## horizontal line



Sport Ligue Application Database

Comp 352

**─**

Group Members

Amanuel

Asmelash

Musie

Samuel Girum

Samuel Mideksa 067/BSC-B4/2022

Yohanes Samuel

# Table of contents

[**Table of contents 2**](#_heading=h.dlzugtdj5lkw)

[**1. Chapter 1: Introduction 4**](#_heading=h.voa1c8gtxo59)

[**1.1. Background 4**](#_heading=h.8x7o43petwvr)

[**1.2. Purpose of the system 4**](#_heading=h.cbzj35rv4oq1)

[**1.3. Statements of the problem 4**](#_heading=h.5yln0klcqexo)

[**1.4. Scope of the project 4**](#_heading=h.zkzlvrl3qka)

[**1.5. Objectives of the project 4**](#_heading=h.lmmcuwyswmgl)

[**1.6. General objective 4**](#_heading=h.o20l8bn8itmc)

[**1.7. Specific objectives 4**](#_heading=h.s7pk8fcnc7e2)

[**1.8. Database Development Methodology 4**](#_heading=h.d0wdodqz9cxx)

[**1.9. Data Sources & Collection Methods 4**](#_heading=h.kjq0dtnjx13x)

[**1.10. DB Analysis and Design Methods 4**](#_heading=h.6qghju98svos)

[**1.11. Deliverables of the Project 4**](#_heading=h.y1pdgxnvt2hh)

[**1.12. Development Tools, Platforms, and Technologies 4**](#_heading=h.1zgp1d49wrnq)

[**1.13. Project Time Plan 4**](#_heading=h.a0mkir6rre01)

[**2. Chapter 2-Requirement Specification 5**](#_heading=h.hapapiep8ufm)

[**2.1. Data Requirements 5**](#_heading=h.c3wukxybusu0)

[**2.2. Transaction Requirements 5**](#_heading=h.7a6btnib4623)

[**2.3. Data Entry Requirements 5**](#_heading=h.le2gi8tbreg2)

[**2.4. Data Retrieval Requirements 5**](#_heading=h.xmqlt0po3d93)

[**2.5. Data Updating Requirements 5**](#_heading=h.81a4e6s9weg8)

[**2.6. Data Removal Requirements 5**](#_heading=h.n50699o3gzu3)

[**3. Chapter 3 Database Design 6**](#_heading=h.t46q8znecc6e)

[**3.1. Conceptual database design of the new system 6**](#_heading=h.o9q19c9b7o2d)

[**3.1.1. Entities with their description 6**](#_heading=h.lw5s5nnq41gg)

[**3.1.2. Attributes with their description 6**](#_heading=h.clvp4r8ueqnk)

[**3.1.3. Relation ships between the entities 6**](#_heading=h.gtmjmvr6uqiq)

[**3.1.4. E-R diagram 6**](#_heading=h.jai3r3tqz0ws)

[**3.2. Logical Database Design 6**](#_heading=h.xtn25o4nzz9o)

[**3.2.1. ER-Relation Mapping 6**](#_heading=h.ijqriw3n1k1v)

[**3.2.2. Validating model with Normalization 6**](#_heading=h.fypzbgu25yvl)

[**3.2.2.1. First Normal Form (1NF) 6**](#_heading=h.llcf1edyo5n7)

[**3.2.2.2. Second Normal Form (2NF) 6**](#_heading=h.tawr0o4kv2o5)

[**3.2.2.3. Third Normal Form (3NF) 6**](#_heading=h.q7lguum8nbq4)

[**3.2.3. Relational Schema with referential Integrity after normalization 6**](#_heading=h.2qmpni4prp9e)

[**3.3. Physical database design 6**](#_heading=h.ta2efqmzbqd2)

[**3.3.1. Physical design strategy 6**](#_heading=h.f29303xa4324)

[**3.3.2. Database Deployment details 6**](#_heading=h.7xpi1zxz2niy)

[**4. Chapter 4 - Implementation and testing 7**](#_heading=h.a3usuan7i6po)

[**4.1. SQL script for creating the database 7**](#_heading=h.w2iz2bqimnkv)

[**4.2. SQL Scripts for creating the tables, view, and indexes. 7**](#_heading=h.es6hgddyh02z)

[**4.3. SQL Scripts for manipulating the database 7**](#_heading=h.9sk2zsxrl6lw)

