

*نمودار پروژه قبل و فایل db و sql در فایل وجود دارند.

*آن هایی که Bold هستند PK هستند.

*از نرم افزار SQLiteStudio-3.2.1 استفاده شده است.

*با توجه به اسلاید ها به Relational model تبدیل شده ولی برای تبدیل به کد sql تغییراتی داده شده است.

بررسی Relational model پروژه و کد sql :

۱. دو موجودیت Team و Players :

*رابطه بین Team و Players به صورت Plays in است و به این شکل است که یک تیم

میتواند یک یا چند بازیکن داشته باشد اما یک بازیکن فقط میتواند در یک تیم بازی کند.

و رابطه به شکل many-one است . پس باید ویژگی رابطه یعنی plays in را در جدول

Players بیاوریم و PK جدول Team را در جدول Players بیاوریم

چون ویژگی position برای Players یک multivalued attributs است باید به صورت

رابطه has با Players تبدیل بکنیم پس یک جدول برای position و PlayersPosition

درست میکنیم.

Team(ID ,Name, LeagueName, WinNo, CupNo, LoseNo, PenaltyNo,

Salary, InjuryNo)

Players(ID ,Name, Nationality, GoalNo, CupNo, AssistNo, GameNo,

Duration of injury, InjuryNo, Team.ID, StartTheContract,

EndTheContract)

Position(ID, name)

PlayersPosition(ID, Players.ID, Position.ID)

حال کد sql :

Team:

```
CREATE TABLE Team(  
    teamID INTEGER PRIMARY KEY AUTOINCREMENT,  
    teamName VARCHAR(255) NOT NULL,  
    LeagueName VARCHAR(255) NOT NULL,
```

```

teamWinNo INT NULL,
teamCupNo INT NULL,
teamLoseNo INT NULL,
PenaltyNo INT NULL,
Salary REAL NULL,
teamInjuryNo INT NULL,
stadiumID INTEGER not NULL,
UNIQUE(teamName,LeagueName)
);

```

چون teamID به صورت PRIMARY KEY AUTOINCREMENT تعریف شده است پس فقط یک PK که همین است می توان تعریف کرد و بقیه PK ها براساس پروژه دوم را به صورت CK تعریف میکنیم که NOT NULL است و UNIQUE.

***این نکته برای همه صدق میکند.**

Players:

```

CREATE TABLE Players(
    playersID INTEGER PRIMARY KEY AUTOINCREMENT,
    playersName VARCHAR(255) NOT NULL ,
    playersNationality VARCHAR(255) NOT NULL ,
    playersGoalNo INT NULL,
    playersCupNo INT NULL,
    AssistNo INT NULL,
    playersGameNo INT NULL,
    DurationOfInjury REAL NULL,
    playersInjuryNo INT NULL,
    playersCardNo INT NULL,
    StartTheContract TEXT NOT NULL,
    EndTheContract TEXT NOT NULL,
    TeamID INTEGER NOT NULL,
    UNIQUE(playersName, playersNationality, TeamID),
    CONSTRAINT FK_Team_tID_tName_LeagueName FOREIGN KEY(TeamID)
    REFERENCES Team(teamID),
);

```

چون یک بازیکن میتواند فقط در یک تیم بازی کند پس teamID به صورت UNIQUE است.

Postiton:

```

CREATE TABLE Position(
    posID INTEGER PRIMARY KEY AUTOINCREMENT,
    posName VARCHAR(255) NOT NULL,
    UNIQUE(posName)
);

```

```
);
```

PlayersPosition:

```
CREATE TABLE PlayersPosition(  
    ppID INTEGER PRIMARY KEY AUTOINCREMENT,  
    playersID INTEGER NOT NULL,  
    PositionID INTEGER NOT NULL,  
    CONSTRAINT FK_pID_playersID FOREIGN KEY(playersID)  
    REFERENCES Players(playersID),  
    CONSTRAINT FK_posID FOREIGN KEY(PositionID)  
    REFERENCES Position(posID)  
);
```

یک بازیکن میتواند یک یا چندین پوزیشن داشته باشد و یک پوزیشن هم می تواند برای یک بازیکن یا چند بازیکن باشد به همین دلیل playersID و PositionID به صورت not null تعریف شده اند و UNIQUE نیستند.

