



THE COPPERBELT UNIVERSITY
SCHOOL OF INFORMATION COMMUNICATION AND TECHNOLOGY

DATABASE SYSTEMS
CS 235 ASSIGNMENT 1

FAZ DATABASE PROJECT

GROUP 19

MEMBERS

NO	SIN	STUDENT NAME	PROGRAM
1.	22109751	ZULU CLIVE	CE
2.	22175863	JOSEPH CHIRWA	CS
3.	22112527	SAKUWAHA HENRY	CE
4.	22106426	KUMWENDA GRACE	CS
5.	22110303	LEWIS SIKANYIKA	CS
6.	21170836	DAVID MULENGA	IS
7.	22998278	SHEKINAH B MULENGA	CS
8.	22103729	NDHLOVU SILAS B	CS

Table of Contents	Page
1. PROJECT OVERVIEW	3
2. INTRODUCTION	3
3. OBJECTIVES	3
4. ER DIAGRAM	3
5. IMPLEMENTATION	5
5.1. TECHNOLOGY STACK	5
5.2. DATABASE SETUP	5
5.3. SCHEMA DESIGN	5
5.4 GENERATING REPORTS	9
REFERENCES	11

1. PROJECT OVERVIEW

The project involves designing a DBMS to efficiently model and manage soccer teams, players, matches and referees. The system will track team details, player information, match schedules and results, player participation and performance in matches. The key entities include teams, players, matches and referees, with relationships established between them to ensure accurate data representation. This DBMS aims to streamline operations, reduce errors and enhance data accessibility and scalability, ultimately improving the management and development of soccer activities within the association.

2. INTRODUCTION

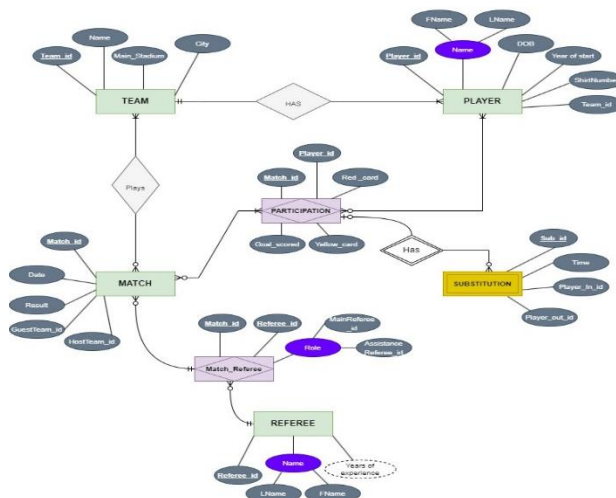
The Football Management System (FMS) is a comprehensive solution designed to stream line the management of football teams, players, matches and referees for the Football Association of Zambia (FAZ). It uses modern web technologies and robust database management practices to provide a user-friendly interface as well as a centralized platform that ensures comprehensive data management.

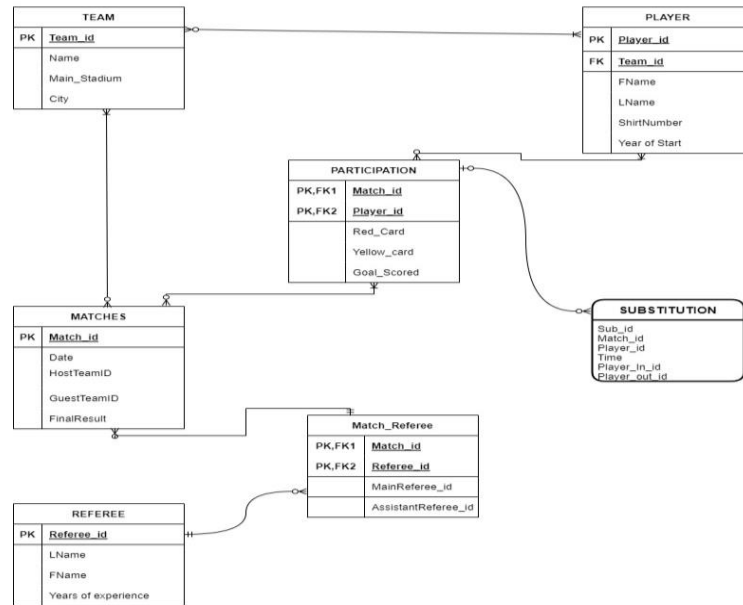
3. OBJECTIVES

The primary objectives of the FMS are:

- Centralized Data Management: Store comprehensive information about teams, players, matches and referees in a single unified database
- Data Integrity: Ensure data consistency and accuracy through well-defined relational database constraints and normalization.
- Operational Efficiency: Simplify the tracking and management of football operations, reducing administrative workload.
- Secure Access Control: Implement authentication mechanisms to ensure that authorized personnel (administrators) can access specific functionality of the system.