[**4.4. SQL Scripts for Database Control 7**](#_heading=h.66374e4reues)

[**4.5. Testing 7**](#_heading=h.aj32su9n0fdx)

[**References 8**](#_heading=h.nc4rf1pu5eex)

[**Appendix (Forms, Reports of the Organization etc.) 9**](#_heading=h.dvlx1z9925wi)

[**Goals 9**](#_heading=h.3at9u9s4e0vp)

[**Specifications 9**](#_heading=h.4p7xi5bvhxdr)

[Lorem Ipsum 12](#_heading=h.56kfpodyq5td)

[**Milestones 12**](#_heading=h.yyrhu7ml5bea)

[I. Lorem ipsum 12](#_heading=h.buwz1tcz7y35)

[II. Dolor sit amet 12](#_heading=h.p2nityf5kx5q)

[**DDL 14**](#_heading=h.cl3vt431klsi)

# 

# Chapter 1: Introduction

## Background

## Purpose of the system

## Statements of the problem

## Scope of the project

## Objectives of the project

## General objective

## Specific objectives

## Database Development Methodology

## Data Sources & Collection Methods

## DB Analysis and Design Methods

## Deliverables of the Project

## Development Tools, Platforms, and Technologies

## Project Time Plan

# Chapter 2-Requirement Specification

## Data Requirements

## Transaction Requirements

## Data Entry Requirements

## Data Retrieval Requirements

## Data Updating Requirements

## Data Removal Requirements

# Chapter 3 Database Design

## Conceptual database design of the new system

## Entities with their description

## Attributes with their description

## Relation ships between the entities

## E-R diagram

## Logical Database Design

## ER-Relation Mapping

## Validating model with Normalization

## First Normal Form (1NF)

## Second Normal Form (2NF)

## Third Normal Form (3NF)

## Relational Schema with referential Integrity after normalization

## Physical database design

## Physical design strategy

## Database Deployment details

# 

# Chapter 4 - Implementation and testing

## SQL script for creating the database

## SQL Scripts for creating the tables, view, and indexes.

## SQL Scripts for manipulating the database

## SQL Scripts for Database Control

## Testing

# References

# Appendix (Forms, Reports of the Organization etc.)

# 

# 

# 

# 

# 

# 

# 

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper.

# Goals

1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit
2. Sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

# Specifications

Thank you for providing the updated structure with camel case and abbreviations in all capital. The structure you provided looks good with the appropriate naming conventions. It follows the recommended conventions for column and table names. Here's a recap of the revised structure:

Tables:

1. Seasons
2. Clubs
3. Venues
4. Managers
5. ClubManager
6. Players
7. Officials
8. Fixtures
9. Matches
10. AppearanceStats
11. Goals
12. Cards
13. Penalties
14. Substitutions
15. AttackStats
16. TeamPlayStats
17. Discipline
18. Defence
19. Goalkeeping
20. PlayerTransactions
21. PlayerContract
22. PlayerSalary
23. ManagerContract
24. ManagerSalary
25. MonthlyPlayerAwards
26. MOTM

- LeagueSeason (seasonID, startDate, endDate, seasonStatus)

- Club (clubName, home, website, twitter, instagram, facebook, youtube, tikTok, squad)

- Venue (venueID, venueName, location, capacity)

- Manager (managerID, firstName, lastName, countryOB, status, DOB, club)

- ManagerClub (managerID, clubID, startDate, endDate)

- Player (playerID, firstName, lastName, DOB, cityOB, countryOB, nationality, height, club, position, squadNumber, twitter, instagram, facebook, youtube, tikTok)

- AppearanceStats (appearanceID, type (start, substitution), playerID, matchID)

- Goal (goalID, matchID, type (headedGoal, leftFootGoal, rightFootGoal, penaltiesScored, freeKickScored, ownGoal), assist, playerID, goalFor, goalAt)

- Card (cardID, matchID, playerID, time, type)

- Penalty (penaltyID, matchID, time, type, goal, teamFor, teamAt)

- Substitution (substitutionID, matchID, playerInID, playerOutID, time, description)

- AttackStats (playerID, matchID, teamFor, headedGoal, leftFootGoal, rightFootGoal, penaltiesScored, freeKickScored, shot, shotOnTarget, hitWoodwork, bigChanceMissed)

