A Simple Simulation of an NBA (National Basketball Association) Database Application

Team Members:
1. Tianlang Gu (tg1529)
2. Jason Lai (jl9338)

High-level Description: this application aims to simulate a relational database system that records all the relevant data for the NBA, similar to Basketball Reference. Being a professional sports league with the third largest annual revenue in the world of 8.8 billion dollars, there is a plethora of data captured on and off the court of the NBA.

Data: We acquire most of our data through the NBA official site and Basketball Reference. Since most data on the sites listed can be transformed into a CSV file, we can easily import them into the database.

User Interaction: through our database application, users can look up essential statistics regarding each player, each game, and their corresponding arena and basketball team. Users can also explore players' news outside the court and the teams' sponsors, as well as checking other relevant data through the use of custom input.

Entity Sets:
- **Players** (*pid*: serial, *name*: string, *tid*: integer, *dob*: date, jerseyNumber: integer)
- **Teams** (*tid*: serial, *name*: string, *homeCity*: string)
- **Coaches** (*cid*: serial, *coach_name*: string, *tid*: integer, *dob*: date)
- **Arenas** (*aid*: serial, *name*: string, *location*: string)
- **PlayerNews** (*title*: string, link: string)
- **Sponsors** (*sid*: integer, name: string)
- **GameDates** (*gameDate*: date)
- **Referees** (*rid*: serial, *name*: string, *dob*: date)

Relationship Sets:
- **Players_belong_to_teams**: consists of entities **Players** and **Teams**, and relationship belong_to.
- **Coaches_train_teams**: consists of entities **Coaches** and **Teams**, and relationship coached_by.
- **Teams_homed_to_arenas**: consists of entities **Teams** and **Arenas**, and relationship homed_to.
- **Games_hosted_in_arenas:** consists of entities **Game** and **Arenas**, and relationship hosted_in.
- **Games_monitored_by_referees**: consists of entities **Game** and **Referees**, and relationship monitored_by.
- **Game**: entity cluster, composed of **Teams** and **GameDates**, and relationship played_by.

Business Rules:

Players are identified by a *pid*. All players have a *name*, a team they belong to (*tid*), a *date of birth*, and a *jersey number*. No two players have the same combination of *name*, *date of birth*, and *jersey number*.

Player news is identified by a *title* and a *pid.* All player news have a *title* and a *link*. No two player news have the same combination of *title* and *link*.

Teams are identified by a *tid*. All teams have a *name* and a *home city*. No two teams have the same name.

Coaches are identified by a *cid*. All coaches have a *name*, the team they coach for (tid), and a *date of birth*. No two coaches have the same combination of *name* and *team*.

Arenas are identified by an *aid*. All arenas have a *name* and a *location*. No two arenas have the same *name*.

Sponsors are identified by a *sid* and a *tid.* All sponsors have a *name*. No two sponsors have the same name.

GameDates are identified by a *gameDate*. No two game dates have the same date.

