

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

Final Project Report

Application URL:

- <http://128.238.64.108:8501>

Team Members:

- Jason Lai (jl9338)
- Tianlang Gu (tg1529)

Project High-level Description:

- Our application aims to simulate a relational database system that records all the relevant data (players/head coaches/referees/games stats, player news, sponsors, etc.) 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 Acquisition & Loading Procedures:

- We acquired most of our data through the NBA official site and Basketball Reference. In doing so, we first downloaded the data and transformed them into 12 CSV files, then used the *psql copy* command to load all the data into our database.

```
j19338@jedi:~/db-final-project/data$ cat teams.csv | psql -U j19338 -d j19338_db -c "COPY
teams from STDIN CSV HEADER"
COPY 30
j19338@jedi:~/db-final-project/data$ cat players_belong_to_teams.csv | psql -U j19338 -d j19338_db -c "COPY players_belong_to_teams from STDIN CSV HEADER"
COPY 464
j19338@jedi:~/db-final-project/data$ cat playernews.csv | psql -U j19338 -d j19338_db -c "COPY playernews from STDIN CSV HEADER"
COPY 20
j19338@jedi:~/db-final-project/data$ cat sponsors.csv | psql -U j19338 -d j19338_db -c "COPY sponsors from STDIN CSV HEADER"
COPY 38
j19338@jedi:~/db-final-project/data$ cat coaches_train_teams.csv | psql -U j19338 -d j19338_db -c "COPY coaches_train_teams from STDIN CSV HEADER"
COPY 30
j19338@jedi:~/db-final-project/data$ cat arenas.csv | psql -U j19338 -d j19338_db -c "COPY arenas from STDIN CSV HEADER"
COPY 29
j19338@jedi:~/db-final-project/data$ cat teams_homed_to_arenas.csv | psql -U j19338 -d j19338_db -c "COPY teams_homed_to_arenas from STDIN CSV HEADER"
COPY 30
j19338@jedi:~/db-final-project/data$ cat gamedates.csv | psql -U j19338 -d j19338_db -c "COPY gamedates from STDIN CSV HEADER"
COPY 6
j19338@jedi:~/db-final-project/data$ cat game.csv | psql -U j19338 -d j19338_db -c "COPY game from STDIN CSV HEADER"
COPY 43
j19338@jedi:~/db-final-project/data$ cat games_hosted_in_arenas.csv | psql -U j19338 -d j19338_db -c "COPY games_hosted_in_arenas from STDIN CSV HEADER"
COPY 43
j19338@jedi:~/db-final-project/data$ cat referees.csv | psql -U j19338 -d j19338_db -c "COPY referees from STDIN CSV HEADER"
COPY 36
j19338@jedi:~/db-final-project/data$ cat games_monitored_by_referees.csv | psql -U j19338 -d j19338_db -c "COPY games_monitored_by_referees from STDIN CSV HEADER"
COPY 43
```

User Interactions:

- Once arriving at the site, users can first have a quick examination of each table and its corresponding data. Then, users can use the text input box, dropdown menu, or radio button to interact with our application and view the corresponding results to its related question. As listed below, we have a total of 5 questions waiting for users to explore.

- Find the most efficient player (having the highest effective field goal%) on a team
- Find all the games that a referee had officiated between 10/18/22 and 10/23/22
- Find the head coaches that won or lost the most games between 10/18/22 and 10/23/22 (the best/worst coaches in that period)
- Find all the games that were played on a selected date between 10/18/22 and 10/23/22
- Find all the player news and sponsors associated with one team

Entity Sets & Cluster:

- Players(pid: integer, *name*: string, *age*: integer, *position*: string, *tid*: integer, *efg*: decimal)
- Teams(tid: integer, *name*: string, *homeCity*: string)
- Coaches(cid: integer, *tid*: integer, *coachName*: string, *startDate*: date)
- Arenas(aid: integer, *name*: string, *location*: string)
- PlayerNews(title: string, pid: integer, *link*: string)
- Sponsors(sid: integer, tid: integer, *name*: string)
- GameDates(gameDate: date)
- Referees(rid: integer, *name*: string, *yoe*: integer)
- Game(winnerTeamId: integer, loserTeamId: integer, gameDate: 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 entity cluster Game and entity Arenas, and relationship *hosted_in*
- Games_monitored_by_referees: consists of entity cluster Game and entity Referees, and relationship *monitored_by*