- TeamPlayStats (playerID, matchID, teamFor, assists, passes, bigChancesCreated, crosses, throughBalls, accurateLongBalls)

- Discipline (playerID, matchID, teamFor, yellowCards, redCards, fouls, offsides)

- Defense (playerID, matchID, teamFor, tackles, blocks, interceptions, clearances, headedClearances, recoveries, duelsWon, successful5050s, aerialBattlesWon, aerialBattlesWon, aerialBattlesLost, errorsLeadingToGoal, ownGoals)

- Goalkeeping (keeperID, matchID, teamFor, saves, penaltiesSaved, punches, highClaims, catches, sweeperClearances, throwOuts, goalKicks)

- Official (officialID, firstName, lastName, role, nationality)

- Fixture (fixtureID, homeTeamID, awayTeamID, date, kickoff, venueID, referee, status)

- Result (matchID, hostClubName, guestClubName, venueID, kickoff, refereeID, assistantReferee1ID, assistantReferee2ID, fourthRefereeID, guestClubName, HTHostScore, HTGuestScore, FTHostScore, FTGuestScore)

- MatchStats (matchID, teamID, possession, shotsOnTarget, shots, touches, passes, tackles, clearances, corners, offsides, foulsConceded)

- MatchEvent (matchEventID, matchID, eventType, playerID, time)

- PlayerTransaction (transactionID, playerID, transactionType (transfer, loan, release), fromTeam, toTeam, transactionDate)

- PlayerContract (contractID, playerID, startDate, endDate, contractType, contractStatus, transferFee, buyoutClause)

- PlayerSalary (salaryID, playerID, startDate, endDate, paymentFrequency, bonuses)

- ManagerContract (contractID, managerID, startDate, endDate, contractStatus)

- ManagerSalary (salaryID, managerID, startDate, endDate)

- POTM (seasonID, month, playerID)

- MOTM (seasonID, month, managerID)

- GOTM (seasonID, month, playerID, goalID)

- SOTM (seasonID, month, playerID)

## Lorem Ipsum

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan.

# Milestones

## Lorem ipsum

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

## Dolor sit amet

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

# DDL

-- !!! datatype for IDs

-- Create League table

CREATE TABLE League (

leagueID INT PRIMARY KEY,

leagueName VARCHAR(255) NOT NULL

);

-- Create Seasons table

CREATE TABLE Seasons (

seasonID INT PRIMARY KEY,

leagueID INT NOT NULL,

startDate DATE NOT NULL,

endDate DATE,

seasonStatus VARCHAR(50),

MOTS INT,

POTS INT,

YPOTS INT,

GOTS INT,

SOTS INT,

GC INT,

GB INT,

GG INT,

Playmaker INT,

MIC INT,

MPG INT,

FOREIGN KEY (leagueID) REFERENCES League(leagueID),

FOREIGN KEY (MOTS) REFERENCES Managers(managerID),

FOREIGN KEY (POTS) REFERENCES Players(playerID),

FOREIGN KEY (YPOTS) REFERENCES Players(playerID),

FOREIGN KEY (GOTS) REFERENCES Goals(goalID),

FOREIGN KEY (SOTS) REFERENCES Players(playerID),

FOREIGN KEY (GC) REFERENCES Players(playerID),

FOREIGN KEY (GB) REFERENCES Players(playerID),

FOREIGN KEY (GG) REFERENCES Players(playerID),

FOREIGN KEY (Playmaker) REFERENCES Players(playerID),

FOREIGN KEY (MIC) REFERENCES Clubs(clubID),

FOREIGN KEY (MPG) REFERENCES Goal(goalID),

CHECK (seasonStatus IN('Upcoming', 'In progress', 'Completed', 'Suspended', 'Postponed', 'Cancelled', 'Ended'))

);

-- Create Clubs table

CREATE TABLE Clubs (

clubID INT PRIMARY KEY,

clubName VARCHAR(255) NOT NULL,

clubAbbreviation CHAR(3) NOT NULL,

home VARCHAR(255) NOT NULL,

website VARCHAR(255) NOT NULL,

twitter VARCHAR(255),

instagram VARCHAR(255),

facebook VARCHAR(255),

youtube VARCHAR(255),

tikTok VARCHAR(255)

);

