Advanced Database Systems Midterm Coursework report

Student Name: Mohamed Fazid
UOL Student ID: 230656806
Module: CM3010 Databases and Advanced Data Techniques
Date: 7 January 2026

Links

- **Dataset URL:**
    1. **Season 2020/2021:** https://www.football-data.co.uk/englandm.php#:~:text=Season%202020/2021-,Premier%20League,-(FT%20%26%20HT%20results
    2. **Season 2021/2022:** https://www.football-data.co.uk/englandm.php#:~:text=Season%202021/2022-,Premier%20League,-(FT%20%26%20HT%20results
    3. **Season 2022/2023:** https://www.football-data.co.uk/englandm.php#:~:text=Season%202022/2023-,Premier%20League,-(FT%20%26%20HT%20results
    4. **Season 2023/2024:** https://www.football-data.co.uk/englandm.php#:~:text=Season%202023/2024-,Premier%20League,-(FT%20%26%20HT%20results
    5. **Season 2024/2025:** https://www.football-data.co.uk/englandm.php#:~:text=Season%202024/2025-,Premier%20League,-(FT%20%26%20HT%20results

- **Lab environment (MySQL + Node app):**
    1. https://hub.labs.coursera.org/connect/sharedyfnenhio?forceRefresh=false&path=%2F%3Ffolder%3D%2Fhome%2Fcoder%2Fproject&sessionMigrationMode=shadow

Introduction

This project applies relational database design and web application development to a real world premier league football dataset covering 5 seasons of league competition. The dataset contains structured information about matches, teams, seasons, referees, disciplinary events and bookmaker odds making it particularly well suited to a relational database approach.

The primary aim of the project is to design and implement a MySQL database that accurately models the underlying structure of premier league match data and supports analytical queries across seasons. By normalising the data into related entities, the database enables efficient querying of trends such as home advantage, team performance, disciplinary behaviour and bookmaker prediction accuracy and many more.

To demonstrate the practical use of the database, a simple Node.js web application is developed. The application connects to the database and presents selected analytical queries through a user facing interface, illustrating how relational data can be explored beyond static analysis or spreadsheet based approaches.

The project is structured into four stages by selecting and critiquing the dataset, modelling the data using an Entity Relationship approach, implementing the database in MySQL with real instance data and developing a web application to query and present the results. Together, these stages demonstrate the end to end process of transforming an open dataset into a functional database driven application.

Stage 1 - Dataset Selection and Critique

1.1 Dataset Description

The dataset used in this project consists of historical football match data for the English Premier League, sourced from the open data repository *football-data.co.uk* (https://www.football-data.co.uk/englandm.php). The data covers 5 consecutive seasons, from 2020-2021 to 2024-2025, with each season provided as a separate CSV file.

Each dataset records match level information, including participating teams, match dates, full-time results and scores. In addition to core match outcomes, the data includes contextual and behavioural attributes such as referees, disciplinary events (fouls, yellow cards and red cards) and betting odds from multiple bookmakers. This combination allows sporting performance to be analysed alongside officiating behaviour and betting market expectations.

The data is structured in a tabular format, with one row per match and a consistent set of attributes across seasons, making it suitable for transformation into a relational database. The dataset is published as open data and is explicitly intended for public analysis and reuse, subject to the terms stated on the source website.

1.2 Dataset Assessment

Quality

The dataset is generally of high quality and suitable for structured database analysis. Core attributes such as teams, match results, dates and seasons are consistently recorded across all five CSV files. This consistency allows matches to be reliably compared both within individual seasons and across multiple seasons.

Some missing values are present, most notably in betting odds and certain officiating related fields. Not all bookmakers provide odds for every match and some referee related attributes are absent in specific fixtures. These gaps reflect real world data availability rather than data corruption. During database modelling this required the use of nullable fields and careful handling of joins to ensure valid matches were not unintentionally excluded from queries.

No systematic inconsistencies were observed in key outcome data such as goals scored or full time results. This increases confidence in the reliability of analytical queries derived from the database. The dataset is published by a well known football statistics provider that is widely used for analysis and research which further supports its reliability.

Level of Detail

The data is recorded at match level with each row representing a single football match. This level of granularity is well suited to the research questions posed in this project because it allows individual match outcomes to be analysed while also supporting aggregation across teams referees and seasons.

The dataset is neither too coarse nor excessively detailed. It provides sufficient contextual information such as disciplinary events and betting odds without introducing unnecessary low level event data such as minute by minute actions. This balance supports efficient relational modelling while maintaining analytical usefulness.

Documentation

Basic documentation is available on the dataset source website including explanations of column names and abbreviations particularly for match outcomes and betting odds. Common football notation is used consistently across seasons which aids interpretation once the conventions are understood.

However the documentation is relatively limited and assumes some prior domain knowledge of football and betting markets. Certain abbreviations for disciplinary statistics or betting odds require interpretation. Despite this the documentation was sufficient to enable accurate modelling and use of the data.

Interrelation

The dataset naturally decomposes into several interrelated entities including teams, matches seasons, referees and bookmakers. Repeated identifiers such as team names appearing across many matches and seasons clearly indicate an underlying relational structure.

These relationships make the dataset particularly suitable for an Entity Relationship modelling approach. Matches form the central entity linking teams seasons referees and betting odds which aligns well with a normalised relational database design.

Use

The dataset is intended for analysis of football matches and related contextual factors. It supports a range of analytical use cases including evaluation of team performance, investigation of home advantage analysis of disciplinary behaviour and assessment of bookmaker prediction accuracy.

Because the dataset spans multiple seasons and combines sporting and betting related attributes it enables longitudinal analysis and comparative studies that benefit from structured querying and aggregation.

Discoverability

The dataset is publicly available and easily discoverable through the football data website. Data is clearly organised by league and season with direct download links for each CSV file. No authentication or special access is required which makes the dataset readily accessible for academic use.

Terms of Use

According to the source website the dataset is provided as open data and is intended for public analysis and reuse. The stated terms permit non commercial and academic use including coursework and research projects. This confirms that the dataset can be legitimately used for this assignment.

1.3 Motivation and Interest

Football analytics is an established and commercially significant field that benefits from structured analysis of large volumes of match data. Understanding patterns in team performance, home advantage and disciplinary behaviour and bookmaker prediction accuracy provides valuable insight into how football competitions operate both on the pitch and within betting markets.

This dataset is of particular interest because it combines sporting outcomes behavioural data and bookmaker expectations across multiple seasons. Using a relational database allows these factors to be analysed systematically through aggregation and comparison across teams seasons and matches. Such analysis would be inefficient and difficult to maintain using flat files alone which motivates the use of a database driven approach.

1.4 Research Questions

The research questions in this project were chosen to explore performance behavioural and predictive aspects of football that emerge only when match data is analysed across multiple seasons and entities. Each question is designed to require structured querying aggregation and comparison which motivates the use of a relational database.

Q1 Which teams demonstrate the highest long term consistency across five seasons
This question was chosen to investigate sustained performance rather than isolated success in a single season. Consistency is an important indicator of team quality and stability but cannot be assessed using individual match results alone. The question uses match outcomes, team identifiers and season data to calculate points or results across all five seasons. Answering this question requires aggregating results by team and season and comparing variation over time which is well suited to database queries.

Q2 Which teams significantly over or under perform relative to bookmaker expectations across five seasons
This question explores whether teams consistently exceed or fall short of market expectations. It was motivated by the inclusion of bookmaker odds in the dataset and the interest in evaluating betting market accuracy. The analysis combines match results with bookmaker odds and season information to compare expected outcomes with actual results. This requires joining match data with bookmaker data and aggregating results over multiple seasons which is difficult to manage using flat files.

Q3 Which teams are most effective at turning losing half time positions into full time results
This question was chosen to examine team resilience and tactical effectiveness. It focuses on matches where teams are losing at half time but recover to draw or win by full time. The data used includes half time scores, full time scores, team identifiers and season

information. Answering this question requires conditional filtering and grouping across many matches which benefits from a relational database structure.

Q4 Do teams with more shots on target convert pressure into wins or do inefficiencies appear
This question investigates whether attacking pressure translates into successful outcomes. It was motivated by the availability of shots on target data alongside match results. The analysis uses match level statistics such as shots on target goals scored and full time results to evaluate conversion efficiency. Aggregating these measures across teams and seasons requires structured queries and careful grouping.

Q5 How have scoring patterns evolved across five seasons overall home away and by match period
This question examines changes in scoring behaviour over time. It uses goal data split by home and away teams and by half of the match across all seasons. The analysis requires grouping by season team location and match period which involves multiple aggregations and comparisons that are naturally handled within a relational database.

Q6 Are bookmakers becoming more accurate at predicting match outcomes over time
This question was motivated by the inclusion of multi season bookmaker odds and an interest in how predictive accuracy evolves. The data used includes implied probabilities derived from betting odds, actual match results and season identifiers. Answering this question requires calculating accuracy metrics per season and comparing trends over time which depends on aggregating large volumes of structured data.

Q7 Is aggressive play linked to success or does it harm performance
This question explores the relationship between discipline and match outcomes. It uses fouls, yellow cards, red cards and match results to assess whether aggressive behaviour correlates with improved or reduced performance. The analysis requires combining disciplinary data with outcome data and aggregating across teams and seasons which supports the use of a database approach.

Q8 Do certain referees consistently award more cards and does this influence match outcomes
This question was chosen to examine potential officiating patterns and their impact on matches. It uses referee identifiers, disciplinary data and match results to compare referees across many fixtures. The question requires grouping by referee and analysing trends across seasons which is not practical to manage using spreadsheets alone.

Overall these questions require joining multiple entities including matches teams seasons referees and bookmakers and performing aggregation filtering and comparison across large datasets. This justifies the use of a relational database and a database driven web application rather than simple scripts or flat file analysis.

1.5 Dataset Attributes Used in This Project

Although the source dataset contains a wide range of match related attributes only a subset of columns was selected for modelling and analysis in this project. The choice of attributes was guided by the research questions and the need to support structured querying and aggregation across multiple seasons.

Core match information was derived from columns recording the match date and participating teams including Match Date, Home Team and Away Team. Match outcomes were captured using Full Time Home Goals (FTHG), Full Time Away Goals (FTAG) and Full Time Result (FTR). These attributes form the basis for analysing team performance scoring patterns and long term consistency across seasons. Season information was derived from the source file structure and stored explicitly to enable grouping and comparison across the five seasons included in the dataset.

Half time performance was analysed using Half Time Home Goals (HTHG), Half Time Away Goals (HTAG) and Half Time Result (HTR). These attributes were required to investigate match momentum and the ability of teams to recover from losing positions at half time and convert these situations into positive full time outcomes.