۲. دو موجودیت Team و Coach:

*رابطه بین Team و Coach به صورت Lead است و به این صورت است که یک تیم فقط یک مربی دارد و یک مربی فقط یک تیم را Lead میکند. پس رابطه one-one است. پس جدول برای Team و Coach و یک جدول هم برای رابطه بین تیم و مربی درست میکنیم به نام CoachLeadTeam که در این جدول ویژگی های رابطه یعنی Lead قرار دارد و PK های Team و Coach به اینجا منتقل میکنیم.

*تیم در مورد شماره ۱ آمده است

Coach(ID, Name, Nationality, GameNo, CupNo)

CoachLeadTeam(ID, Coach.ID, Team.ID, StartTheContract, EndTheContract)

Sk={Coach.ID, Team.ID}

Coach:

```
CREATE TABLE Coach(  
    coachID INTEGER PRIMARY KEY AUTOINCREMENT,  
    coachName VARCHAR(255) NOT NULL,  
    coachNationality VARCHAR(255) NOT NULL,  
    coachGameNo INTEGER NULL,  
    coachCupNo INTEGER NULL,
```

```
    UNIQUE(coachName, coachNationality),  
);
```

CoachLeadTeam:

```
CREATE TABLE CoachLeadTeam(  
    leadID INTEGER PRIMARY KEY AUTOINCREMENT,  
    coachID INTEGER NOT NULL,  
    teamID INTEGER NOT NULL,  
    StartTheContract TEXT NOT NULL,  
    EndTheContract TEXT NOT NULL,  
    UNIQUE(coachID, teamID),  
    CONSTRAINT CK_coach_name_nationality FOREIGN KEY(coachID)  
    REFERENCES Coach(coachID),  
    CONSTRAINT SK_team_ID FOREIGN KEY(teamID)  
    REFERENCES Team(teamID)  
);
```

یک مربی یک تیم را Lead میکند و یک تیم یک مربی دارد پس UNIQUE است.

۳. دو موجودیت Team و TheManager:

*رابطه Team و TheManager به صورت Manage است و یک تیم فقط میتواند یک مدیر داشته باشد و یک مدیر فقط میتواند یک تیم را مدیریت بکند. پس رابطه one-one است.
پس جدول برای Team و TheManager و یک جدول هم برای Manage درست میکنیم.

TheManager(ID, Name, Nationality, CupNo, GameNo)

**TheManager_Manage_Team(ID, TheManager.ID,
Team.ID, StartTheContract, EndTheContract)**

SK={Team.ID, TheManager.ID}

The Manager:

```
CREATE TABLE TheManager(  
    managerID INTEGER PRIMARY KEY AUTOINCREMENT,  
    managerName VARCHAR(255) NOT NULL,  
    managerNationality VARCHAR(255) NOT NULL,  
    managerCupNo INTEGER NULL,  
    managerGameNo INTEGER NULL,  
    UNIQUE(managerName, managerNationality)  
);
```

TheManger_Manage_Team:

```
CREATE TABLE Manager_Manage_Team(
    manageID INTEGER PRIMARY KEY AUTOINCREMENT,
    managerID INTEGER NOT NULL,
    teamID INTEGER NOT NULL,
    StartTheContract TEXT NOT NULL,
    EndTheContract TEXT NOT NULL,
    UNIQUE(managerID, teamID),
    CONSTRAINT CK_TheManager_name_nationality FOREIGN KEY(managerID)
    REFERENCES TheManager(managerID),
    CONSTRAINT SK_Team_ID FOREIGN KEY(teamID)
    REFERENCES Team(teamID)
);
```

۴. دو موجودیت Team و Sponser:

*رابطه Team و Sponser به صورت Deal است و یک تیم می تواند با هیچ یا چند Sponser قرار داد ببندد و یک اسپانسر هم میتواند با هیچ یا چند تیم قرار داد ببندد پس رابطه many-many است. به صورت زیر است