-- Create Venues table

CREATE TABLE Venues (

venueID INT PRIMARY KEY,

venueName VARCHAR(255),

venueLocation VARCHAR(255),

capacity INT NOT NULL

);

-- Create Managers table

CREATE TABLE Managers (

managerID INT PRIMARY KEY,

firstName VARCHAR(255) NOT NULL,

lastName VARCHAR(255) NOT NULL,

nationality VARCHAR(255) NOT NULL,

managerStatus VARCHAR(50) NOT NULL,

DOB DATE NOT NULL,

clubID INT,

managerBio text,

FOREIGN KEY (clubID) REFERENCES Clubs(clubID),

CHECK (firstName NOT LIKE %[^A-Za-z.\\p{L}]),

CHECK (lastName NOT LIKE %[^A-Za-z.\\p{L}]),

CHECK (managerStatus IN('Active', 'Inactive'))

);

-- Create ManagerClub table

--PRIMARY KEY not selected

CREATE TABLE ManagerClub (

managerID INT,

clubID INT,

startDate DATE,

endDate DATE,

FOREIGN KEY (managerID) REFERENCES Managers(managerID),

FOREIGN KEY (clubID) REFERENCES Clubs(clubID)

);

-- Create Players table

CREATE TABLE Players (

playerID INT PRIMARY KEY,

firstName NVARCHAR(255) NOT NULL,

lastName NVARCHAR(255) NOT NULL,

DOB DATE NOT NULL,

cityOB VARCHAR(255) NOT NULL,

countryOB VARCHAR(255) NOT NULL,

nationality VARCHAR(255) NOT NULL,

height INT NOT NULL,

clubID INT NOT NULL,

position VARCHAR(255) NOT NULL,

squadNumber INT NOT NULL,

twitter VARCHAR(255),

instagram VARCHAR(255),

facebook VARCHAR(255),

youtube VARCHAR(255),

tikTok VARCHAR(255),

FOREIGN KEY (clubID) REFERENCES Clubs(clubID),

CHECK (firstName NOT LIKE %[^A-Za-z.\\p{L}]),

CHECK (lastName NOT LIKE %[^A-Za-z.\\p{L}])

);

-- Create AppearanceStats table

CREATE TABLE AppearanceStats (

appearanceID INT PRIMARY KEY,

apperanceType VARCHAR(50) NOT NULL,

playerID INT NOT NULL,

matchID INT NOT NULL,

FOREIGN KEY (playerID) REFERENCES Players(playerID),

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

CHECK (type IN('Start', 'Substitution'))

);

--Tables to store match events

-- Create Goals table

CREATE TABLE Goals (

goalID INT PRIMARY KEY,

matchID INT NOT NULL,

goalType VARCHAR(50) NOT NULL,

assistBy INT,

playerID INT NOT NULL,

goalFor INT NOT NULL,

goalAt INT NOT NULL,

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

FOREIGN KEY (assistBy) REFERENCES Players(playerID),

FOREIGN KEY (playerID) REFERENCES Players(playerID),

FOREIGN KEY (goalFor) REFERENCES Clubs(clubID),

FOREIGN KEY (goalAt) REFERENCES Clubs(clubID),

CHECK (type IN('Headed goal', 'Goal with right foot', 'Goal with left foot', 'Penalty Scored', 'Freekick scored', 'Own goal'))

);

-- Create Cards table

CREATE TABLE Cards (

cardID INT PRIMARY KEY,

matchID INT NOT NULL,

playerID INT NOT NULL,

cardMinute SMALLINT NOT NULL,

cardType VARCHAR(50) NOT NULL,

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

FOREIGN KEY (playerID) REFERENCES Players(playerID)

CHECK (cardType IN('Yellow', 'Red'))

);

-- Create Penalties table

CREATE TABLE Penalties (

penaltyID INT PRIMARY KEY,

matchID INT NOT NULL,

penaltyMinute SMALLINT,

penaltyType VARCHAR(50) NOT NULL,

goal BOOLEAN NOT NULL,

teamFor INT NOT NULL,

teamAt INT NOT NULL,

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

FOREIGN KEY (teamFor) REFERENCES Clubs(clubID),

FOREIGN KEY (teamAt) REFERENCES Clubs(clubID),

CHECK (penaltyType IN('Regular penalty', 'Penalty shootout', 'Retake'))

);