Disciplinary behaviour was analysed using Home Team Fouls Committed (HF), Away Team Fouls Committed (AF), Home Team Yellow Cards (HY), Away Team Yellow Cards (AY), Home Team Red Cards (HR) and Away Team Red Cards (AR). These columns were selected to support analysis of aggressive play and its relationship with match success as well as to examine refereeing patterns. Referee identity was captured using Referee Name which allowed disciplinary outcomes to be linked to individual officials across multiple matches and seasons.

Attacking pressure and efficiency were analysed using shooting statistics including Home Team Shots (HS), Away Team Shots (AS), Home Team Shots on Target (HST) and Away Team Shots on Target (AST). These attributes were used to evaluate whether sustained attacking pressure translates into goals and match wins or whether inefficiencies are present.

Bookmaker expectations were captured using betting odds provided by bookmakers particularly Bet365 Home Win Odds (B365H), Bet365 Draw Odds (B365D) and Bet365 Away Win Odds (B365A). These columns were used to derive implied match outcome probabilities and to assess bookmaker prediction accuracy when compared with actual match results.

Columns that were not directly relevant to the research questions or that introduced unnecessary complexity were excluded from the database design. This selective approach ensured that the resulting schema remained focused while still supporting all planned analytical queries across performance discipline refereeing and betting accuracy.

1.6 Stage 1 Summary

This stage identified and critically evaluated a real world open dataset suitable for relational database implementation. The English Premier League match data was found to be of generally high quality with consistent recording of core attributes such as match outcomes, team participation and season structure across five consecutive seasons.

Key limitations including missing betting odds for some matches and limited documentation were identified and critically assessed. These limitations reflect real world data collection constraints rather than fundamental flaws and informed modelling decisions such as the use of nullable attributes and careful query design.

A focused subset of dataset attributes was selected including match results scoring data half time information disciplinary records refereeing data shooting statistics and bookmaker odds. These attributes directly support the research questions developed in this stage and provide sufficient relational structure for meaningful multi season analysis.

Overall the dataset naturally decomposes into interrelated entities which strongly motivates the use of an Entity Relationship model and a normalised relational schema. This stage therefore establishes a clear and coherent foundation for the data modelling and database implementation presented in the following stages.

Stage 2 - Data Modelling

2.1 Conceptual E R Model

A conceptual Entity Relationship model was developed to represent the structure of the Premier League match dataset and to support the research questions through structured querying across seasons, teams, referees and bookmaker data. The model is centred on the Match entity which represents an individual football fixture and acts as the core event linking all other entities.

Entities identified in the model

- League
- Season
- Team
- Referee
- Bookmaker
- Match
- Match Odds

Entity roles

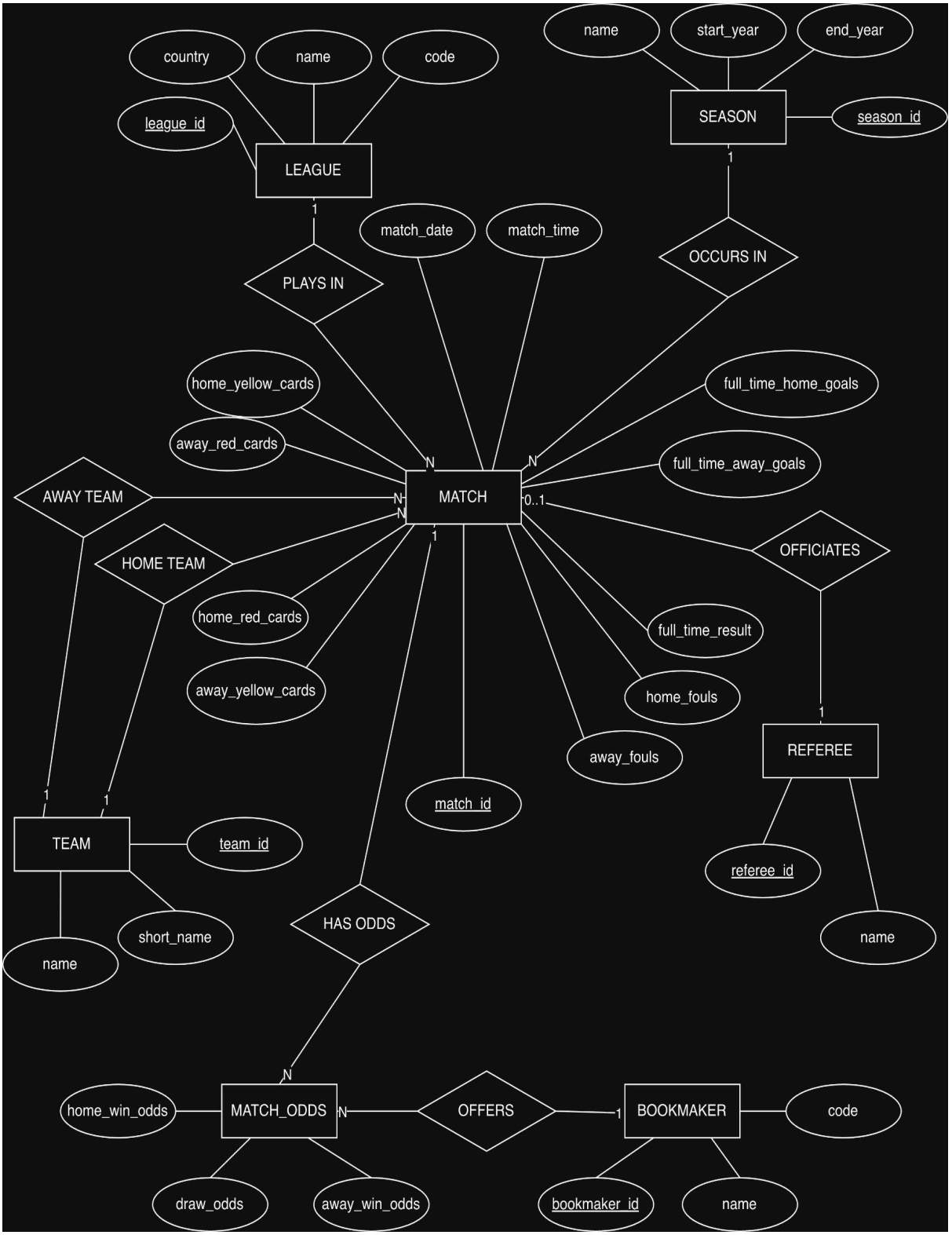
- League represents the competition context including league code name and country
- Season represents the temporal grouping of matches using a season identifier and year range
- Team represents football clubs participating in matches
- Referee represents match officials who may officiate multiple fixtures
- Bookmaker represents betting providers offering match outcome odds
- Match represents individual fixtures including results and match statistics
- Match Odds represents bookmaker specific odds associated with a match

Key modelling decisions shown in the diagram

- Match is the central entity connected to all other entities
- Home and away team participation is modelled using two distinct relationships
- Referee involvement is optional and linked directly to Match
- Betting odds are separated into a dedicated associative entity
- Match attributes include only match level facts and statistics

Only attributes required to answer the research questions were included in the conceptual model including match outcomes, goals, half time, results, shooting statistics, disciplinary data, referee identity and bookmaker odds. Additional source attributes were excluded to avoid unnecessary complexity.

2.2 Conceptual E R Diagram



2.3 Cardinality and Relational Compatibility

Cardinality constraints are explicitly shown in the conceptual E R diagram and were used to guide the relational implementation. The Match entity forms the centre of the model and participates in all major relationships.

League to Match

- One League is associated with many Matches
- Each Match belongs to exactly one League

This reflects the fact that a competition contains many fixtures while each fixture is played within a single league.

Season to Match

- One Season is associated with many Matches
- Each Match occurs in exactly one Season

This supports grouping and comparison of matches across multiple seasons.

Team to Match

- One Team participates in many Matches
- Each Match has exactly one Home Team
- Each Match has exactly one Away Team

Home and away roles are modelled as separate relationships rather than a many to many structure. This design choice simplifies queries related to home advantage team performance and match outcomes.

Referee to Match

- One Referee may officiate many Matches
- A Match may have zero or one Referee

Referee participation is optional because referee information is missing for some fixtures. This is represented in the model using optional cardinality.

Bookmaker to Match

- One Bookmaker provides odds for many Matches
- One Match may have odds from many Bookmakers

This many to many relationship is resolved using the Match Odds associative entity which stores outcome odds per bookmaker per match.

Relational compatibility

- Many to many relationships are resolved using associative entities
- Optional relationships are explicitly modelled
- No repeating groups or multivalued attributes are present

The conceptual model maps directly to the relational schema without modification. As a result no separate relational E R diagram was required.

2.4 Relational Schema

The relational schema was derived directly from the conceptual E R model and implemented in MySQL. Each table corresponds to an entity or associative entity shown in the diagram.

League

- Stores competition identity and metadata
- Primary key league_id

Season

- Stores season name and year range
- Primary key season_id

Team

- Stores football club identity
- Primary key team_id
- Referenced twice by Match to represent home and away participation

Referee

- Stores referee identity
- Primary key referee_id

Bookmaker

- Stores betting provider identity
- Primary key bookmaker_id

Match

- Central fact table
- Stores match date league season teams results goals and match statistics
- Foreign keys link Match to League Season Team and Referee

Match Odds

- Associative table between Match and Bookmaker
- Stores bookmaker specific home draw and away odds
- Composite primary key match_id bookmaker_id

Primary keys enforce entity integrity and foreign keys enforce referential integrity across the schema. This structure supports reliable multi table queries aligned with the conceptual model.

2.5 Normalisation Analysis

The relational schema was analysed against the standard normal forms to ensure structural correctness minimise data redundancy and support reliable multi table querying. Normalisation was applied to a level that balances theoretical soundness with practical usability for analytical queries. The resulting design avoids update insertion and deletion anomalies while remaining efficient and readable for the types of joins and aggregations required in this project.

First Normal Form 1NF

The database satisfies First Normal Form because all attributes contain atomic values and each table represents a single entity or relationship instance. There are no repeating groups or multivalued attributes within any table. Match related data such as goals, shots, fouls corners and disciplinary events are stored as individual scalar attributes rather than as combined or list based values. Each table includes a clearly defined primary key which uniquely identifies each record and enforces entity integrity.

Second Normal Form 2NF

The schema satisfies Second Normal Form because all non key attributes are fully functionally dependent on the entire primary key of their respective tables. Most tables in the schema use a single attribute primary key and therefore satisfy this requirement trivially. In the case of the match_odds table which uses a composite primary key consisting of match_id and bookmaker_id all non key attributes representing betting odds depend on the combination of both identifiers. There are no partial dependencies where an attribute depends on only one component of a composite key.

