Predicting Premier League Performance through Player and Team Statistics

Jinuk Seo[1], Wonjune Lee[1]
[1]*Computational and Data Sciences, George Mason University, Incheon, Republic of Korea*
{jseo24, wlee40}@gmu.edu

Keywords: Premier League, Football Analytics, SQL

Abstract: This project explores how SQL-based data analysis can help predict team success in the English Premier League. Using multiple relational tables containing player statistics, team standings, and expected goals, we investigate the relationship between individual performance and overall team results. We designed a normalized database in SQLite and implemented advanced SQL techniques, including JOINs, aggregate functions, nested queries, string operations, and the WITH clause. Our research focused on identifying key factors that correlate with league performance—such as whether teams with top scorers rank higher or how expected goals reflect actual outcomes. We also proposed a simple predictive model based on metrics like average player rating, possession, and xG. Our findings suggest that data-driven analysis can highlight performance trends and support early predictions of future champions. This paper demonstrates the value of relational databases and SQL for sports analytics and provides a foundation for further exploration using larger datasets or machine learning.

1 INTRODUCTION

The English Premier League (EPL) is one of the most competitive and popular football leagues in the world. With millions of fans and billions in revenue, teams constantly look for ways to gain a competitive edge. In recent years, data analytics has become a powerful tool in professional football, helping clubs improve tactics, recruitment, and performance analysis.

This project investigates how SQL-based relational databases can be used to analyze team and player performance in the EPL and make predictions about future outcomes. Specifically, we ask: *Do teams with top scorers tend to perform better? How does expected goals (xG) relate to actual performance? Can we find patterns that point to next season's possible winners?*

To explore these questions, we designed a structured database using SQLite and imported multiple datasets related to the 2023–2024 Premier League season. We then used SQL queries to join, group, and compare key metrics, such as goals scored, player ratings, possession, and xG.

2 METHODOLOGY / DATABASE DESIGN

To effectively represent and analyze the 2023–2024 Premier League season data, we designed a relational database grounded in an Entity-Relationship (ER) model. This model captures the relationships between individual players, teams, and their performances using six key entities:

- player_top_scorers
- player_player_ratings
- player_big_chances_missed
- player_expected_goals
- possession_percentage_team
- pl_table_2023_24

The central entity in our schema is pl_table_2023_24, which stores each team's season summary including total points, goal differences, and match results. All other tables are connected to it using the Team field as a foreign key.

Each player-based table (e.g., player_expected_goals, player_top_scorers) maintains a one-to-many (1:N) relationship with the central pl_table_2023_24 table. This design allows us to retrieve aggregate and individual performance metrics for players within their team context.

We also normalized our schema to minimize redundancy. For instance, player-level performance metrics are separated by category (e.g., big chances missed vs. expected goals), which provides flexibility in data analysis and ensures data integrity. The primary keys and foreign keys across the schema help enforce referential consistency and enable efficient join operations for more complex queries.

3 SQL QUERY AND ANALYSIS

To gain deeper insights into how individual and team performances relate to final league outcomes, we implemented several SQL queries covering various analytical perspectives. Each query was designed to showcase different SQL capabilities, such as aggregation, set operations, string functions, grouping, and nested subqueries.

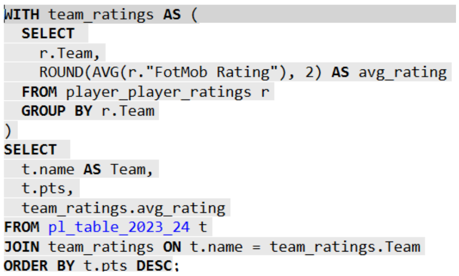
1. **Top Scorers vs Team Rank**
   This query used a nested subquery to find the highest scorer on each team and join it with team performance data. It highlighted whether having a top scorer correlates with team success.
   Concepts used: Nested queries, joins, ordering, grouping.

```
SELECT
    p."Team",
    p."Player",
    p."Goals",
    t.pts,
    t.goalConDiff
FROM player_top_scorers p
JOIN pl_table_2023_24 t
    ON p."Team" = t.name
WHERE p."Goals" = (
    SELECT MAX(p2."Goals")
    FROM player_top_scorers p2
    WHERE p2."Team" = p."Team"
)
ORDER BY t.idx;
```

2. **Actual vs Expected Goals Efficiency**
   We used a WITH clause to calculate the difference between each player's actual and expected goals (xG). This revealed who over- or under-performed their expected numbers.
   Concepts used: CTEs (WITH), aggregate functions, arithmetic operations.