4. ER DIAGRAM





Relationships:

- Team - Player: One team can have many players (one-to-many). A team can have multiple players, but a player belongs to only one team.
- Team - Matches (HostTeamID): One team can host many matches (one-to-many). A team can host multiple matches, but a match has only one host team.
- Team - Matches (GuestTeamID): One team can play many matches as a guest (one-to-many). A team can play multiple matches as a guest, but a match has only one guest team.
- Player - Participation: One player can participate in many matches (one-to-many). A player can participate in multiple matches, but a participation record is associated with only one player.
- Matches - Participation: One match can have many participations (one-to-many). A match can have multiple players participating, but a participation record is associated with only one match.
- Player - Substitution: One player can be involved in many substitutions (one-to-many). A player can be substituted in or out multiple times, but a substitution record is associated with only one player.
- Matches - Substitution: One match can have many substitutions (one-to-many). A match can have multiple substitutions, but a substitution record is associated with only one match.
- Referee - MatchReferee: One referee can officiate many matches (one-to-many). A referee can officiate multiple matches, but a match referee assignment is associated with only one referee.
- Matches - MatchReferee: One match can have many referees (one-to-many). A match can have multiple referees (e.g., main referee, assistant referees), but a match referee assignment is associated with only one match.

5. IMPLEMENTATION

5.1. TECHNOLOGY STACK

The FMS project utilizes the following technologies:

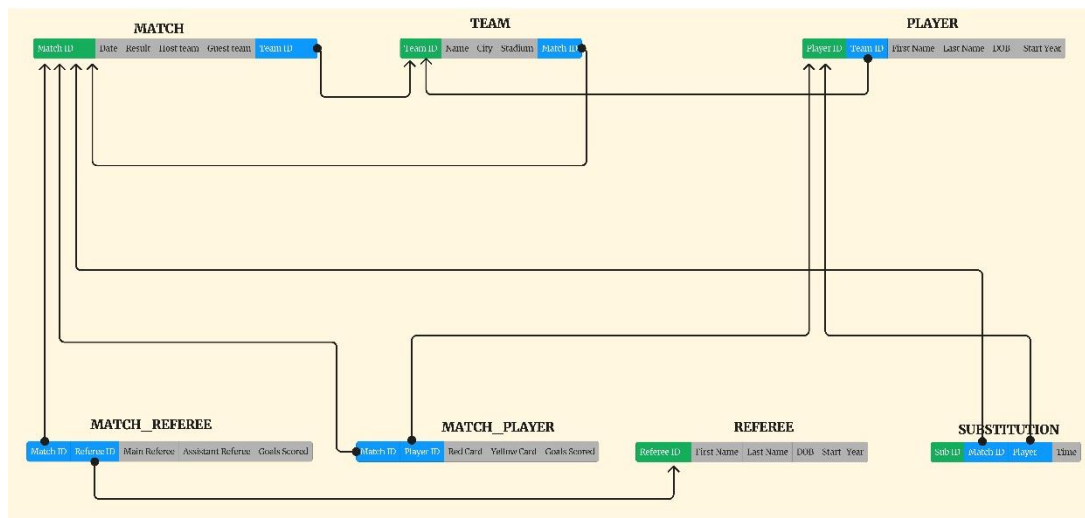
- Front-end: HTML, CSS, JavaScript
- Back-end: Node.js (runtime environment for server-side operations)
- Database Management: SQLite (for development and testing), MySQL (for production)
- Tools: MySQL workbench (for DB design and management), Visual Studio Code (for code editing and development)
- Generation of Reports: SSRs

5.2. DATABASE SETUP

```
-- Create database
CREATE DATABASE faz_db;

-- Use the created database
USE faz_db;
```

5.3. SCHEMA DESIGN



Team Table

Stores information about each team, including its unique ID, name, main stadium, and city.

```
CREATE TABLE Team (  
    TeamID INT PRIMARY KEY,  
    TeamName VARCHAR(50) NOT NULL,  
    MainStadium VARCHAR(50),  
    City VARCHAR(50)  
);
```

Player Table

Contains details of players, including their unique ID, name, date of birth, start year, and shirt number. Each player is associated with a team through the TeamID foreign key.

```
CREATE TABLE Player (  
    PlayerID INT PRIMARY KEY,  
    FName VARCHAR(50) NOT NULL,  
    LName VARCHAR(50) NOT NULL,  
    DoB DATE NOT NULL,  
    StartYear INT NOT NULL,  
    ShirtNumber INT NOT NULL,  
    TeamID INT,  
    FOREIGN KEY (TeamID) REFERENCES Team(TeamID)  
);
```