Third Normal Form 3NF

The database satisfies Third Normal Form because there are no transitive dependencies between non key attributes. Descriptive information is stored only in the table to which it logically belongs and is referenced elsewhere using foreign keys. For example team referee and bookmaker names are stored in their respective entity tables rather than being duplicated in the match table. The match table itself contains only foreign keys linking to related entities and factual match data such as scores, results and match statistics. This design prevents update anomalies and ensures that changes to descriptive data are made in a single location.

The database therefore satisfies at least Third Normal Form. Further normalisation such as Boyce Codd Normal Form or Fourth Normal Form was not applied because the schema does not exhibit multivalued dependencies or overlapping candidate keys. Additional decomposition would increase query complexity without providing meaningful reductions in redundancy and would negatively impact the clarity and efficiency of analytical queries used in this project.

Stage 3 - Database Implementation

3.1 Database Creation

The database was implemented in MySQL within the lab environment using the script schema.sql. A dedicated database named football_analysis was created using the InnoDB storage engine and UTF8MB4 character encoding to support consistent storage of text data and to ensure transactional integrity. The implementation directly reflects the conceptual data model developed in Stage 2.

Schema Implementation Overview

Before presenting the SQL implementation the following points summarise how the conceptual model was realised in the relational schema.

- Core domain entities including league season team referee and bookmaker were implemented as separate tables using surrogate integer primary keys
- The match table was implemented as a central fact table capturing match level outcomes and statistics and linking all contextual entities through foreign keys
- Home and away team roles were explicitly modelled using two foreign keys referencing the team table
- Referee participation was implemented as an optional foreign key to accommodate missing officiating data
- Bookmaker odds were normalised into a separate associative table to resolve the many to many relationship between matches and bookmakers
- A composite primary key was used in the match_odds table to ensure a single odds record per bookmaker per match

The following code sections illustrate how these design decisions were implemented in SQL.

Section of code

Database setup and core entity tables

This code creates the database and defines the core entity tables used to store leagues seasons teams referees and bookmakers.

```
CREATE DATABASE IF NOT EXISTS football_analysis DEFAULT CHARACTER SET utf8mb4
DEFAULT COLLATE utf8mb4_unicode_ci;

USE football_analysis;

CREATE TABLE league (league_id INT AUTO_INCREMENT PRIMARY KEY, code VARCHAR(10) NOT
NULL, name VARCHAR(100) NOT NULL, country VARCHAR(100) NOT NULL, UNIQUE (code))
ENGINE=InnoDB;

CREATE TABLE season (season_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(20) NOT
NULL, start_year SMALLINT NOT NULL, end_year SMALLINT NOT NULL, UNIQUE (name))
ENGINE=InnoDB;

CREATE TABLE team (team_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT
NULL, short_name VARCHAR(20) NULL, UNIQUE (name)) ENGINE=InnoDB;

CREATE TABLE referee (referee_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100)
NOT NULL, UNIQUE (name)) ENGINE=InnoDB;

CREATE TABLE bookmaker (bookmaker_id INT AUTO_INCREMENT PRIMARY KEY, code
VARCHAR(20) NOT NULL, name VARCHAR(100) NOT NULL, UNIQUE (code)) ENGINE=InnoDB;

INSERT INTO bookmaker (code,name) VALUES ('B365','Bet365') ON DUPLICATE KEY UPDATE
name=VALUES(name);
```

Central fact table

This code defines the match table which stores match level facts and links all related entities using foreign keys.

```
CREATE TABLE `match` (

match_id INT AUTO_INCREMENT PRIMARY KEY,

league_id INT NOT NULL, season_id INT NOT NULL,

match_date DATE NOT NULL, match_time TIME NULL,

home_team_id INT NOT NULL, away_team_id INT NOT NULL,
```

```
full_time_home_goals TINYINT NOT NULL, full_time_away_goals TINYINT NOT NULL,
full_time_result ENUM('H','D','A') NOT NULL,

half_time_home_goals TINYINT NOT NULL, half_time_away_goals TINYINT NOT NULL,
half_time_result ENUM('H','D','A') NULL,

referee_id INT NULL,

home_shots SMALLINT NULL, away_shots SMALLINT NULL,

home_shots_on_target SMALLINT NULL, away_shots_on_target SMALLINT NULL,

home_fouls SMALLINT NULL, away_fouls SMALLINT NULL,

home_corners SMALLINT NULL, away_corners SMALLINT NULL,

home_yellow_cards SMALLINT NULL, away_yellow_cards SMALLINT NULL,

home_red_cards SMALLINT NULL, away_red_cards SMALLINT NULL,

original_div_code VARCHAR(10) NULL,

CONSTRAINT fk_match_league FOREIGN KEY (league_id) REFERENCES league (league_id),

CONSTRAINT fk_match_season FOREIGN KEY (season_id) REFERENCES season (season_id),

CONSTRAINT fk_match_home_team FOREIGN KEY (home_team_id) REFERENCES team (team_id),

CONSTRAINT fk_match_away_team FOREIGN KEY (away_team_id) REFERENCES team (team_id),

CONSTRAINT fk_match_referee FOREIGN KEY (referee_id) REFERENCES referee
(referee_id)

) ENGINE=InnoDB;
```

Associative table for bookmaker odds

This code defines the match_odds table which resolves the many to many relationship between matches and bookmakers.

```
CREATE TABLE match_odds (

match_id INT NOT NULL, bookmaker_id INT NOT NULL,

home_win_odds DECIMAL(6,2) NULL, draw_odds DECIMAL(6,2) NULL, away_win_odds
DECIMAL(6,2) NULL,

PRIMARY KEY (match_id,bookmaker_id),
```

```
CONSTRAINT fk_odds_match FOREIGN KEY (match_id) REFERENCES `match` (match_id) ON
DELETE CASCADE,

CONSTRAINT fk_odds_bookmaker FOREIGN KEY (bookmaker_id) REFERENCES bookmaker
(bookmaker_id)

) ENGINE=InnoDB;
```

3.2 Data Insertion

Data was inserted using a scripted loading pipeline in load_all.sql that combines LOAD DATA LOCAL INFILE with controlled insert statements into the normalised tables. The loading process was designed to be repeatable across five seasons and to ensure all tables and most fields are populated many times so that multi table joins and aggregations are meaningful.

The insertion workflow follows a consistent pattern:

- Reference data is inserted for the Premier League and the five seasons
- Existing data is cleared in a safe order to allow the script to be re run without duplication
- A staging table called ‘raw_match’ is created to mirror the source CSV structure
- Each season CSV is loaded into the staging table
- Unique teams and referees are inserted into their respective tables
- Match records are inserted into the central ‘match’ table with validation and type conversion
- Bookmaker odds are inserted into ‘match_odds’ by linking each raw record to its corresponding match

Basic preprocessing is handled during transformation rather than by modifying the source CSV files. Dates are converted using STR_TO_DATE. Outcome fields are cleaned using TRIM. Half time result values are validated and set to null if not recognised. Referee values are filtered so empty strings do not create invalid records. Nullable fields in the schema allow matches to be loaded even when some statistics or referee details are missing.

The database contains sufficient instance data because all five seasons are loaded and each season contains a full set of match rows. This ensures repeated usage of the league

season team match and odds tables and provides adequate data density to support the analytical queries in Stage 3.4.

Code snippet

Snippet A reference data setup

This snippet inserts the league record and the five seasons used throughout the dataset.

```
USE football_analysis;


INSERT INTO league (code, name, country)
VALUES ('E0', 'Premier League', 'England')
ON DUPLICATE KEY UPDATE
 name = VALUES(name),
 country = VALUES(country);


INSERT INTO season (name, start_year, end_year)
VALUES
('2020-2021', 2020, 2021),
('2021-2022', 2021, 2022),
('2022-2023', 2022, 2023),
('2023-2024', 2023, 2024),
('2024-2025', 2024, 2025)
ON DUPLICATE KEY UPDATE
 start_year = VALUES(start_year),
 end_year   = VALUES(end_year);
```

Snippet B reset block for rerunnable loads
This snippet clears dependent tables in a safe order so the script can be executed multiple times.

```
-- ------------------------------------

-- 1. Reset data so script is re-runnable

-- ------------------------------------

SET FOREIGN_KEY_CHECKS = 0;

TRUNCATE TABLE match_odds;

TRUNCATE TABLE `match`;

TRUNCATE TABLE referee;

TRUNCATE TABLE team;

SET FOREIGN_KEY_CHECKS = 1;
```

Snippet C staging table definition
This snippet creates the raw staging table that mirrors the key CSV columns required for transformation.

```
-- ------------------------------------

-- 2. Staging table (mirrors CSV columns)

-- ------------------------------------

DROP TABLE IF EXISTS raw_match;


CREATE TABLE raw_match (

   `Div`   VARCHAR(10),

   `Date`  VARCHAR(20),

   HomeTeam VARCHAR(100),

   AwayTeam VARCHAR(100),


   FTHG TINYINT,

   FTAG TINYINT,

   FTR  VARCHAR(5),
```

```
  HTHG TINYINT,
  HTAG TINYINT,
  HTR  VARCHAR(5),

  Referee VARCHAR(100),

  HS  SMALLINT,
  `AS` SMALLINT,
  HST SMALLINT,
  AST SMALLINT,
  HF  SMALLINT,
  AF  SMALLINT,
  HC  SMALLINT,
  AC  SMALLINT,
  HY  SMALLINT,
  AY  SMALLINT,
  HR  SMALLINT,
  AR  SMALLINT,

  B365H DECIMAL(6,2),
  B365D DECIMAL(6,2),
  B365A DECIMAL(6,2)
);
```

