Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática

Phase 1 Report

Martim Emauz 58668 Miguel Martins 58661 Rúben Torres 62531

Bases de Dados Avançadas

Mestrado em Engenharia Informática

2024

1 Project Contribution

Martim Emauz (33%) - Participated in the research process for the dataset, Created new CSV data to complement the given dataset. Created the queries.

Miguel Martins (33%) - Participated in the research process for the dataset, Selected the CSV files to be used and Modeled the relational schema. Created the mySQL tables.

Rúben Torres (33%) - Participated in the research process for the dataset, created the MongoDB collections.

2 Dataset

We chose a dataset offering an in-depth statistical analysis of the 2023/24 Premier League Football season, containing information from players and teams throughout the course of the season.
The dataset includes more than 50 CSV files, of which we used 20, with between 20 to 449 lines entries each. All of the data it contains is provided by *FotMob*.
The dataset is available in the following web page: `https://www.kaggle.com/datasets/whisperingkahuna/premier-league-2324-team-and-player-insights` Additional data was created by us, regarding coaches, team and player information.

3 Relational Schema

Based on the dataset, we created tables representing the players and teams, and additionally added a table for the coaches. The relationships between tables also meant we created the Team_Coach and Team_Player tables. to represent the data, we created two tables for statistics, representing the player and team statistics available in the CSV files.
The following code was used to create the tables in MySQL:

```
# create tables
# team
mycursor.execute("""
CREATE TABLE IF NOT EXISTS team (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    position_league INT,
    country VARCHAR(3),
    city VARCHAR(255),
    region VARCHAR(255),
    matches INT
);
""")

# player
mycursor.execute("""
CREATE TABLE IF NOT EXISTS player (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    country VARCHAR(3),
    strong_foot BOOLEAN,
    matches INT,
    minutes INT
);
""")

# coach
mycursor.execute("""
CREATE TABLE IF NOT EXISTS coach (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    age INT,
    country VARCHAR(3),
    experience_years INT,
    titles INT
);
""")

# team_player
mycursor.execute("""
CREATE TABLE IF NOT EXISTS team_player (
    id INT AUTO_INCREMENT PRIMARY KEY,
    team_id INT,
    player_id INT,
        FOREIGN KEY (team_id) REFERENCES team(id),
        FOREIGN KEY (player_id) REFERENCES player(id)
);
""")

# team_coach
mycursor.execute("""
CREATE TABLE IF NOT EXISTS team_coach (
    id INT AUTO_INCREMENT PRIMARY KEY,
    team_id INT,
    coach_id INT,
        FOREIGN KEY (team_id) REFERENCES team(id),
        FOREIGN KEY (coach_id) REFERENCES coach(id)
);
""")

# team_stats
mycursor.execute("""
CREATE TABLE IF NOT EXISTS team_stats (
    id INT AUTO_INCREMENT PRIMARY KEY,
    team_id INT,
    cross_success_pct FLOAT,
    accurate_crosses_per_90 FLOAT,
    corners_taken FLOAT,
    team_rating FLOAT,
    touches_in_opp_box FLOAT,
    possession_pct FLOAT,
       FOREIGN KEY (team_id) REFERENCES team(id)
);
""")

# player_stats
mycursor.execute("""
CREATE TABLE IF NOT EXISTS player_stats (
    id INT AUTO_INCREMENT PRIMARY KEY,

    -- Player identification
    player_id INT,
    player_rating FLOAT,
    player_of_match FLOAT,

    -- Goal-related statistics
    goals FLOAT,
    expected_goals FLOAT,
    shot_conversion_rate FLOAT,
    penalties_scored FLOAT,

    -- Assist statistics
    assists FLOAT,
    expected_assists FLOAT,
    secondary_assists FLOAT,
    big_chances_created FLOAT,
    big_chances_missed FLOAT,

    -- Passing metrics
    pass_success_pct FLOAT,
    accurate_passes_90 FLOAT,
    accurate_long_balls_90 FLOAT,
    successful_long_balls_pct FLOAT,

    -- Defensive statistics
    clean_sheets FLOAT,
    goals_conceded FLOAT,
    total_clearances FLOAT,
    total_interceptions FLOAT,

    -- Dribbling metrics
    successful_dribbles_90 FLOAT,
    dribble_success_rate FLOAT,

    -- Foul statistics
    yellow_cards FLOAT,
    red_cards FLOAT,
    fouls_per_90 FLOAT,
       FOREIGN KEY (player_id) REFERENCES player(id)
);
""")
```

The following code was used to create the collections in MongoDB:

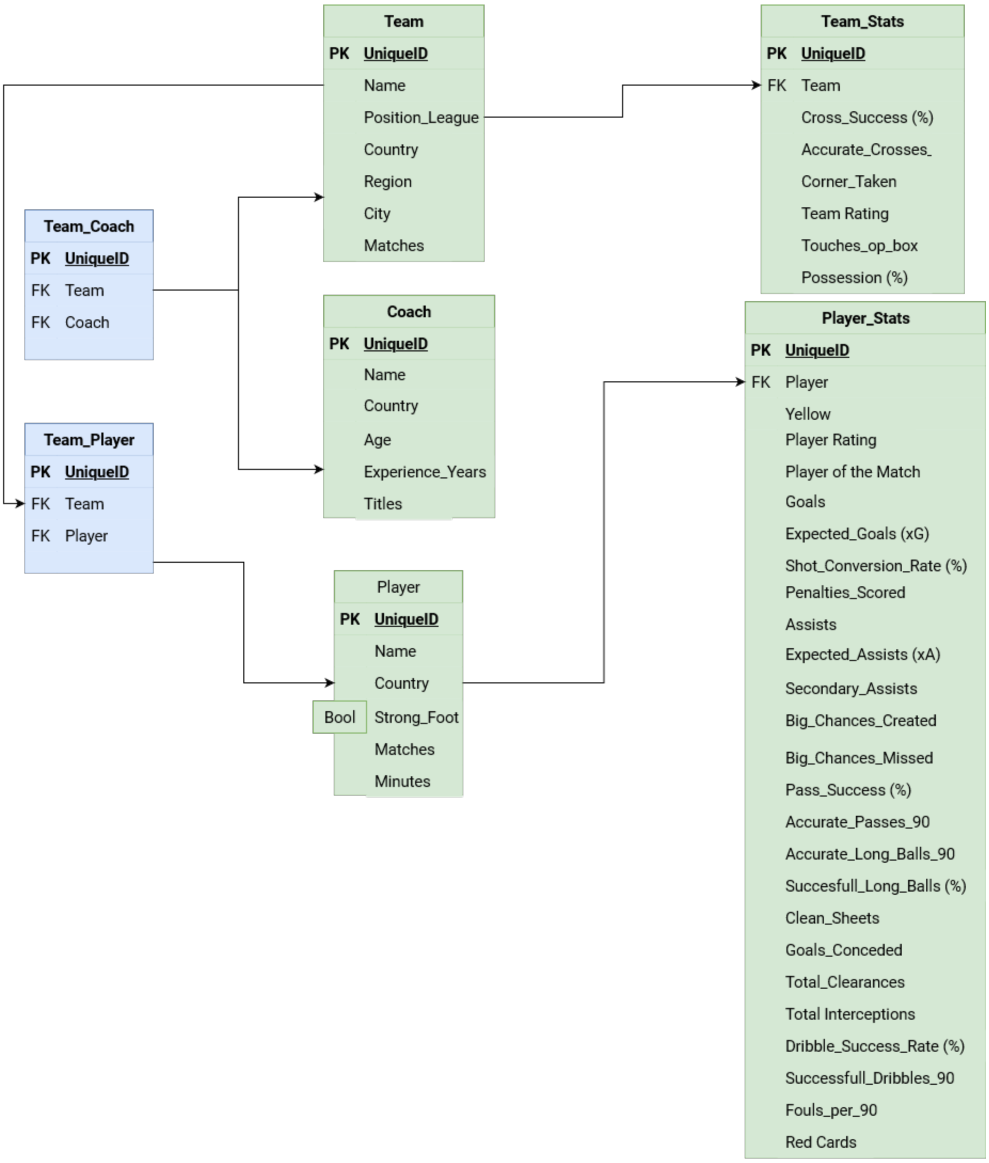
```
# team (key), name, age, country, experience_years, titles;
df_coaches = df_coaches.rename(
    columns={
        "Club": "team",
        "Name": "name",
        "Country": "country",
        "Age": "age",
        "Experience": "experience_years",
        "Titles": "titles",
    }
)
columns = ["team", "name", "age", "country", "experience_years", "titles"]
df_coaches = df_coaches[columns]
coach = db["coach"] # MySQL == coach + team_coach
coach_result = coach.insert_many(df_coaches.to_dict("records"))
print(f"Added {len(coach_result.inserted_ids)} coaches!")

# team (key), name, country, strong_foot, matches, minutes;
df_players = df_players.rename(
    columns={
        "Team": "team",
        "Player": "name",
        "Country": "country",
        "Matches": "matches",
        "Minutes": "minutes",
    }
)
df_players["strong_foot"] = [
    random.choice([True, False]) for _ in range(len(df_players))
]
columns = ["team", "name", "country", "strong_foot", "matches", "minutes"]
df_players = df_players[columns]
player = db["player"] # MySQL == player + team_player
player_result = player.insert_many(df_players.to_dict("records"))
print(f"Added {len(player_result.inserted_ids)} players!")

# name (key), position_league, country, city, region, matches;
df_teams = df_teams.rename(
    columns={
        "idx": "position_league",
        "played": "matches",
    }
)
df_teams["country"] = "ENG"
df_teams["city"] = df_teams["name"].map(
    lambda team: teams_data.get(team, {}).get("city", "Unknown")
)
df_teams["region"] = df_teams["name"].map(
    lambda team: teams_data.get(team, {}).get("region", "Unknown")
)
columns = ["name", "position_league", "country", "city", "region", "matches"]
df_teams = df_teams[columns]
team = db["team"]
team_result = team.insert_many(df_teams.to_dict("records"))
print(f"Added {len(team_result.inserted_ids)} teams!")
```

Additionally, information was added on top of what was in the dataset. The Coach table contains information manually inserted regarding the league's coaches. In the Team table, the City and Region were also created by us, and in the Player table, the Strong Foot was randomly assorted as a boolean variable. Values set as "None" in the dataset were defined as 0 in the database.

The relational schema diagram illustrates the tables and their relationships. Tables colored in green were included on both MySQL and MongoDB, while tables in blue were only included in MySQL.



4 Additional Features

The dataset was complemented with additional data and data types, as mentioned above, for a more complete data pool. No other additional features or elements were developed.

5 Undone Features

No features were left undone.

6 Known Errors

No errors or bugs were found.