Matches Table

Records match details, including the unique match ID, host and guest teams, date of the match, and the final result. Foreign keys link to the Team table for both the host and guest teams.

```
CREATE TABLE Matches (  
    MatchID INT AUTO_INCREMENT PRIMARY KEY,  
    HostTeamID INT,  
    GuestTeamID INT,
```

```

        Date DATE NOT NULL,

        FinalResult VARCHAR(255),

        FOREIGN KEY (HostTeamID) REFERENCES Team(TeamID),

        FOREIGN KEY (GuestTeamID) REFERENCES Team(TeamID)

    );

```

Referee Table

Stores information about referees, including their unique ID, name, date of birth, and years of experience.

```

CREATE TABLE Referee (

    RefereeID INT PRIMARY KEY,

    FName VARCHAR(50) NOT NULL,

    LName VARCHAR(50) NOT NULL,

    DoB DATE NOT NULL,

    YearsOfExperience INT NOT NULL

);

```

Participation Table:

Captures player participation in matches, including goals scored, yellow cards, and red cards. The primary key is a composite of MatchID and PlayerID, linking to the Matches and Player tables.

```

CREATE TABLE Participation (

    MatchID INT,

    PlayerID INT,

    GoalsScored INT,

    YellowCard BOOLEAN,

    RedCard BOOLEAN,

    PRIMARY KEY (MatchID, PlayerID),

    FOREIGN KEY (MatchID) REFERENCES Matches(MatchID),

    FOREIGN KEY (PlayerID) REFERENCES Player(PlayerID)

);

```

Substitution Table:

Records player substitutions during matches, including the match ID, the player coming out, the player going in, and the time of substitution. The primary key is a composite of MatchID, OutPlayerID, and SubstitutionTime, with foreign keys linking to the Participation and Player tables.

```
CREATE TABLE Substitution (  
    MatchID INT,  
    OutPlayerID INT,  
    InPlayerID INT,  
    SubstitutionTime TIME,  
    PRIMARY KEY (MatchID, OutPlayerID, SubstitutionTime),  
    FOREIGN KEY (MatchID, OutPlayerID) REFERENCES  
    Participation(MatchID, PlayerID),  
    FOREIGN KEY (InPlayerID) REFERENCES Player(PlayerID)  
);
```

MatchReferee Table:

Associates referees with matches and specifies their role (main referee or assistant referee). The primary key is a composite of MatchID and RefereeID, with foreign keys linking to the Matches and Referee tables.

```
CREATE TABLE MatchReferee (  
    MatchID INT,  
    RefereeID INT,  
    RefereeRole VARCHAR(50),  
    PRIMARY KEY (MatchID, RefereeID),  
    FOREIGN KEY (MatchID) REFERENCES Matches(MatchID),  
    FOREIGN KEY (RefereeID) REFERENCES Referee(RefereeID)  
);
```


5.4 GENERATING REPORTS

1. To display each transaction showing Match details, Match Date, and Match scores:

```
USE faz_db;

SELECT
    M.MatchID,
    M.Date,
    HT.TeamName AS HostTeam,
    GT.TeamName AS GuestTeam,
    M.FinalResult
FROM
    Matches M
JOIN
    Team HT ON M.HostTeamID = HT.TeamID
JOIN
    Team GT ON M.GuestTeamID = GT.TeamID
ORDER BY
    M.Date,
    M.MatchID;
```

2. Sort player names alphabetically within each match transaction:

```
USE faz_db;
SELECT
    part.MatchID,
    CONCAT(p.FName, ' ', p.LName) AS PlayerName,
    t.TeamName AS TeamName
FROM
    Participation part
JOIN
    Player p ON part.PlayerID = p.PlayerID
JOIN
    Team t ON p.TeamID = t.TeamID
ORDER BY
    p.LName, p.FName;
```

REFERENCES

Book: Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems (7th Edition). Pearson.

Transfermarkt. (n.d.). Football Player Information and Statistics. Retrieved from <https://www.transfermarkt.com/>

W3Schools. (n.d.). SQL Tutorial. Retrieved from <https://www.w3schools.com/sql>

Book: Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database System Concepts (7th Edition). McGraw-Hill.

JavaScript: Flanagan, D. (2020). JavaScript: The Definitive Guide (7th Edition). O'Reilly Media.

Node.js: Teixeira, M. (2018). Node.js Design Patterns (2nd Edition). Packt Publishing.

HTML & CSS: Duckett, J. (2011). HTML and CSS: Design and Build Websites. John Wiley & Sons.
Web Development Tools

Book: Hoye, R., Smith, A., Nicholson, M., Stewart, B., & Westerbeek, H. (2015). Sport Management: Principles and Applications (4th Edition). Routledge.