Snippet D One season loading example

This snippet shows the complete loading pattern for a single season using the 2020-2021 CSV file as an example. The same sequence of operations is reused for each of the five seasons in the dataset.

```
-- ***************

-- SEASON 2020-2021

-- ***************

TRUNCATE TABLE raw_match;



LOAD DATA LOCAL INFILE '/home/coder/project/DBWT midterms/Data/Season
2020:2021.csv'

INTO TABLE raw_match

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 LINES

(

 `Div`, `Date`,

 HomeTeam, AwayTeam,

 FTHG, FTAG, FTR,

 HTHG, HTAG, HTR,

 Referee,

 HS, `AS`, HST, AST,

 HF, AF, HC, AC,

 HY, AY, HR, AR,

 B365H, B365D, B365A

);



-- teams & referees from this season

INSERT IGNORE INTO team (name)

SELECT DISTINCT HomeTeam FROM raw_match
```

```sql
UNION
SELECT DISTINCT AwayTeam FROM raw_match;


INSERT IGNORE INTO referee (name)
SELECT DISTINCT Referee
FROM raw_match
WHERE Referee IS NOT NULL AND Referee <> '';


-- matches
INSERT INTO `match` (
   league_id, season_id,
   match_date, match_time,
   home_team_id, away_team_id,
   full_time_home_goals, full_time_away_goals, full_time_result,
   half_time_home_goals, half_time_away_goals, half_time_result,
   referee_id,
   home_shots, away_shots,
   home_shots_on_target, away_shots_on_target,
   home_fouls, away_fouls,
   home_corners, away_corners,
   home_yellow_cards, away_yellow_cards,
   home_red_cards, away_red_cards,
   original_div_code
)
SELECT
   l.league_id,
   s.season_id,
   STR_TO_DATE(r.`Date`, '%d/%m/%Y'),
   NULL,
   ht.team_id,
   at.team_id,
```

```
   r.FTHG,
   r.FTAG,
   TRIM(r.FTR),
   r.HTHG,
   r.HTAG,
   CASE WHEN TRIM(r.HTR) IN ('H','D','A') THEN TRIM(r.HTR) ELSE NULL END,
   ref.referee_id,
   r.HS,
   r.`AS`,
   r.HST,
   r.AST,
   r.HF,
   r.AF,
   r.HC,
   r.AC,
   r.HY,
   r.AY,
   r.HR,
   r.AR,
   r.`Div`
FROM raw_match r
JOIN league l ON l.code = 'E0'
JOIN season s ON s.name = '2020-2021'
JOIN team ht ON ht.name = r.HomeTeam
JOIN team at ON at.name = r.AwayTeam
LEFT JOIN referee ref ON ref.name = r.Referee
WHERE TRIM(r.FTR) IN ('H','D','A');


-- odds
INSERT INTO match_odds (
   match_id, bookmaker_id,
```

```
   home_win_odds, draw_odds, away_win_odds
)
SELECT
   m.match_id,
   b.bookmaker_id,
   r.B365H, r.B365D, r.B365A
FROM raw_match r
JOIN league l ON l.code = 'E0'
JOIN season s ON s.name = '2020-2021'
JOIN team ht ON ht.name = r.HomeTeam
JOIN team at ON at.name = r.AwayTeam
JOIN `match` m
 ON m.league_id = l.league_id
AND m.season_id = s.season_id
AND m.home_team_id = ht.team_id
AND m.away_team_id = at.team_id
AND m.match_date = STR_TO_DATE(r.`Date`, '%d/%m/%Y')
JOIN bookmaker b ON b.code = 'B365';
```

3.3 Critical Reflection

The implemented database reflects the structure and content of the source dataset effectively and supports all research questions identified in Stage 1. The relational design allows matches, teams, referees, seasons and bookmaker odds to be queried consistently across five seasons. However, several limitations were observed during implementation. These limitations arise from both the characteristics of the dataset and deliberate modelling decisions.

Data Completeness and Availability

One limitation is the incomplete availability of certain attributes in the source data.

- Referee information is missing for some fixtures, which required the referee relationship in the ‘match’ table to be optional
- This preserves all match records but limits the completeness of referee based analysis because not all matches can be attributed to an official
- Bookmaker odds are not available for every match, even for the same bookmaker
- As a result, analyses involving betting accuracy or implied probabilities operate on a subset of matches rather than the full dataset

These issues reflect real world data availability and reporting practices rather than errors in the database design.

Granularity of Match Statistics

A second limitation relates to the level of detail provided by the dataset.

- Match statistics are recorded as aggregates at match level
- Attributes such as shots, fouls and cards do not include timing or player attribution
- This prevents more detailed analyses such as identifying when events occur during a match or which players are responsible

This limitation is inherent to the dataset structure and constrains the modelling scope to match level analysis only.

Modelling Scope and Extensibility Trade Offs

There are also trade-offs between modelling simplicity and future extensibility.

- The schema was intentionally limited to attributes required to answer the research questions
- Additional betting markets, alternative bookmakers and detailed event level data were excluded to avoid unnecessary complexity
- While this improves schema clarity and query performance, extending the system would require additional tables and data sources

Overall Assessment

Overall, these limitations do not prevent the database from fulfilling its intended purpose. Instead, they demonstrate realistic constraints imposed by real world data and show that modelling decisions were made consciously to balance data quality, analytical usefulness and implementation complexity.

3.4 Queries Addressing Research Questions

Each research question identified in Stage 1 was addressed using SQL queries executed against the implemented database. The queries make extensive use of joins, aggregation, window functions and conditional logic to analyse match outcomes, performance trends, behavioural patterns and bookmaker accuracy across five seasons.

Question 1

Which teams demonstrate the highest long term consistency across five seasons?

Question 1 SQL query:

```
WITH team_points AS (
   SELECT
       t.team_id,
       t.name AS team_name,
       s.name AS season,
       SUM(
           CASE
               WHEN m.home_team_id = t.team_id AND m.full_time_result = 'H' THEN 3
```

```
                WHEN m.away_team_id = t.team_id AND m.full_time_result = 'A' THEN 3
                WHEN m.full_time_result = 'D' THEN 1
                ELSE 0
            END
        ) AS total_points
    FROM `match` m
    JOIN season s ON m.season_id = s.season_id
    JOIN team t ON t.team_id IN (m.home_team_id, m.away_team_id)
    GROUP BY t.team_id, t.name, s.name
),
team_stats AS (
    SELECT
        team_name,
        AVG(total_points) AS avg_points,
        STDDEV(total_points) AS variation,
        COUNT(*) AS seasons_played
    FROM team_points
    GROUP BY team_name
    HAVING COUNT(*) = 5
)
SELECT
    team_name,
    avg_points,
    variation,
    (avg_points / (1 + variation)) AS consistency_score
FROM team_stats
```

```
ORDER BY consistency_score DESC;
```

This query measures team consistency by calculating total points per season for each team and then aggregating those values across all five seasons. Consistency is assessed using two metrics: average points per season and the variation in points across seasons. Teams with high average points and low variation are ranked highest using a derived consistency score.

The query joins the match, team and season tables and assigns points based on match outcomes. A common table expression is used to calculate per season totals, followed by a second aggregation to compute long term averages and variation. Only teams appearing in all five seasons are included to ensure fair comparison.

Question 2

Which teams significantly over or under perform relative to bookmaker expectations across five seasons?

Question 2 SQL query:

```
WITH match_probs AS (
   SELECT
       m.match_id,
       m.season_id,
       s.name AS season_name,
       m.home_team_id,
       m.away_team_id,
       m.full_time_result,
       mo.home_win_odds,
       mo.draw_odds,
       mo.away_win_odds,
       (1.0 / mo.home_win_odds) AS inv_h,
       (1.0 / mo.draw_odds) AS inv_d,
```

```
       (1.0 / mo.away_win_odds) AS inv_a
   FROM `match` m
   JOIN match_odds mo ON mo.match_id = m.match_id
   JOIN season s ON s.season_id = m.season_id
   JOIN bookmaker b ON b.bookmaker_id = mo.bookmaker_id
   WHERE b.code = 'B365'
     AND mo.home_win_odds IS NOT NULL
     AND mo.draw_odds IS NOT NULL
     AND mo.away_win_odds IS NOT NULL
     AND mo.home_win_odds  > 0
     AND mo.draw_odds      > 0
     AND mo.away_win_odds  > 0
),
match_probs_norm AS (
   SELECT
       mp.*,
       (mp.inv_h + mp.inv_d + mp.inv_a) AS denom,
       (mp.inv_h / (mp.inv_h + mp.inv_d + mp.inv_a)) AS p_home,
       (mp.inv_d / (mp.inv_h + mp.inv_d + mp.inv_a)) AS p_draw,
       (mp.inv_a / (mp.inv_h + mp.inv_d + mp.inv_a)) AS p_away
   FROM match_probs mp
),
team_match_points AS (
   -- home team perspective
   SELECT
       mpn.season_id,
```

```
    mpn.season_name,
    t_home.team_id,
    t_home.name AS team_name,
    mpn.match_id,
    (3 * mpn.p_home + 1 * mpn.p_draw) AS expected_points,
    CASE
        WHEN mpn.full_time_result = 'H' THEN 3
        WHEN mpn.full_time_result = 'D' THEN 1
        ELSE 0
    END AS actual_points
FROM match_probs_norm mpn
JOIN team t_home ON t_home.team_id = mpn.home_team_id

UNION ALL

-- away team perspective
SELECT
    mpn.season_id,
    mpn.season_name,
    t_away.team_id,
    t_away.name AS team_name,
    mpn.match_id,
    (3 * mpn.p_away + 1 * mpn.p_draw) AS expected_points,
    CASE
        WHEN mpn.full_time_result = 'A' THEN 3
        WHEN mpn.full_time_result = 'D' THEN 1
```

```
            ELSE 0

        END AS actual_points

    FROM match_probs_norm mpn

    JOIN team t_away ON t_away.team_id = mpn.away_team_id

)

SELECT

    tmp.team_name,

    COUNT(*) AS matches_considered,

    COUNT(DISTINCT tmp.season_id) AS seasons_covered,

    ROUND(SUM(tmp.actual_points), 2) AS total_actual_points,

    ROUND(SUM(tmp.expected_points), 2) AS total_expected_points,

    ROUND(SUM(tmp.actual_points) / COUNT(*), 3) AS avg_actual_points_per_game,

    ROUND(SUM(tmp.expected_points) / COUNT(*), 3) AS avg_expected_points_per_game,

    ROUND(SUM(tmp.actual_points) - SUM(tmp.expected_points), 3) AS
points_over_expected,

    ROUND(

        (SUM(tmp.actual_points) - SUM(tmp.expected_points)) / COUNT(*),

        3

    ) AS points_over_expected_per_game

FROM team_match_points tmp

GROUP BY tmp.team_id, tmp.team_name

HAVING COUNT(DISTINCT tmp.season_id) = 5

ORDER BY points_over_expected_per_game DESC;
```

This query compares actual points earned by teams with expected points derived from bookmaker odds. Implied probabilities are calculated from Bet365 odds and normalised to

account for bookmaker margin. Expected points per match are then calculated and compared with actual match outcomes.

The query separates home and away perspectives to ensure expected values are computed correctly for both teams in each match. Results are aggregated across all seasons to identify teams that consistently outperform or underperform expectations. This analysis demonstrates the use of probabilistic modelling combined with relational joins and aggregation.

Question 3

Which teams are most effective at turning losing half time positions into full time results?