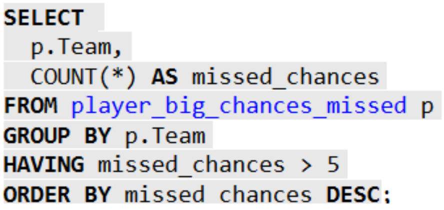
```
WITH goal_efficiency AS (
    SELECT
        "Player", "Team",
        "Actual Goals",
        "Expected Goals (xG)",
        ROUND("Actual Goals" - "Expected Goals (xG)", 2) AS difference
    FROM player_expected_goals
)
SELECT *
FROM goal_efficiency
ORDER BY difference DESC
LIMIT 10;
```

3. **Team Possession vs Expected Goals**
   Using a common table expression (CTE), we combined possession and xG data per team and ranked teams based on these combined metrics.
   Concepts used: WITH clause, joins, grouping, aggregate functions.



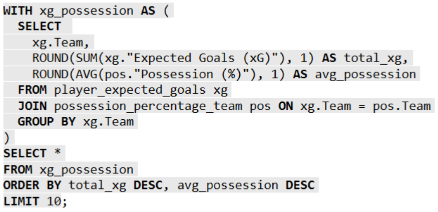
```sql
WITH team_ratings AS (
  SELECT
    r.Team,
    ROUND(AVG(r."FotMob Rating"), 2) AS avg_rating
  FROM player_player_ratings r
  GROUP BY r.Team
)
SELECT
  t.name AS Team,
  t.pts,
  team_ratings.avg_rating
FROM pl_table_2023_24 t
JOIN team_ratings ON t.name = team_ratings.Team
ORDER BY t.pts DESC;
```

4. **Average Player Ratings vs League Points** We computed the average FotMob rating per team and compared it with their final points. This analysis helped determine if higher-rated squads achieve better results. *Concepts used: joins, aggregate functions, grouping, ordering.*



```sql
SELECT
  p.Team,
  COUNT(*) AS missed_chances
FROM player_big_chances_missed p
GROUP BY p.Team
HAVING missed_chances > 5
ORDER BY missed_chances DESC;
```

5. **Shot Conversion and Big Chance Misses** We examined if teams that missed fewer big chances had a better shot conversion rate or better standings. *Concepts used: joins, HAVING clause, set membership, filtering.*



```sql
WITH xg_possession AS (
  SELECT
    xg.Team,
    ROUND(SUM(xg."Expected Goals (xG)"), 1) AS total_xg,
    ROUND(AVG(pos."Possession (%)"), 1) AS avg_possession
  FROM player_expected_goals xg
  JOIN possession_percentage_team pos ON xg.Team = pos.Team
  GROUP BY xg.Team
)
SELECT *
FROM xg_possession
ORDER BY total_xg DESC, avg_possession DESC
LIMIT 10;
```

Through these queries, we were able to not only validate intuitive football knowledge (e.g., better players contribute to team success) but also identify outliers and over-performers. The structured schema and SQL analysis together enabled a robust and scalable framework for football performance prediction.

4 QUERY RESULT ANALYSIS

4.1 Top Scorers and Team Performance

Interpretation: This query shows the leading goal scorers from each team and compares their goal count with their team's overall league performance—measured by points and goal difference.

Insight: Teams with high-performing strikers generally rank higher in the league. For example, Manchester City's Erling Haaland scored 27 goals, contributing to the team's 91 points and a +62 goal difference.

4.2 Goal Efficiency vs Expected Goals (xG)

Interpretation: This query compares each player's actual goals with their expected goals (xG), which represents the quality of scoring opportunities.

Insight: Players like Cole Palmer (+3.8) significantly overperformed their xG, indicating strong finishing skills. In contrast, Darwin Núñez underperformed (–5.4), suggesting missed chances. This metric highlights goal-scoring efficiency.

4.3 Team Ratings vs League Points

Interpretation: This query connects average player ratings for each team with total league points.

Insight: Teams with higher average player ratings tend to finish higher in the table. Manchester City, with a rating of 7.24, achieved the highest point total (91), reinforcing the value of consistently high individual performances.

4.4 Attacking Efficiency (Top 5 Scorers' Average Goals)

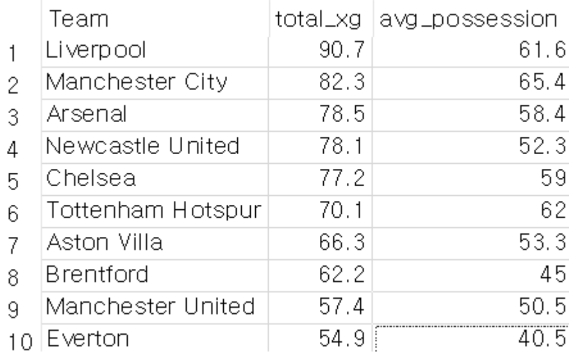
Interpretation: This shows the average number of goals scored by each team's top five attacking players.