Sponser(ID, Name)

Deal(ID, Team.ID, Sponser.ID, DealTime, DealAmount)

Sponser:

```
CREATE TABLE Sponser(
    sponserID INTEGER PRIMARY KEY AUTOINCREMENT,
    sponserName VARCHAR(255) NOT NULL,
    UNIQUE(sponserName)
);
```

Deal:

```
CREATE TABLE TeamDealSponser(
    dealID INTEGER PRIMARY KEY AUTOINCREMENT,
    teamID INTEGER NULL,
    sponserID INTEGER NULL,
    DealTime TEXT NULL,
    DealAmount REAL NULL,
    CONSTRAINT SK_Team_name_leaguenam FOREIGN KEY(teamID)
    REFERENCES Team(teamID),
    CONSTRAINT SK_Sponser_ID_name FOREIGN KEY(sponserID)
    REFERENCES Sponser(sponserID)
);
```

```
);
```

۵. دو موجودیت User و Team :

*رابطه Team و User به صورت Favorite است و هر کاربر می تواند هیچ یا چند تیم را دوست داشته باشد و هر تیم می تواند هیچ یا چند کاربر آن را دوست داشته باشند. پس رابطه many- many است. پس داریم:

User(ID, UserName, pass, Email)

FavoriteTeam(ID, User.ID, Team.ID)

User:

```
CREATE TABLE User(  
    userID INTEGER PRIMARY KEY AUTOINCREMENT,  
    userName VARCHAR(255) NOT NULL,  
    password TEXT NOT NULL,  
    userEmail VARCHAR(255) NOT NULL,  
    UNIQUE(userName,userEmail)  
);
```

FavoriteTeam:

```
CREATE TABLE UserFavoriteTeam(  
    favoriteTeamID INTEGER PRIMARY KEY AUTOINCREMENT,  
    userID INTEGER NULL,  
    teamID INTEGER NULL,  
    CONSTRAINT CK_Team_ID FOREIGN KEY(teamID)  
    REFERENCES Team(teamID),  
    CONSTRAINT CK_User_ID FOREIGN KEY(userID)  
    REFERENCES User(userID)  
);
```

۶. دو موجودیت Team و Transfer :

*رابطه Team و Transfer به صورت Contract است و یک تیم می تواند هیچ یا چند نقل و انتقال داشته باشد. پس رابطه many-many است.

Transfer(ID, SellerTeam, PurchaserTeam)

Contract(ID, Team.ID, Transfer.ID, Cost)

Transfer:

```
CREATE TABLE Transfer(  
    transferID INTEGER PRIMARY KEY AUTOINCREMENT,  
    SellerTeam VARCHAR(255) NOT NULL,  
    PurchaserTeam VARCHAR(255) NOT NULL,  
);
```

Contract:

```
CREATE TABLE Contract(  
    contractID INTEGER PRIMARY KEY AUTOINCREMENT,  
    teamID INTEGER NULL,  
    transferID INTEGER NULL,  
    CONSTRAINT CK_TeamID FOREIGN KEY(teamID)  
    REFERENCES Team(teamID),  
    CONSTRAINT CK_TransferID FOREIGN KEY(transferID)  
    REFERENCES Transfer(transferID)  
);
```

۷. دو موجودیت User و Players :

*رابطه User و Players به صورت Favorite است و یک کاربر می تواند هیچ یا چند تیم را دوست داشته باشد و یک تیم توسط هیچ یا چند کاربر دوست داشته شود. پس رابطه -many many است.

* جدول User و Players در بالاتر آمده است.