Question 3 SQL query:

```
WITH losing_positions AS (
   -- Home team losing at half-time
   SELECT
       m.match_id,
       s.name AS season_name,
       th.team_id,
       th.name AS team_name,
       'H' AS side,
       m.full_time_result,
       1 AS losing_ht,
       CASE WHEN m.full_time_result = 'H' THEN 1 ELSE 0 END AS turned_to_win,
       CASE WHEN m.full_time_result = 'D' THEN 1 ELSE 0 END AS turned_to_draw,
       CASE WHEN m.full_time_result IN ('H','D') THEN 1 ELSE 0 END AS
turned_to_non_loss
   FROM `match` m
   JOIN season s ON s.season_id = m.season_id
   JOIN team th ON th.team_id = m.home_team_id
```

```
    WHERE m.half_time_result = 'A'

    UNION ALL

    -- Away team losing at half-time
    SELECT
        m.match_id,
        s.name AS season_name,
        ta.team_id,
        ta.name AS team_name,
        'A' AS side,
        m.full_time_result,
        1 AS losing_ht,
        CASE WHEN m.full_time_result = 'A' THEN 1 ELSE 0 END AS turned_to_win,
        CASE WHEN m.full_time_result = 'D' THEN 1 ELSE 0 END AS turned_to_draw,
        CASE WHEN m.full_time_result IN ('A','D') THEN 1 ELSE 0 END AS
turned_to_non_loss
    FROM `match` m
    JOIN season s ON s.season_id = m.season_id
    JOIN team ta ON ta.team_id = m.away_team_id
    WHERE m.half_time_result = 'H'
)
SELECT
    lp.team_name,
    COUNT(*) AS losing_ht_games,
    SUM(lp.turned_to_win)       AS full_comebacks_to_win,
```

```
    SUM(lp.turned_to_draw)      AS comebacks_to_draw,
    SUM(lp.turned_to_non_loss)  AS comebacks_to_non_loss,
    ROUND(SUM(lp.turned_to_win) * 100.0 / COUNT(*), 2)      AS win_comeback_pct,
    ROUND(SUM(lp.turned_to_draw) * 100.0 / COUNT(*), 2)     AS draw_comeback_pct,
    ROUND(SUM(lp.turned_to_non_loss) * 100.0 / COUNT(*), 2) AS non_loss_comeback_pct
FROM losing_positions lp
GROUP BY lp.team_id, lp.team_name
HAVING COUNT(*) >= 5
ORDER BY win_comeback_pct DESC, non_loss_comeback_pct DESC;
```

This query identifies matches where teams were losing at half time and evaluates whether they recovered to draw or win by full time. Home and away cases are handled separately and then combined to create a unified view of comeback performance.

The output reports the number of losing half time situations per team along with the proportion converted into wins, draws and non losses. This supports analysis of resilience and tactical effectiveness using conditional logic and grouping across many matches.

Question 4

Do teams with more shots on target convert pressure into wins or do inefficiencies appear?

Question 4 SQL query:

```
WITH shot_diff AS (
    SELECT
        m.match_id,
        m.season_id,
        m.home_team_id,
        m.away_team_id,
        m.home_shots_on_target,
```

```
        m.away_shots_on_target,
        m.full_time_result,
        CASE
            WHEN m.home_shots_on_target > m.away_shots_on_target THEN 'H'
            WHEN m.away_shots_on_target > m.home_shots_on_target THEN 'A'
            ELSE 'T'
        END AS pressure_side
    FROM `match` m
),
pressure_team_view AS (
    SELECT
        CASE
            WHEN sd.pressure_side = 'H' THEN sd.home_team_id
            WHEN sd.pressure_side = 'A' THEN sd.away_team_id
        END AS team_id,
        sd.full_time_result,
        sd.pressure_side
    FROM shot_diff sd
    WHERE sd.pressure_side <> 'T'
)
SELECT
    t.name AS team_name,
    COUNT(*) AS matches_with_pressure,
    SUM(
        CASE
            WHEN (ptv.pressure_side = 'H' AND ptv.full_time_result = 'H')
```

```
            OR (ptv.pressure_side = 'A' AND ptv.full_time_result = 'A')
          THEN 1 ELSE 0 END
   ) AS wins_with_pressure,
   SUM(CASE WHEN ptv.full_time_result = 'D' THEN 1 ELSE 0 END) AS
draws_with_pressure,
   SUM(
       CASE
          WHEN ptv.full_time_result IN ('H','A')
           AND NOT (
              (ptv.pressure_side = 'H' AND ptv.full_time_result = 'H') OR
              (ptv.pressure_side = 'A' AND ptv.full_time_result = 'A')
           )
          THEN 1 ELSE 0 END
   ) AS losses_despite_pressure,
   ROUND(
       SUM(
          CASE
             WHEN (ptv.pressure_side = 'H' AND ptv.full_time_result = 'H')
               OR (ptv.pressure_side = 'A' AND ptv.full_time_result = 'A')
             THEN 1 ELSE 0 END
       ) * 100.0 / COUNT(*), 2
   ) AS win_rate_with_pressure_pct,
   ROUND(SUM(CASE WHEN ptv.full_time_result = 'D' THEN 1 ELSE 0 END) * 100.0 /
COUNT(*), 2)
       AS draw_rate_with_pressure_pct,
   ROUND(
       SUM(
```

```
            CASE
                WHEN ptv.full_time_result IN ('H','A')
                 AND NOT (
                    (ptv.pressure_side = 'H' AND ptv.full_time_result = 'H') OR
                    (ptv.pressure_side = 'A' AND ptv.full_time_result = 'A')
                 )
                THEN 1 ELSE 0 END
        ) * 100.0 / COUNT(*), 2
    ) AS loss_rate_despite_pressure_pct
FROM pressure_team_view ptv
JOIN team t ON t.team_id = ptv.team_id
GROUP BY ptv.team_id, t.name
HAVING COUNT(*) >= 10
ORDER BY win_rate_with_pressure_pct DESC, loss_rate_despite_pressure_pct ASC;
```

This query defines attacking pressure as having more shots on target than the opponent. Matches are classified based on which team applied pressure and whether that pressure resulted in a win, draw or loss.

By grouping results by team, the query calculates win rates, draw rates and loss rates in matches where teams dominated shots on target. This highlights whether pressure reliably translates into success or whether inefficiencies are present.

Question 5

How have scoring patterns evolved across five seasons?

Question 5 SQL query:

```
USE football_analysis;
```

```sql
SELECT

   s.name AS season_name,

   COUNT(*) AS matches,

   ROUND(AVG(m.full_time_home_goals + m.full_time_away_goals), 2) AS
avg_total_goals,

   ROUND(AVG(m.full_time_home_goals), 2) AS avg_home_goals,

   ROUND(AVG(m.full_time_away_goals), 2) AS avg_away_goals,

   ROUND(

       100.0 * SUM(m.full_time_home_goals) /

       NULLIF(SUM(m.full_time_home_goals + m.full_time_away_goals), 0),

       2

   ) AS pct_goals_by_home_team,

   ROUND(AVG(m.half_time_home_goals + m.half_time_away_goals), 2) AS
avg_first_half_goals,

   ROUND(

       AVG(

           (m.full_time_home_goals + m.full_time_away_goals) -

           (m.half_time_home_goals + m.half_time_away_goals)

       ), 2

   ) AS avg_second_half_goals,

   ROUND(

       100.0 * SUM(

           (m.full_time_home_goals + m.full_time_away_goals) -

           (m.half_time_home_goals + m.half_time_away_goals)

       ) /

       NULLIF(SUM(m.full_time_home_goals + m.full_time_away_goals), 0),

       2
```

```
   ) AS pct_goals_in_second_half,

   ROUND(STDDEV_POP(m.full_time_home_goals + m.full_time_away_goals), 2) AS
goal_variability

FROM `match` m

JOIN season s ON s.season_id = m.season_id

GROUP BY s.name

ORDER BY s.name;
```

This query analyses goal scoring trends over time by computing average total goals per match, home versus away goal proportions and first half versus second half scoring patterns for each season. It also includes a measure of goal variability to indicate changes in match volatility.

The query groups matches by season and uses aggregate functions to reveal longitudinal trends that may suggest tactical or stylistic shifts in the league.

Question 6

Are bookmakers becoming more accurate at predicting match outcomes over time?

Question 6 SQL query:

```
USE football_analysis;


WITH bookmaker_predictions AS (

   SELECT

       s.name AS season_name,

       m.full_time_result AS actual_result,

       CASE

           WHEN mo.home_win_odds <= mo.draw_odds

            AND mo.home_win_odds <= mo.away_win_odds THEN 'H'

           WHEN mo.draw_odds <= mo.home_win_odds
```

```
            AND mo.draw_odds <= mo.away_win_odds THEN 'D'
            ELSE 'A'
        END AS predicted_result,
        LEAST(mo.home_win_odds, mo.draw_odds, mo.away_win_odds) AS confidence_odds
    FROM match_odds mo
    JOIN `match` m ON m.match_id = mo.match_id
    JOIN season s  ON s.season_id = m.season_id
),
classified AS (
    SELECT
        season_name,
        CASE WHEN predicted_result = actual_result THEN 1 ELSE 0 END AS
correct_prediction,
        CASE
            WHEN confidence_odds <= 1.60 THEN 'high'
            WHEN confidence_odds <= 2.20 THEN 'medium'
            ELSE 'low'
        END AS confidence_level
    FROM bookmaker_predictions
),
season_metrics AS (
    SELECT
        season_name,
        ROUND(SUM(correct_prediction) * 100.0 / COUNT(*), 2) AS
overall_prediction_accuracy_pct,
        SUM(confidence_level = 'medium') AS medium_confidence_match_count,
        ROUND(
```

```
            SUM(CASE WHEN confidence_level = 'medium' THEN correct_prediction ELSE 0
END) * 100.0 /

            NULLIF(SUM(confidence_level = 'medium'), 0),

            2

        ) AS medium_confidence_accuracy_pct

    FROM classified

    GROUP BY season_name

),

final AS (

    SELECT

        season_name,

        overall_prediction_accuracy_pct,

        ROUND(

            overall_prediction_accuracy_pct

            - LAG(overall_prediction_accuracy_pct) OVER (ORDER BY season_name),

            2

        ) AS overall_accuracy_change_pp,

        medium_confidence_match_count,

        medium_confidence_accuracy_pct,

        ROUND(

            medium_confidence_accuracy_pct

            - LAG(medium_confidence_accuracy_pct) OVER (ORDER BY season_name),

            2

        ) AS medium_confidence_accuracy_change_pp

    FROM season_metrics

)
```

```sql
SELECT
   season_name,
   overall_prediction_accuracy_pct,
   overall_accuracy_change_pp,
   medium_confidence_match_count,
   medium_confidence_accuracy_pct,
   medium_confidence_accuracy_change_pp
FROM final
ORDER BY season_name;
```

This query evaluates bookmaker prediction accuracy by comparing predicted outcomes derived from the lowest odds with actual match results. Accuracy is calculated per season and changes between seasons are measured using window functions.

Predictions are also grouped by confidence level to assess whether medium confidence odds show improved accuracy over time. This allows analysis of both overall prediction quality and temporal trends.

Question 7

Is aggressive play linked to success or does it harm performance?