-- Create Substitutions table

CREATE TABLE Substitutions (

substitutionID INT PRIMARY KEY,

matchID INT NOT NULL,

playerInID INT NOT NULL,

playerOutID INT NOT NULL,

substitutionMinute SMALLINT NOT NULL,

description VARCHAR(255),

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

FOREIGN KEY (playerInID) REFERENCES Players(playerID),

FOREIGN KEY (playerOutID) REFERENCES Players(playerID)

);

-- Create AttackStats table

CREATE TABLE AttackStats (

attackID INT PRIMARY KEY,

playerID INT NOT NULL,

matchID INT NOT NULL,

teamFor INT NOT NULL,

headedGoal SMALLINT NOT NULL SET DEFAULT 0,

leftFootGoal SMALLINT NOT NULL SET DEFAULT 0,

rightFootGoal SMALLINT NOT NULL SET DEFAULT 0,

penaltiesScored SMALLINT NOT NULL SET DEFAULT 0,

freekickScored SMALLINT NOT NULL SET DEFAULT 0,

shot SMALLINT NOT NULL SET DEFAULT 0,

shotOnTarget SMALLINT NOT NULL SET DEFAULT 0,

hitWoodwork SMALLINT NOT NULL SET DEFAULT 0,

bigChanceMissed SMALLINT NOT NULL SET DEFAULT 0,

FOREIGN KEY (playerID) REFERENCES Players(playerID),

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

FOREIGN KEY (teamFor) REFERENCES Clubs(clubID)

);

-- Create TeamPlayStats table

CREATE TABLE TeamPlayStats (

teamPlayID INT PRIMARY KEY,

playerID INT NOT NULL,

matchID INT NOT NULL,

teamFor INT NOT NULL,

assists SMALLINT NOT NULL SET DEFAULT 0,

passes SMALLINT NOT NULL SET DEFAULT 0,

bigChancesCreated SMALLINT NOT NULL SET DEFAULT 0,

crosses SMALLINT NOT NULL SET DEFAULT 0,

throughBalls SMALLINT NOT NULL SET DEFAULT 0,

accurateLongBalls SMALLINT NOT NULL SET DEFAULT 0,

FOREIGN KEY (playerID) REFERENCES Players(playerID),

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

FOREIGN KEY (teamFor) REFERENCES Clubs(clubID)

);

-- Create Discipline table

CREATE TABLE Discipline (

disciplineID INT PRIMARY KEY,

playerID INT NOT NULL,

matchID INT NOT NULL,

teamFor INT NOT NULL,

yellowCards SMALLINT NOT NULL SET DEFAULT 0,

redCards SMALLINT NOT NULL SET DEFAULT 0,

fouls SMALLINT NOT NULL SET DEFAULT 0,

offsides SMALLINT NOT NULL SET DEFAULT 0,

FOREIGN KEY (playerID) REFERENCES Players(playerID),

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

FOREIGN KEY (teamFor) REFERENCES Clubs(clubID)

);

-- Create Defence table

CREATE TABLE Defence (

defenceID INT PRIMARY KEY,

playerID INT NOT NULL,

matchID INT NOT NULL,

teamFor INT NOT NULL,

tackles SMALLINT NOT NULL SET DEFAULT 0,

blocks SMALLINT NOT NULL SET DEFAULT 0,

interceptions SMALLINT NOT NULL SET DEFAULT 0,

clearances SMALLINT NOT NULL SET DEFAULT 0,

headedClearances SMALLINT NOT NULL SET DEFAULT 0,

recoveries SMALLINT NOT NULL SET DEFAULT 0,

duelsWon SMALLINT NOT NULL SET DEFAULT 0,

successful50\_50s SMALLINT NOT NULL SET DEFAULT 0,

aerialBattlesWon SMALLINT NOT NULL SET DEFAULT 0,

aerialBattlesLost SMALLINT NOT NULL SET DEFAULT 0,

errorsLeadingToGoal SMALLINT NOT NULL SET DEFAULT 0,

ownGoals SMALLINT NOT NULL SET DEFAULT 0,

FOREIGN KEY (playerID) REFERENCES Players(playerID),

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

FOREIGN KEY (teamFor) REFERENCES Clubs(clubID)

);

