TEAM YULONG QIAO(YQ2212) & JIANPU MA(JM4437)

HOOP TAVERN

WIKI, SCHEDULE AND RECOMMENDATIONS FOR BASKETBALL FANATICS

Objective

This web application provides the following functions:

1. Team Wiki - User gets access to detailed information about one basketball team by searching, including precise player profile, game schedule, team management and other supporting details
2. Data Analysis - User obtains data analysis & comparison between teams
3. Thoughts Share - User shares thoughts by commenting on a particular game and rating it
4. Recommendation System - User receives recommendations about future games to watch based on algorithms taking user behavior and user identity categories as input

Challenging Part

1. Correct and efficient coding for a comparatively complicated ER diagram
2. Algorithm design and realization of data analysis and recommendation system

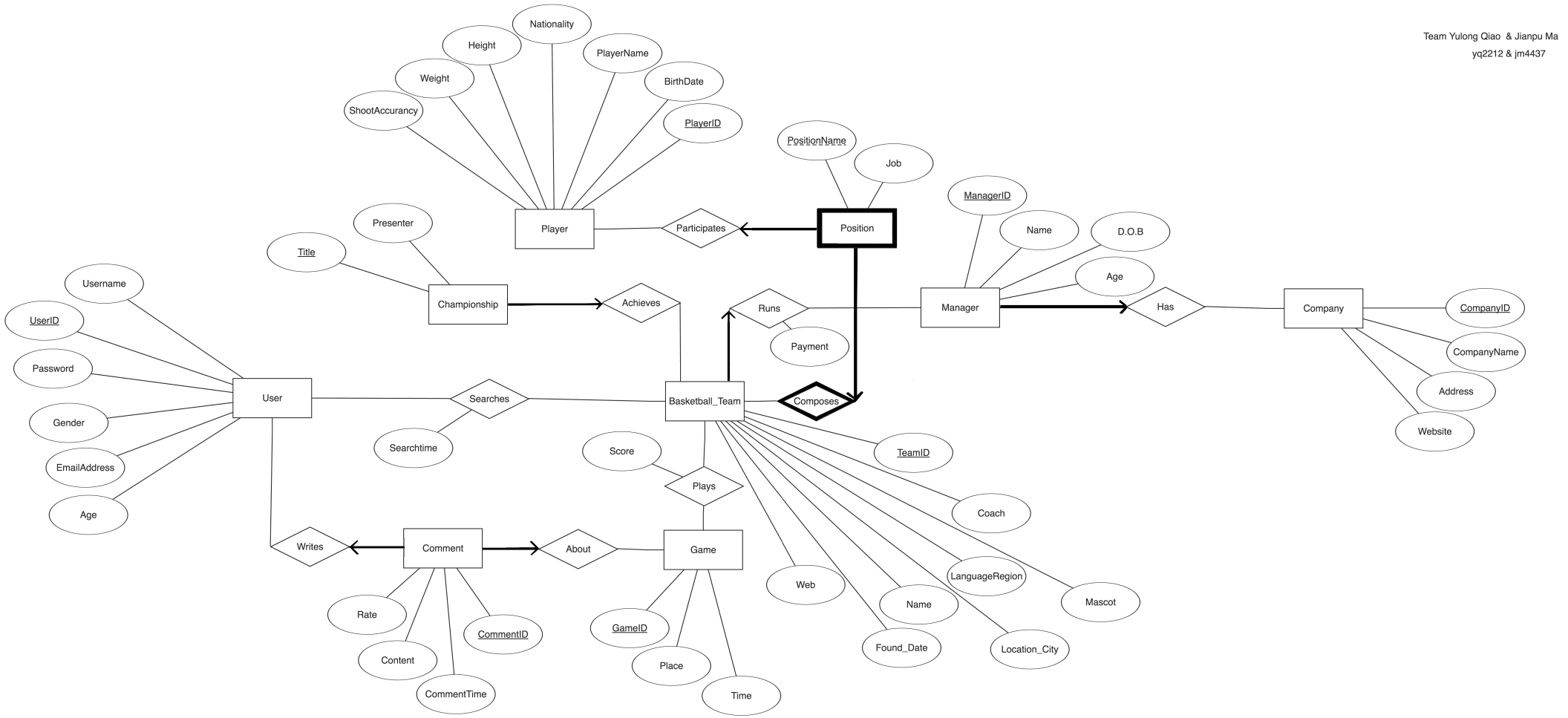
Interesting Part

1. A panoramic display of basketball teams
2. “Enemy-Recommendation”: a user consecutively rating low one team’s games indicates that he/she is a fan of its “enemy team”, thus mutating the recommendation priority
3. Algorithms tracking user’s behavior thus customizing the recommendation