This research question is addressed using two queries.

Question 7A - Team-level aggression vs performance

Question 7A SQL query:

```sql
USE football_analysis;

WITH team_match AS (
   SELECT
       m.season_id,
```

```sql
       m.match_id,

       m.home_team_id AS team_id,

       m.home_fouls         AS fouls,

       m.home_yellow_cards  AS yellows,

       m.home_red_cards     AS reds,

       (m.home_fouls + 3*m.home_yellow_cards + 10*m.home_red_cards) AS
aggression_index,

       CASE

           WHEN m.full_time_result = 'H' THEN 3

           WHEN m.full_time_result = 'D' THEN 1

           ELSE 0

       END AS points,

       (m.full_time_home_goals - m.full_time_away_goals) AS goal_diff

   FROM `match` m



   UNION ALL



   SELECT

       m.season_id,

       m.match_id,

       m.away_team_id AS team_id,

       m.away_fouls         AS fouls,

       m.away_yellow_cards  AS yellows,

       m.away_red_cards     AS reds,

       (m.away_fouls + 3*m.away_yellow_cards + 10*m.away_red_cards) AS
aggression_index,

       CASE
```

```
            WHEN m.full_time_result = 'A' THEN 3

            WHEN m.full_time_result = 'D' THEN 1

            ELSE 0

        END AS points,

        (m.full_time_away_goals - m.full_time_home_goals) AS goal_diff

    FROM `match` m

),

team_summary AS (

    SELECT

        t.team_id,

        t.name AS team_name,

        COUNT(*) AS matches,

        ROUND(AVG(points), 3) AS avg_points_per_match,

        ROUND(AVG(goal_diff), 3) AS avg_goal_diff_per_match,

        ROUND(AVG(fouls), 2)   AS avg_fouls_per_match,

        ROUND(AVG(yellows), 2) AS avg_yellows_per_match,

        ROUND(AVG(reds), 3)    AS avg_reds_per_match,

        ROUND(AVG(aggression_index), 2) AS avg_aggression_index

    FROM team_match tm

    JOIN team t ON t.team_id = tm.team_id

    GROUP BY t.team_id, t.name

)

SELECT

    team_name,

    matches,

    avg_aggression_index,
```

```
   avg_fouls_per_match,

   avg_yellows_per_match,

   avg_reds_per_match,

   avg_points_per_match,

   avg_goal_diff_per_match

FROM team_summary

WHERE matches >= 190

ORDER BY avg_aggression_index DESC, avg_points_per_match DESC;
```

This query analyses aggression at the team level by constructing an aggression index based on fouls, yellow cards and red cards. Average aggression is compared with average points and goal difference per match to identify correlations between discipline and performance.

Question 7B - Outcome rates by aggression band

Question 7B SQL query:

```
USE football_analysis;

WITH team_match AS (

   SELECT

       m.season_id,

       m.match_id,

       m.home_team_id AS team_id,

       (m.home_fouls + 3*m.home_yellow_cards + 10*m.home_red_cards) AS
aggression_index,

       CASE

           WHEN m.full_time_result = 'H' THEN 3

           WHEN m.full_time_result = 'D' THEN 1

           ELSE 0

       END AS points,
```

```
       (m.full_time_home_goals - m.full_time_away_goals) AS goal_diff
   FROM `match` m

   UNION ALL

   SELECT
       m.season_id,
       m.match_id,
       m.away_team_id AS team_id,
       (m.away_fouls + 3*m.away_yellow_cards + 10*m.away_red_cards) AS
aggression_index,
       CASE
           WHEN m.full_time_result = 'A' THEN 3
           WHEN m.full_time_result = 'D' THEN 1
           ELSE 0
       END AS points,
       (m.full_time_away_goals - m.full_time_home_goals) AS goal_diff
   FROM `match` m
),
banded AS (
   SELECT
       *,
       NTILE(3) OVER (ORDER BY aggression_index) AS aggression_tercile
   FROM team_match
)
SELECT
```

```
   CASE aggression_tercile

       WHEN 1 THEN 'Low aggression'

       WHEN 2 THEN 'Medium aggression'

       WHEN 3 THEN 'High aggression'

   END AS aggression_band,

   COUNT(*) AS team_match_rows,

   ROUND(AVG(aggression_index), 2) AS avg_aggression_index,

   ROUND(AVG(points), 3) AS avg_points_per_match,

   ROUND(AVG(goal_diff), 3) AS avg_goal_diff_per_match,

   ROUND(100.0 * AVG(points = 3), 2) AS win_rate_pct,

   ROUND(100.0 * AVG(points = 1), 2) AS draw_rate_pct,

   ROUND(100.0 * AVG(points = 0), 2) AS loss_rate_pct

FROM banded

GROUP BY aggression_tercile

ORDER BY aggression_tercile;
```

This query takes a league wide perspective by grouping team match performances into low, medium and high aggression bands. Match outcomes are compared across these bands to assess whether aggressive play is associated with higher success or poorer results.

Question 8

Do certain referees consistently award more cards and does this influence match outcomes?

This research question is addressed using two queries.

Question 8A - Referee card tendencies (yellow/red split)

Question 8A SQL query:

```
WITH ref_match AS (

 SELECT
```

```
   m.referee_id,
   m.season_id,
   (COALESCE(m.home_yellow_cards,0) + COALESCE(m.away_yellow_cards,0)) AS
yellows_in_match,
   (COALESCE(m.home_red_cards,0)    + COALESCE(m.away_red_cards,0))    AS
reds_in_match
 FROM `match` m
 WHERE m.referee_id IS NOT NULL
),

ref_season AS (
 SELECT
   referee_id,
   season_id,
   COUNT(*) AS matches_officiated,
   AVG(yellows_in_match) AS avg_yellows_per_match,
   AVG(reds_in_match)    AS avg_reds_per_match
 FROM ref_match
 GROUP BY referee_id, season_id
),

ref_overall AS (
 SELECT
   referee_id,
   SUM(matches_officiated) AS total_matches,
   COUNT(DISTINCT season_id) AS seasons_covered,
```

```
   AVG(avg_yellows_per_match) AS avg_yellows,
   STDDEV_POP(avg_yellows_per_match) AS yellows_stddev,

   AVG(avg_reds_per_match) AS avg_reds,
   STDDEV_POP(avg_reds_per_match) AS reds_stddev
 FROM ref_season
 GROUP BY referee_id
),

ref_extremes AS (
 SELECT
   referee_id,
   MIN(yellows_in_match) AS min_yellows_in_a_match,
   MAX(yellows_in_match) AS max_yellows_in_a_match,
   MIN(reds_in_match)    AS min_reds_in_a_match,
   MAX(reds_in_match)    AS max_reds_in_a_match
 FROM ref_match
 GROUP BY referee_id
)

SELECT
 r.name AS referee_name,
 ro.total_matches,
 ro.seasons_covered,

 ROUND(ro.avg_yellows, 3) AS avg_yellow_cards_per_match,
```

```
 ROUND(ro.yellows_stddev, 3) AS yellow_consistency_stddev,

 re.min_yellows_in_a_match,

 re.max_yellows_in_a_match,



 ROUND(ro.avg_reds, 3) AS avg_red_cards_per_match,

 ROUND(ro.reds_stddev, 3) AS red_consistency_stddev,

 re.min_reds_in_a_match,

 re.max_reds_in_a_match



FROM ref_overall ro

JOIN ref_extremes re ON re.referee_id = ro.referee_id

JOIN referee r ON r.referee_id = ro.referee_id

ORDER BY avg_yellow_cards_per_match DESC;
```

This query analyses referee behaviour by calculating average yellow and red cards per match across seasons. Measures of consistency and extremes are included to identify referees with distinctive disciplinary patterns.

Question 8B - Outcome rates by referee card intensity (low/medium/high)

Question 8B SQL query:

```
WITH ref_avg AS (

 SELECT

   m.referee_id,

   AVG(

     (COALESCE(m.home_yellow_cards,0) + COALESCE(m.away_yellow_cards,0)) +

     2*(COALESCE(m.home_red_cards,0)    + COALESCE(m.away_red_cards,0))

   ) AS ref_avg_weighted_cards,

   COUNT(*) AS ref_match_count
```

```
 FROM `match` m
 WHERE m.referee_id IS NOT NULL
 GROUP BY m.referee_id
 HAVING COUNT(*) >= 30
),
ranked AS (
 SELECT
   referee_id,
   ref_avg_weighted_cards,
   ref_match_count,
   NTILE(3) OVER (ORDER BY ref_avg_weighted_cards) AS card_intensity_tile
 FROM ref_avg
),
match_labeled AS (
 SELECT
   m.match_id,
   m.full_time_result,
   r.card_intensity_tile
 FROM `match` m
 JOIN ranked r ON r.referee_id = m.referee_id
)
SELECT
 CASE card_intensity_tile
   WHEN 1 THEN 'LOW_CARD_REFEREES'
   WHEN 2 THEN 'MEDIUM_CARD_REFEREES'
   WHEN 3 THEN 'HIGH_CARD_REFEREES'
```

```
 END AS referee_group,

 COUNT(*) AS matches,

 ROUND(100 * AVG(full_time_result = 'H'), 2) AS home_win_pct,

 ROUND(100 * AVG(full_time_result = 'D'), 2) AS draw_pct,

 ROUND(100 * AVG(full_time_result = 'A'), 2) AS away_win_pct

FROM match_labeled

GROUP BY card_intensity_tile

ORDER BY card_intensity_tile;
```

This query groups referees into low, medium and high card intensity categories and examines match outcomes under each group. This allows assessment of whether officiating style is associated with systematic differences in results.

Overall Assessment

All research questions were successfully answered using SQL queries executed against the database. Each query required joining multiple entities and performing aggregation across large datasets, demonstrating the effectiveness of the relational design. No research questions were unanswerable using the available data, although some analyses operate on subsets of matches due to missing bookmaker odds or referee information, as discussed in Section 3.3.

Stage 4 - Web Application

4.1 Application Overview

A dynamic web application was developed using Node.js and the Express framework to provide an interactive interface to the Football Analysis Database. The application serves as a practical demonstration of how a normalised relational database can be queried and presented to users in a structured and meaningful way.

The primary purpose of the application is to expose the results of the analytical SQL queries developed in Stage 3 without requiring users to write SQL directly. Instead, users interact with a clean dashboard that presents research questions and dynamically renders the corresponding results.

Server-side rendering is implemented using EJS (Embedded JavaScript) templates, allowing query results to be transformed into structured HTML views at runtime. The interface follows a modern, minimal dashboard-style layout with a consistent dark theme to maximise readability of charts and tabular data. The design intentionally prioritises data clarity over visual complexity, ensuring that analytical outputs remain the focal point of the application.

