scored.

"Manchester City" and "Manchester United" are in first and second place, respectively, after clubs were ranked by total goals scored.

There are 195 players whose age group between 26-30.

A correlation between Goals and xG is 0.681959 suggests a moderately strong positive correlation between goals and expected goals (xG). Therefore, in this case, the moderately strong positive correlation indicates that there is a meaningful relationship between goals and xG, and we can use xG to make predictions about future goal-scoring performance with a reasonable level of confidence.

The season was played behind closed doors due to the COVID-19 pandemic. However, fans were allowed to return to stadiums towards the end of the season in limited numbers.