Modify History

1. Set *Web* as *Basketball_Team*’s attribute instead of an entity
2. Remove *Constitutes* relationship between *Player* and *Basketball_Team,* add *PlayerID* as *Player*’s primary key
3. Switch *Payment* from the attribute of *Manager* to the attribute of the *Runs* relationship. For entity *Manager*, add one more attribute *ManagerID* and set it as the only primary key
4. Add *CompanyID* attribute of entity *Company*, and set it as the only primary key; remove attribute *Board*
5. Remove attribute *Searchlist* of relationship *Searches* between *User* and *BasketBall_Team*, add *Searchtime*
6. Reset relationship *Comments* to entity, link it to *User* and *Game* through relationship *Writes* and *About*. Now the attributes of *Comments* are primary key *CommentID*, *CommentTime*, *Rate* and *Content*
7. Reset attribute *Score* of entity *Game* to attribute of relationship *Plays*. Rephrase attribute *GameNumber* of entity *Game* to *GameID*
8. remove attribute *Season* of entity *Championship*, redesign attribute *Coach* of entity *Basketball_Team*

Team Yulong Qiao & Jianpu Ma
yq2212 & jm4437



```
-- SQL Schema of HOOP TAVERN: WIKI, SCHEDULE AND RECOMMENDATIONS FOR BASKETBALL FANATICS
-- Team Yulong Qiao(yq2212) & Jianpu Ma(jm4437)

CREATE TABLE User(
    UserID int,
    Username varchar(50),
    Password varchar(100),
    Gender char(2),
    EmailAddress varchar(100),
    Age int,
    PRIMARY KEY (UserID),
    CHECK(
    Gender='F' or Gender='M'
    )
)

CREATE TABLE Basketball_Team(
    TeamID int,
    Coach varchar(255),
    Mascot varchar(100),
    LanguageRegion varchar(100),
    Location_City varchar(100),
    Name varchar(255),
    Found_Date date,
    Web varchar(255),
    PRIMARY KEY (TeamID)
)

CREATE TABLE Searches(
    UserID int,
    TeamID int,
    Searchtime timestamp,
    PRIMARY KEY (UserID, TeamID),
    FOREIGN KEY (UserID) REFERENCES User,
    FOREIGN KEY (TeamID) REFERENCES Basketball_Team
)

--Merge Writes and Comment into one table
CREATE TABLE Comment_Written(
    UserID int NOT NULL,
    CommentID int,
    CommentTime timestamp,
    Content varchar(255),
    Rate int,
    PRIMARY KEY (CommentID),
    FOREIGN KEY (UserID) REFERENCES User ON DELETE NO ACTION,
    CHECK(
    Rate=1 or Rate=2 or Rate=3 or Rate=4 or Rate=5 or Rate=6 or Rate=7 or Rate=8 or Rate=9 or Rate=10
    )
)

CREATE TABLE Game(
    GameID int,
    Place varchar(255),
    Time timestamp,
    PRIMARY KEY(GameID)
)

CREATE TABLE About(
    GameID int NOT NULL,
    CommentID int,
    PRIMARY KEY (CommentID) REFERENCES Comment_Written,
    FOREIGN KEY (GameID) REFERENCES Game ON DELETE NO ACTION
)

CREATE TABLE Plays(
    GameID int,
    TeamID int,
    Score int,
    PRIMARY KEY (TeamID, GameID),
    FOREIGN KEY (TeamID) REFERENCES Basketball_Team,
    FOREIGN KEY (GameID) REFERENCES Game
)

--Merge Championship and Achieves into one table
CREATE TABLE Championship_Achieved(
    Title varchar(255),
    Presenter varchar(100),
    TeamID int NOT NULL,
    PRIMARY KEY (Title),
    FOREIGN KEY (TeamID) REFERENCES Basketball_Team ON DELETE NO ACTION
)
```

```
CREATE TABLE Player (
    PlayerID int,
    BirthDate date,
    PlayerName varchar(100),
    Nationality varchar(100),
    Height real,
    Weight real,
    ShootAccuracy int,
    PRIMARY KEY (PlayerID),
    CHECK(
    ShootAccuracy < 100 and ShootAccuracy >0 and Height>0 and Weight>0
    )
)

--Write the table for weak entity
CREATE TABLE Position_Composes(
    TeamID int,
    PositionName varchar(50),
    Job varchar(255),
    PRIMARY KEY (TeamID, PositionName),
    FOREIGN KEY (TeamID) REFERENCES Basketball_Team ON DELETE CASCADE
)

CREATE TABLE Participates (
    PlayerID int NOT NULL,
    TeamID int,
    PositionName varchar(50),
    PRIMARY KEY (TeamID, PositionName) REFERENCES Position_Composes,
    FOREIGN KEY (PlayerID) REFERENCES Player ON DELETE NO ACTION
)

--Merge Manager and Has into one table
CREATE TABLE Manager_Has(
    ManagerID int,
    Name varchar(100),
    Age int,
    DateOfBirth date,
    CompanyID int NOT NULL,
    PRIMARY KEY (ManagerID),
    FOREIGN KEY (CompanyID) REFERENCES Company ON DELETE NO ACTION,
    CHECK(
    Age>0
    )
)

CREATE TABLE Runs (
    TeamID int,
    ManagerID int NOT NULL,
    Payment int,
    PRIMARY KEY (TeamID) REFERENCES Basketball_Team,
    FOREIGN KEY (ManagerID) REFERENCES Manager_Has ON DELETE NO ACTION,
    CHECK(
    Payment>0
    )
)

CREATE TABLE Company (
    CompanyID int,
    CompanyName varchar(100),
    Address varchar(255),
    Website varchar(255),
    PRIMARY KEY (CompanyID)
)
```