4.2 Database Interaction and Application Architecture

The application was designed using a modular architecture that separates database access, routing logic and presentation layers. This structure improves maintainability and reflects best practice in database-driven application development.

Database Connectivity

Database access is managed using the mysql2 library with a connection pool, rather than individual connections per request. This ensures efficient reuse of database connections and allows multiple concurrent queries to be executed without unnecessary overhead. This is particularly important when queries involve complex operations such as Common Table Expressions (CTEs) or window functions.

Query Abstraction

All SQL queries used to answer the research questions are stored in a dedicated query module rather than being embedded directly in route handlers. This abstraction:

- Keeps routing logic concise and readable
- Reduces the risk of introducing SQL errors when modifying routes
- Allows complex analytical queries to be maintained and optimised independently of the web interface

This design decision directly reflects the separation of concerns principle and mirrors real-world database-backed application architectures.

Asynchronous Query Execution

The application uses asynchronous async/await patterns when executing database queries. This ensures that long-running analytical queries do not block the event loop, allowing the application to remain responsive while processing multi-season aggregations and joins across large tables.

4.3 Data Visualisation and Dynamic Rendering

A key feature of the application is its ability to adaptively present query results based on the structure and purpose of the data returned.

Dynamic Rendering Logic

Query results are passed to a shared rendering template that automatically determines how the data should be displayed:

- Time-series results (e.g. season-based trends) are rendered as line charts
- Categorical comparisons (e.g. referee intensity bands) are rendered as bar charts
- Relationship and efficiency analyses (e.g. pressure vs outcomes) are rendered as scatter or bubble charts
- Where visualisation is not appropriate, results are displayed as structured tables

This adaptive logic avoids hard-coding chart types per question and allows new queries to be added with minimal frontend changes.

Client-Side Visualisation

The application integrates Chart.js to generate interactive charts in the browser. These charts provide:

- Tooltips for detailed values
- Clear axis labelling and legends
- Visual comparison across teams, seasons, or referee groups

This visual layer complements the SQL analysis by making patterns such as performance trends, inefficiencies and variability immediately interpretable.

Multi-Part Question Handling

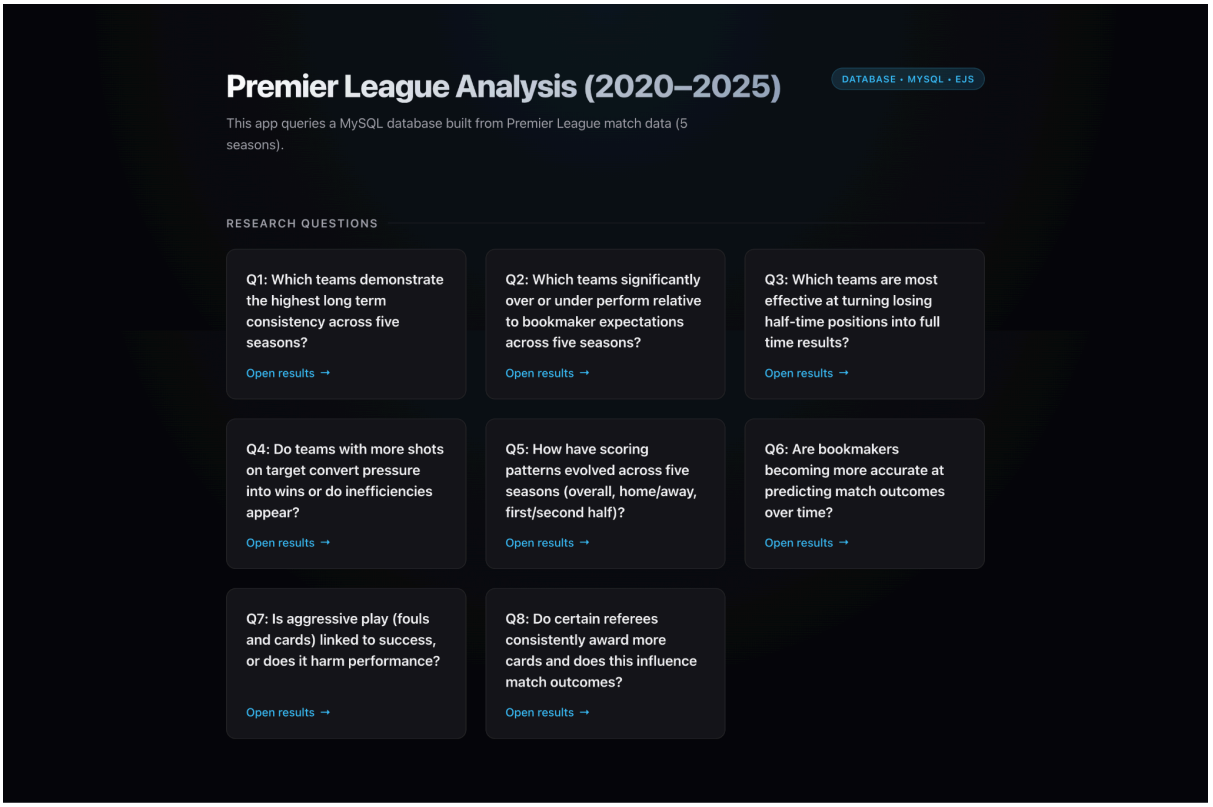
For more complex research questions (Question 7 and Question 8), the application supports multi-part outputs:

- **Part A** focuses on aggregated or visual insights
- **Part B** presents detailed tabular breakdowns for deeper inspection

This approach ensures that visualisation enhances analysis rather than oversimplifying it.

4.4 Screenshots

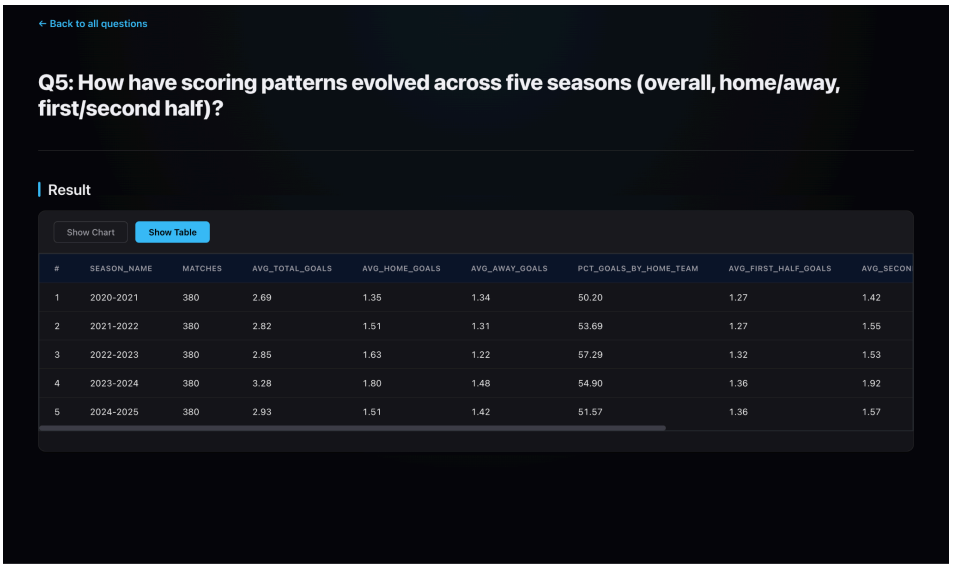
Figure 4.1 - Main Application Dashboard



Landing page displaying available research questions in a structured dashboard layout.

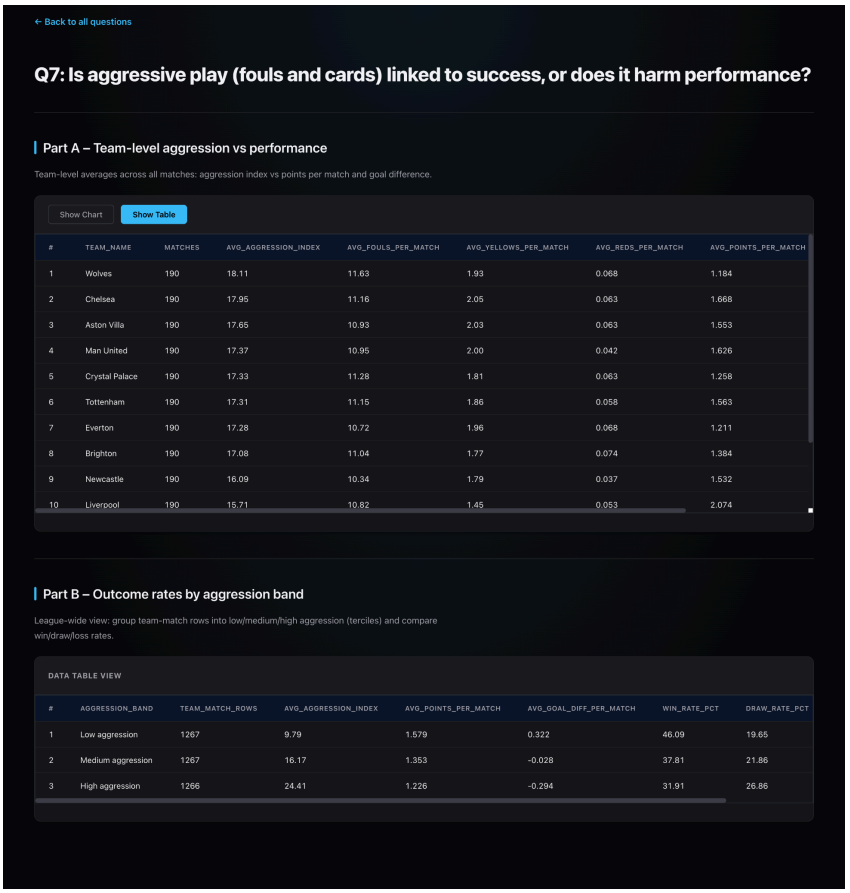
Figure 4.2 & 4.3 - Tabular Results View

Question 5 table:



#	SEASON_NAME	MATCHES	AVG_TOTAL_GOALS	AVG_HOME_GOALS	AVG_AWAY_GOALS	PCT_GOALS_BY_HOME_TEAM	AVG_FIRST_HALF_GOALS	AVG_SECON
1	2020-2021	380	2.69	1.35	1.34	50.20	1.27	1.42
2	2021-2022	380	2.82	1.51	1.31	53.69	1.27	1.55
3	2022-2023	380	2.85	1.63	1.22	57.29	1.32	1.53
4	2023-2024	380	3.28	1.80	1.48	54.90	1.36	1.92
5	2024-2025	380	2.93	1.51	1.42	51.57	1.36	1.57

Question 7A & 7B tables:



#	TEAM_NAME	MATCHES	AVG_AGGRESSION_INDEX	AVG_FOULS_PER_MATCH	AVG_YELLOWS_PER_MATCH	AVG_REDS_PER_MATCH	AVG_POINTS_PER_MATCH
1	Wolves	190	18.11	11.63	1.93	0.068	1.184
2	Chelsea	190	17.95	11.16	2.05	0.063	1.668
3	Aston Villa	190	17.65	10.93	2.03	0.063	1.553
4	Man United	190	17.37	10.95	2.00	0.042	1.626
5	Crystal Palace	190	17.33	11.28	1.81	0.063	1.258
6	Tottenham	190	17.31	11.15	1.86	0.058	1.563
7	Everton	190	17.28	10.72	1.96	0.068	1.211
8	Brighton	190	17.08	11.04	1.77	0.074	1.384
9	Newcastle	190	16.09	10.34	1.79	0.037	1.532
10	Liverpool	190	15.71	10.82	1.45	0.053	2.074

#	AGGRESSION_BAND	TEAM_MATCH_ROWS	AVG_AGGRESSION_INDEX	AVG_POINTS_PER_MATCH	AVG_GOAL_DIFF_PER_MATCH	WIN_RATE_PCT	DRAW_RATE_PCT
1	Low aggression	1267	9.79	1.579	0.322	46.09	19.65
2	Medium aggression	1267	16.17	1.353	-0.028	37.81	21.86
3	High aggression	1266	24.41	1.226	-0.294	31.91	26.86

Detailed table output for questions requiring granular inspection of records or grouped statistics.