Insight: Teams with balanced scoring across multiple attackers exhibit stronger overall attacking efficiency. Manchester City led with an average of 13.5 goals among its top five scorers.

4.5 XG and Possession Strength

Interpretation: This compares each team's total xG with their average possession percentage.

Insight: Teams that maintain higher possession, such as Manchester City (65.4%), also tend to generate more high-quality chances (xG). This suggests a strategic advantage through ball control and sustained offensive pressure.



	Team	total_xg	avg_possession
1	Liverpool	90.7	61.6
2	Manchester City	82.3	65.4
3	Arsenal	78.5	58.4
4	Newcastle United	78.1	52.3
5	Chelsea	77.2	59
6	Tottenham Hotspur	70.1	62
7	Aston Villa	66.3	53.3
8	Brentford	62.2	45
9	Manchester United	57.4	50.5
10	Everton	54.9	40.5

5. DISCUSSION AND CONCLUSION

This study explored how player-level and team-level performance indicators relate to success in the Premier League and how they can help predict future outcomes. By building a normalized SQL database and running analytical queries, we were able to uncover several important trends in the 2023–2024 season.

First, our findings confirmed a strong correlation between top scorers and overall team performance. Teams with efficient goal scorers like Erling Haaland and Cole Palmer consistently ranked higher in points and goal difference, suggesting that having a reliable striker significantly boosts team results.

Second, the expected goals (xG) analysis showed that goal efficiency varies widely among players. While some players outperform their xG, others fall short, revealing valuable insights for scouting and tactical planning. Similarly, our ratings and possession analysis highlighted that high average player ratings and ball control often align with better team performance.

Taken together, these results suggest that a combination of attacking depth, player efficiency, and team consistency are critical to achieving top rankings. Teams that consistently maintain high possession, generate quality scoring chances (high xG), and spread goals across multiple players are better positioned for long-term success.

As a final note, while this project used historical data to identify patterns, future research could include machine learning models for actual prediction. Incorporating more seasons, player injuries, or transfer data could also enhance the accuracy of forecasting next season's league champion.

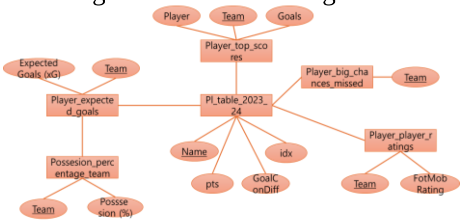
In conclusion, data analytics provides a powerful lens through which to evaluate football performance. Our SQL-based approach not only helped us uncover key success factors but also offered a practical framework for future strategic decision-making in sports analytics.

REFERENCES

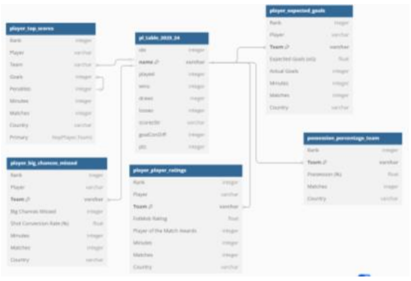
Ali, K. (2024). *Premier League 23/24.* Kaggle. https://www.kaggle.com/datasets/whisperingkahuna/premier-league-2324-team-and-player-insights

APPENDIX

E-R Diagram for Premier League Database Project



Relational Schemas


player_top_scores
Rank integer
Player varchar
Team varchar
Goals integer
Penalties integer
Minutes integer
Matches integer
Country varchar
Primary Key(Player, Team)
pl_table_2023_24
idx integer
name varchar
played integer
wins integer
draws integer
losses integer
scoresStr varchar
goalConDiff integer
pts integer
player_expected_goals
Rank integer
Player varchar
Team varchar
Expected Goals (xG) float
Actual Goals integer
Minutes integer
Matches integer
Country varchar
possession_percentage_team
Rank integer
Team varchar
Possession (%) float
Matches integer
Country varchar
player_big_chances_missed
Rank integer
Player varchar
Team varchar
Big Chances Missed integer
Shot Conversion Rate (%) float
Minutes integer
Matches integer
Country varchar
player_player_ratings
Rank integer
Player varchar
Team varchar
Fotblob Rating float
Player of the Match Awards integer
Minutes integer
Matches integer
Country varchar