FavoritePlayers(ID, User.ID, Players.ID, Grade)

FavoritePlayers:

```
CREATE TABLE UserFavoritePlayers(  
    favoritePlayersID INTEGER PRIMARY KEY AUTOINCREMENT,  
    userID INTEGER NULL,  
    playersID INTEGER NULL,  
    grade REAL NULL,  
    CONSTRAINT CK_UserID FOREIGN KEY(userID)  
    REFERENCES User(userID),  
    CONSTRAINT CK_PlayersID FOREIGN KEY(playersID)  
    REFERENCES Players(playersID)  
);
```

۸. دو موجودیت User و Match :

*رابطه User و Match به صورت Watch است و یک کاربر می تواند هیچ یا چند بازی را دیده باشد و یک بازی می تواند یک یا چند کاربر آن را دیده باشند. پس رابطه many-many است.

*جدول User بالاتر است.

Match(ID, HomeTeam, AwayTeam, StadiumName, GoalNo, CardNo, Day, Month, Year, HomeGameOrNot)
WatchMatch(ID, User.ID, Match.ID)

Match:

```
CREATE TABLE Match_(
    matchID INTEGER PRIMARY KEY AUTOINCREMENT,
    StadiumName VARCHAR(255) NOT NULL,
    Year INTEGER NOT NULL,
    Month INTEGER NOT NULL,
    Day TEXT NOT NULL,
    HomeTeam VARCHAR(255) NOT NULL,
    AwayTeam VARCHAR(255) NOT NULL,
    homegame_or_not TEXT NULL,
    matchGoalNo INTEGER NULL,
    matchCardNo INTEGER NULL,
    UNIQUE(StadiumName, Year, Month, Day, HomeTeam, AwayTeam)
);
```

WatchMatch:

```
CREATE TABLE UserWatchMatch(
    WatchID INTEGER PRIMARY KEY AUTOINCREMENT,
    matchID INTEGER NULL,
    userID INTEGER NULL,
    CONSTRAINT FK_matchID FOREIGN KEY(matchID)
    REFERENCES Match_(matchID),
    CONSTRAINT FK_userID FOREIGN KEY(userID)
    REFERENCES User(userID)
);
```

۹. دو موجودیت Match و Team :

*رابطه Team و Match به صورت plays است و یک تیم یک یا چند بازی انجام داده است و یک مسابقه برای یک تا چند تیم است. پس رابطه many-many است.

TeamPlaysMatch(ID, Team.ID, Match.ID)

TeamPlaysMatch:

```
CREATE TABLE TeamMatch(  
    TeamMatchID INTEGER PRIMARY KEY AUTOINCREMENT,  
    matchID INTEGER NOT NULL,  
    teamID INTEGER NOT NULL,  
    CONSTRAINT CK_TeamID FOREIGN KEY(teamID)  
    REFERENCES Team(teamID),  
    CONSTRAINT CK_MatchID FOREIGN KEY(matchID)  
    REFERENCES Match_(matchID)  
);
```

۱۰. دو موجودیت Team و Stadium:

*رابطه Team و Stadium به صورت has است و هر تیم میتواند یک یا چند استادیوم داشته باشد و بر عکس. پس رابطه many-many است.

**Stadium(ID, Name, CityName, Capacity, ConstructionAmount,
StartConstruction, EndConstruction, ConstructionTime)
StadiumTeam(ID, Team.ID, Stadium.ID)**

Stadium:

```
CREATE TABLE Stadium(  
    stadiumID INTEGER PRIMARY KEY AUTOINCREMENT,  
    StadiumName VARCHAR(255) NOT NULL,  
    stadiumCityName VARCHAR(255) NOT NULL,  
    capacity INTEGER NULL,  
    ConstructionAmount REAL NULL,  
    ConstructionTime REAL NULL,  
    TimeToStartConstruction REAL NULL,  
    EndTimeConstructions REAL NULL,  
    UNIQUE(StadiumName, stadiumCityName)  
);
```

StadiumTeam:

```
CREATE TABLE StadiumTeam(  
    stID INTEGER PRIMARY KEY AUTOINCREMENT,  
    teamID INTEGER NOT NULL,  
    stadiumID NOT NULL,  
    CONSTRAINT FK_TeamID FOREIGN KEY(teamID)  
    REFERENCES Team(teamID),  
    CONSTRAINT FK_stadiumID FOREIGN KEY(stadiumID)  
    REFERENCES Stadium(stadiumID)  
);
```