Figure 4.4 - 4.6 – Visual Analytics View

Question 3 table and chart:

#	TEAM_NAME	LOSING_HT_GAMES	FULL_COMEBACKS_TO_WIN	COMEBACKS_TO_DRAW	COMEBACKS_TO_NON_LOSS	WIN_COMEBACK_PCT	DRAW_COMEBAC
1	Man City	27	9	6	15	33.33	22.22
2	Man United	52	11	7	18	21.15	13.46
3	Liverpool	37	7	16	23	18.92	43.24
4	Tottenham	54	8	6	14	14.81	11.11
5	Bournemouth	44	6	4	10	13.64	9.09
6	Chelsea	37	5	5	10	13.51	13.51
7	Fulham	49	6	6	12	12.24	12.24
8	Newcastle	50	6	10	16	12.00	20.00
9	Brighton	60	7	15	22	11.67	25.00
10	Arsenal	35	4	8	12	11.43	22.86

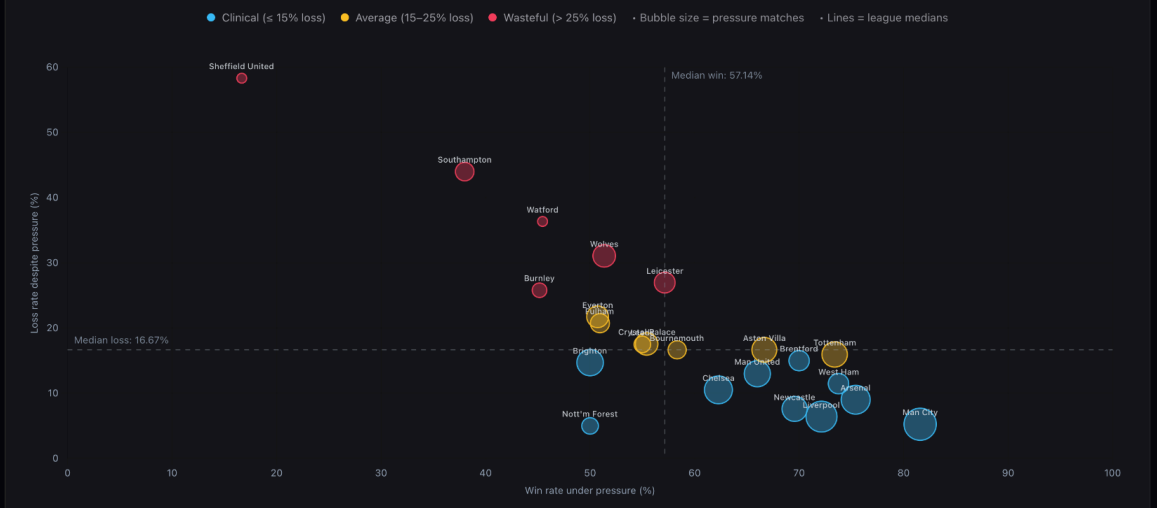
Question 4 table and chart:

← Back to all questions

Q4: Do teams with more shots on target convert pressure into wins or do inefficiencies appear?

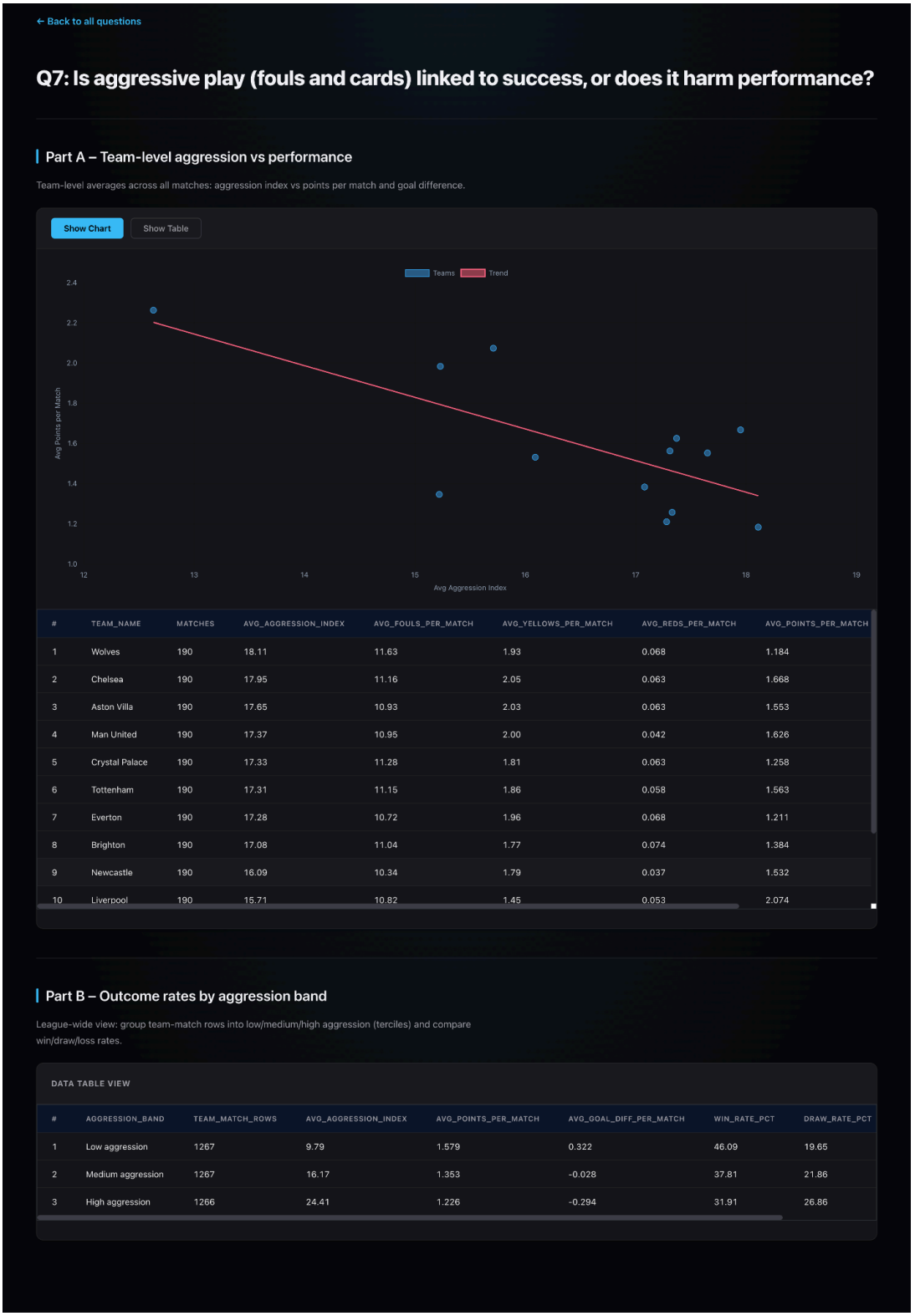
Result

Show Chart | Show Table



#	TEAM_NAME	MATCHES_WITH_PRESSURE	WINS_WITH_PRESSURE	DRAWS_WITH_PRESSURE	LOSSES_DESPITE_PRESSURE	WIN_RATE_WITH_PRESSURE_PCT
1	Man City	152	124	20	8	81.58
2	Arsenal	122	92	19	11	75.41
3	West Ham	61	45	9	7	73.77
4	Tottenham	94	69	10	15	73.40
5	Liverpool	140	101	30	9	72.14
6	Brentford	60	42	9	9	70.00
7	Newcastle	92	64	21	7	69.57
8	Aston Villa	90	60	15	15	66.67
9	Man United	100	66	21	13	66.00
10	Chelsea	114	71	31	12	62.28

Question 7 table and chart:



Interactive charts generated from aggregated SQL results, demonstrating trends across seasons or teams.

4.5 Evaluation Against Project Goals

The web application successfully meets the objectives defined at the start of the project:

- **Effective Database Interaction**
  All research questions defined in Stage 1 are executed directly against the normalised MySQL schema, demonstrating correct use of joins, constraints and aggregation.
- **Improved Accessibility**
  Users can explore complex analytical results without writing SQL, while still benefiting from advanced database queries developed in Stage 3.
- **Analytical Insight**
  The combination of SQL aggregation and visualisation exposes patterns such as long-term team consistency, referee behaviour and bookmaker accuracy-that are not immediately visible in raw datasets.
- **Performance and Usability**
  Query execution is efficient and the interface remains responsive even when working with five seasons of match data.

Referencing and Academic Practice

Libraries and Frameworks Used

- Express.js - Web application framework
- MySQL2 - Database connectivity
- EJS - Server-side templating
- Chart.js - Client-side data visualisation

Data Source

Premier League match data (2020-2025), sourced from an open football statistics provider and transformed into a custom normalised relational schema. All original code written for this project is clearly structured and commented. External libraries are used appropriately and acknowledged.

Enhancements and Advanced Features

Beyond the core requirements, the application demonstrates additional technical depth:

- **Advanced SQL Integration**
  The interface visualises results derived from CTEs and window functions, reinforcing the analytical strength of the database layer.
- **Robust UI Behaviour**
  Defensive frontend logic prevents runtime failures when datasets are sparse or incomplete.
- **Scalable Design**
  The modular architecture allows new research questions or datasets to be added without restructuring the application.