Referees are identified by a *rid*. All referees have a *name* and a *dob***.**

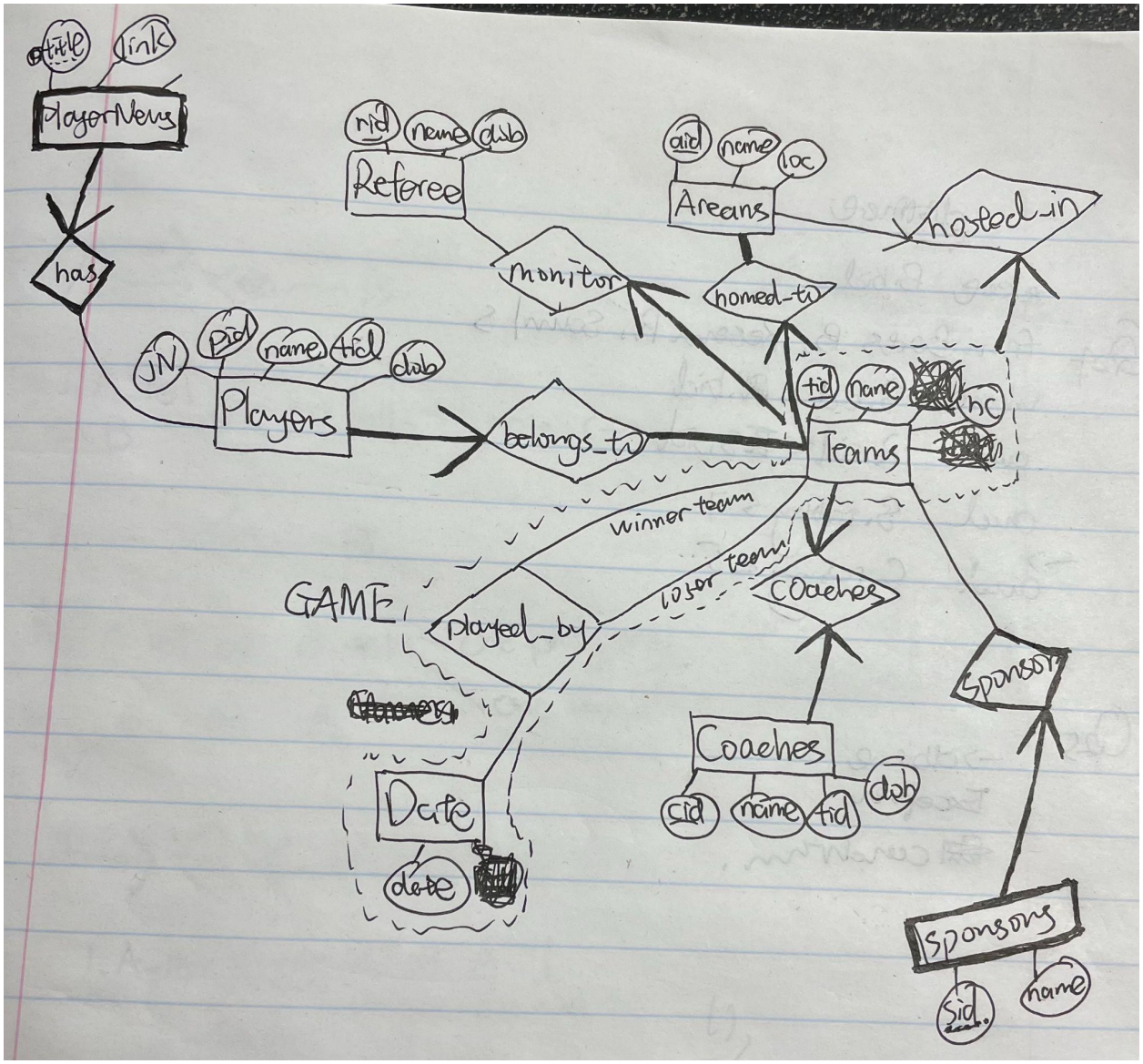
Each team has at least one player and at most 15 players. Each player belongs to exactly one team. Each team is trained by exactly one head coach and each head coach trains exactly one team. Each team resides in exactly one arena. Each arena is homed to at least one team and at most two teams.

Sponsors sponsor teams, and are only included in our database if the team they sponsor is in the database. Each sponsor sponsors exactly one team. Each team can be sponsored by any number of sponsors.

Player news is generated by players, and is only included in our database if the relevant player is recorded in the database. Each player news refers to exactly one player. Each player can have any number of player news.

Each game is played by exactly two teams and has exactly one game date. No two games have the same two participating teams and game date. Each game is monitored by exactly one official referee and is hosted in exactly one arena. Each referee can monitor any number of games. Each arena can host at most one game.