-- Create Goalkeeping table

CREATE TABLE Goalkeeping (

goalKeepingID INT PRIMARY KEY,

keeperID INT NOT NULL,

matchID INT NOT NULL,

teamFor INT NOT NULL,

saves SMALLINT NOT NULL SET DEFAULT 0,

penaltiesSaved SMALLINT NOT NULL SET DEFAULT 0,

punches SMALLINT NOT NULL SET DEFAULT 0,

highClaims SMALLINT NOT NULL SET DEFAULT 0,

catches SMALLINT NOT NULL SET DEFAULT 0,

sweeperClearances SMALLINT NOT NULL SET DEFAULT 0,

throwOuts SMALLINT NOT NULL SET DEFAULT 0,

goalKicks SMALLINT NOT NULL SET DEFAULT 0,

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

FOREIGN KEY (teamFor) REFERENCES Clubs(clubID)

);

-- Create Officials table

CREATE TABLE Officials (

officialID INT PRIMARY KEY,

firstName NVARCHAR(255) NOT NULL ,

highestRole VARCHAR(50) NOT NULL

lastName NVARCHAR(255) NOT NULL ,

nationality VARCHAR(255) NOT NULL,

CHECK (firstName NOT LIKE %[^A-Za-z.\\p{L}]),

CHECK (highestRole IN('Referee', 'Assistant referee', 'Fourth official')),

CHECK (lastName NOT LIKE %[^A-Za-z.\\p{L}])

);

-- Create Fixtures table

CREATE TABLE Fixtures (

fixtureID INT PRIMARY KEY,

homeTeamID INT NOT NULL,

awayTeamID INT NOT NULL,

fixtureDate DATE NOT NULL,

kickoff TIME NOT NULL,

venueID INT NOT NULL,

refreeID INT NOT NULL,

assistantRefree1ID INT NOT NULL,

assistantRefree2ID INT NOT NULL,

fourthOfficialID INT NOT NULL,

fixtureStatus VARCHAR(50) NOT NULL,

FOREIGN KEY (homeTeamID) REFERENCES Clubs(clubID),

FOREIGN KEY (awayTeamID) REFERENCES Clubs(clubID),

FOREIGN KEY (venueID) REFERENCES Venues(venueID),

FOREIGN KEY (refreeID) REFERENCES Officials(officialID),

FOREIGN KEY (assistantRefree1ID) REFERENCES Officials(officialID),

FOREIGN KEY (assistantRefree2ID) REFERENCES Officials(officialID),

FOREIGN KEY (fourthOfficialID) REFERENCES Officials(officialID),

CHECK (refereeID IN (SELECT officialID FROM Officials WHERE highestRole = 'Referee')),

CHECK (assistantReferee1ID IN (SELECT officialID FROM Officials WHERE highestRole IN('Referee', 'Assistant referee'))),

CHECK (assistantReferee2ID IN (SELECT officialID FROM Officials WHERE highestRole IN('Referee', 'Assistant referee'))),

CHECK (fourthOfficialID IN (SELECT officialID FROM Officials WHERE highestRole IN('Referee', 'Assistant referee', 'Fourth offical'))),

CHECK (fixtureStatus IN('Scheduled', 'In progress', 'Completed', 'Canceled', 'Postponed', 'Forfeit', 'Abandoned'))

);

-- Create Results table

CREATE TABLE Results (

resultID INT PRIMARY KEY,

matchID INT NOT NULL,

fixtureID INT NOT NULL,

HTHomeTeamScore INT NOT NULL,

HTAwayTeamScore INT NOT NULL,

FTHomeTeamScore INT NOT NULL,

FTAwayTeamScore INT NOT NULL,

FOREIGN KEY (matchID) REFERENCES Matches(matchID),

FOREIGN KEY (fixtureID) REFERENCES Fixtures(fixtureID),

);

-- Create PlayerTransactions table

CREATE TABLE PlayerTransactions (

transactionID INT PRIMARY KEY,

playerID INT NOT NULL,

transactionType VARCHAR(50) NOT NULL,

fromTeam INT,

toTeam INT NOT NULL,

transactionDate DATE NOT NULL,

FOREIGN KEY (playerID) REFERENCES Players(playerID),

FOREIGN KEY (fromTeam) REFERENCES Clubs(clubID),

FOREIGN KEY (toTeam) REFERENCES Clubs(clubID),

CHECK (transactionType IN('Loan', 'Transfer', 'Release', 'Resign'))

);

-- Create PlayerContract table

CREATE TABLE PlayerContract (

contractID INT PRIMARY KEY,

playerID INT NOT NULL,

clubID INT NOT NULL,

startDate DATE NOT NULL,

endDate DATE NOT NULL,

contractValue MONEY NOT NULL,

contractType VARCHAR(50) NOT NULL,

contractStatus VARCHAR(50) NOT NULL,

transferFee MONEY NOT NULL,

buyoutClause MONEY,

FOREIGN KEY (playerID) REFERENCES Players(playerID),

FOREIGN KEY (clubID) REFERENCES Clubs(clubID),

CHECK (contractType IN('Professional', 'Youth Development', 'Academy')),

CHECK (contractStatus IN('Active', 'Expired', 'Cancelled', 'Terminated'))

);

-- Create PlayerSalary table

CREATE TABLE PlayerSalary (

salaryID INT PRIMARY KEY,

playerID INT NOT NULL,

startDate DATE NOT NULL,

endDate DATE NOT NULL,

salaryAmount MONEY NOT NULL,

paymentFrequency VARCHAR(50) NOT NULL,

bonuses MONEY,

bonusPaymentFrequency VARCHAR(50),

FOREIGN KEY (playerID) REFERENCES Players(playerID)

);

-- Create ManagerContract table

CREATE TABLE ManagerContract (

contractID INT PRIMARY KEY,

managerID INT NOT NULL,

clubID INT NOT NULL,

startDate DATE NOT NULL,

endDate DATE NOT NULL,

contractStatus VARCHAR(50) NOT NULL,

FOREIGN KEY (clubID) REFERENCES Clubs(clubID),

FOREIGN KEY (managerID) REFERENCES Managers(managerID),

CHECK (contractStatus IN('Active', 'Expired', 'Cancelled', 'Terminated'))

);

-- Create ManagerSalary table

CREATE TABLE ManagerSalary (

salaryID INT PRIMARY KEY,

managerID INT NOT NULL,

startDate DATE NOT NULL,

endDate DATE NOT NULL,

salaryAmount MONEY NOT NULL,

paymentFrequency VARCHAR(50) NOT NULL,

bonuses MONEY,

bonusPaymentFrequency VARCHAR(50),

FOREIGN KEY (managerID) REFERENCES Managers(managerID)

);

-- Create Awards tables

-- In season monthly awards

CREATE TABLE MonthlyPlayerAwards (

seasonID INT NOT NULL,

awardYear INT NOT NULL,

awardMonth INT NOT NULL,

playerID INT NOT NULL,

matchID INT,

goalID INT,

PRIMARY KEY (seasonID, awardMonth),

FOREIGN KEY (seasonID) REFERENCES Seasons(seasonID),

FOREIGN KEY (playerID) REFERENCES Players(playerID),

CHECK (awardMonth BETWEEN 1 AND 12),

CHECK (award IN ('Player of the Month', 'Goal of the Month', 'Save of the Month'))

)

CREATE TABLE MOTM (

seasonID INT,

awardMonth INT,

managerID INT,

PRIMARY KEY (seasonID, awardMonth),

FOREIGN KEY (seasonID) REFERENCES Seasons(seasonID),

FOREIGN KEY (managerID) REFERENCES Managers(managerID),

CHECK (awardMonth BETWEEN 1 AND 12)

);

-- All-Time Seasonal awards

CREATE TABLE SeasonalPlayerAwards (

seasonID INT NOT NULL,

playerID INT NOT NULL,

award VARCHAR(100) NOT NULL,

matchID INT,

goalID INT,

PRIMARY KEY (seasonID, awardMonth),

FOREIGN KEY (seasonID) REFERENCES Seasons(seasonID),

FOREIGN KEY (playerID) REFERENCES Players(playerID),

CHECK (awardMonth BETWEEN 1 AND 12),

CHECK (award IN (''))

)

CREATE TABLE SeasonalTeamAwards

CREATE TABLE MOTS (

MOTSawardID INT PRIMARY KEY,

seasonID INT,

managerID INT,

FOREIGN KEY (seasonID) REFERENCES Seasons(seasonID),

FOREIGN KEY (managerID) REFERENCES Managers(managerID)

);

<iframe width="560" height="315" src='https://dbdiagram.io/embed/647cb7d6722eb774945d5960'> </iframe>