ER Diagram:


title
link
PlayerNews
has
rid
name
club
Referee
aid
name
loc
Areans
hosted_in
monitor
homed_to
JN
pid
name
tid
club
Players
belongs_to
tid
name
hc
Teams
winner team
loser team
GAME
played_by
Coaches
Sponsor
Coaches
sid
name
tid
club
Date
date
Sponsors
sid
name

Schema.sql:

```sql
drop table if exists Teams cascade;
drop table if exists Players_belong_to_teams cascade;
drop table if exists PlayerNews cascade;
drop table if exists Sponsors cascade;
drop table if exists Coaches_train_teams cascade;
drop table if exists Arenas cascade;
drop table if exists Teams_homed_to_arenas cascade;
drop table if exists GameDates cascade;
drop table if exists Game cascade;
drop table if exists Games_hosted_in_arenas cascade;
drop table if exists Referees cascade;
drop table if exists Games_monitored_by_referees cascade;

create table Teams (
    tid serial primary key,
    name varchar(128) unique not null,
    homeCity varchar(128) not null
);

-- Players Teams exactly one TO one or more
create table Players_belong_to_teams (
    pid serial primary key,
    name varchar(128) not null,
    dob date not null,
    jerseyNum integer not null,
    tid integer not null,
    constraint unique_name_dob_jNum unique (name, dob, jerseyNum),
    foreign key (tid) references Teams(tid)
);

-- weak entity
create table PlayerNews (
    pid integer,
    title varchar(128),
    link varchar(256) not null,
    constraint unique_title_link unique (title, link),
    primary key (pid, title),
    foreign key (pid) references Players_belong_to_teams(pid) on delete cascade
);

```

```
41  -- weak entity
42  create table Sponsors (
43      sid integer,
44      tid integer,
45      name varchar(128) unique not null,
46      primary key (sid, tid),
47      foreign key (tid) references Teams(tid) on delete cascade
48  );
49
50  -- Teams Coaches exactly one TO exactly one
51  create table Coaches_train_teams (
52      cid serial primary key,
53      tid integer unique not null,
54      coach_name varchar(128) not null,
55      dob date not null,
56      constraint unique_cName_tid unique (coach_name, tid)
57  );
58
59  create table Arenas (
60      aid serial primary key,
61      name varchar(128) unique not null,
62      location varchar(256) not null
63  );
64
65  -- Teams Arenas exactly one TO at least one and at most two
66  create table Teams_homed_to_arenas (
67      tid integer primary key,
68      aid integer not null,
69      foreign key (aid) references Arenas(aid)
70  );
71
72  create table GameDates (
73      gameDate date primary key
74  );
75
```

```
-- entity cluster, composed of GameDates and Teams entities
create table Game (
    winnerTeamId integer,
    loserTeamId integer,
    gameDate date,
    primary key (winnerTeamId, loserTeamId, gameDate),
    foreign key (winnerTeamId) references Teams(tid),
    foreign key (loserTeamId) references Teams(tid),
    foreign key (gameDate) references GameDates(gameDate)
);

-- Games Arenas exactly one TO at most one
create table Games_hosted_in_arenas (
    winnerTeamId integer,
    loserTeamId integer,
    gameDate date,
    aid integer unique not null,
    primary key (winnerTeamId, loserTeamId, gameDate),
    foreign key (winnerTeamId, loserTeamId, gameDate) references Game(winnerTeamId, loserTeamId, gameDate),
    foreign key (aid) references Arenas(aid)
);

create table Referees (
    rid serial primary key,
    name varchar(128) not null,
    dob date not null
);

-- Games Referees exactly one TO any
create table Games_monitored_by_referees (
    winnerTeamId integer,
    loserTeamId integer,
    gameDate date,
    rid integer not null,
    primary key (winnerTeamId, loserTeamId, gameDate),
    foreign key (winnerTeamId, loserTeamId, gameDate) references Game(winnerTeamId, loserTeamId, gameDate),
    foreign key (rid) references Referees(rid)
);
```