Identifying Relationship Sets:

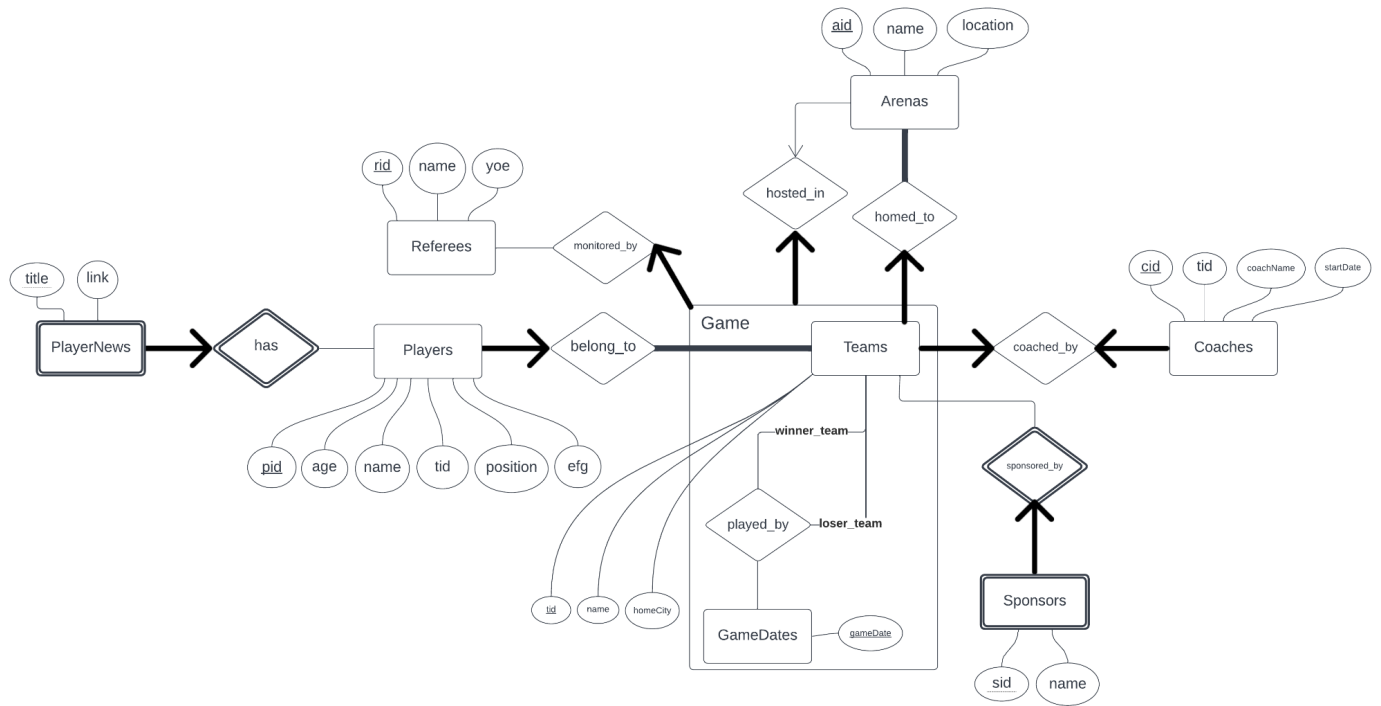
- Sponsors: consists of identifying owner entity Teams and weak entity Sponsors, and relationship *sponsored_by*
- PlayerNews: consists of identifying owner entity Players and weak entity PlayerNews, and relationship *has*

Business Rules:

- Players are identified by a *pid*. All players have a *name*, *age*, playing position (*position*), and a team they belong to (*tid*). Not every player has an effective field goal percentage (*efg*) statistics. No two players have the same combination of *name*, *age*, and *position*.
- 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 *homeCity*. No two teams have the same *name*.
- Coaches (all of them are head coaches) are identified by a *cid*. All coaches have a *coachName*, the team they coach for (*tid*), and the date they began coaching (*startDate*). No two coaches coach for the same team. No two coaches have the same combination of *coachName* and *tid*.
- 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*.
- Referees are identified by a *rid*. All referees have a *name* and a *yoe* (years of experience in officiating basketball games).
- Each team has at least one player and at most 15 players. Each player belongs to exactly one team. Each team is coached 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 two games in one day. However, each arena can host multiple games throughout the seasons.

ER Diagram



Schema.sql

```
1 drop table if exists Teams cascade;
2 drop table if exists Players_belong_to_teams cascade;
3 drop table if exists PlayerNews cascade;
4 drop table if exists Sponsors cascade;
5 drop table if exists Coaches_train_teams cascade;
6 drop table if exists Arenas cascade;
7 drop table if exists Teams_homed_to_arenas cascade;
8 drop table if exists GameDates cascade;
9 drop table if exists Game cascade;
10 drop table if exists Games_hosted_in_arenas cascade;
11 drop table if exists Referees cascade;
12 drop table if exists Games_monitored_by_referees cascade;
13
14 create table Teams (
15     tid integer primary key,
16     name varchar(128) unique not null,
17     homeCity varchar(128) not null
18 );
19
20 create table Players_belong_to_teams (
21     pid integer primary key,
22     name varchar(128) not null,
23     age integer not null,
24     position varchar(128) not null,
25     tid integer not null,
26     efg decimal,
27     constraint unique_name_age_pos unique (name, age, position),
28     foreign key (tid) references Teams(tid)
29 );
30
31 create table PlayerNews (
32     pid integer,
33     title varchar(128) not null,
34     link varchar(256) not null,
35     constraint unique_title_link unique (title, link),
36     primary key (pid, title),
37     foreign key (pid) references Players_belong_to_teams(pid) on delete cascade
38 );
39
```

```
40 create table Sponsors (  
41     sid integer,  
42     tid integer,  
43     name varchar(128) unique not null,  
44     primary key (sid, tid),  
45     foreign key (tid) references Teams(tid) on delete cas  
46 );  
47  
48 create table Coaches_train_teams (  
49     cid integer primary key,  
50     tid integer unique not null,  
51     coachName varchar(128) not null,  
52     startDate date not null,  
53     constraint unique_cName_tid unique (coachName, tid)  
54 );  
55  
56 create table Arenas (  
57     aid integer primary key,  
58     name varchar(128) unique not null,  
59     location varchar(256) not null  
60 );  
61  
62 create table Teams_homed_to_arenas (  
63     tid integer primary key,  
64     aid integer not null,  
65     foreign key (aid) references Arenas(aid)  
66 );  
67  
68 create table GameDates (  
69     gameDate date primary key  
70 );  
71  
72 create table Game (  
73     winnerTeamId integer,  
74     loserTeamId integer,  
75     gameDate date,  
76     primary key (winnerTeamId, loserTeamId, gameDate),  
77     foreign key (winnerTeamId) references Teams(tid),  
78     foreign key (loserTeamId) references Teams(tid),  
79     foreign key (gameDate) references GameDates(gameDate)  
80 );  
81
```

```
82 create table Games_hosted_in_arenas (  
83     winnerTeamId integer,  
84     loserTeamId integer,  
85     gameDate date,  
86     aid integer not null,  
87     primary key (winnerTeamId, loserTeamId, gameDate),  
88     foreign key (winnerTeamId, loserTeamId, gameDate) references Game(winnerTeamId, loserTeamId, gameDate),  
89     foreign key (aid) references Arenas(aid)  
90 );  
91  
92 create table Referees (  
93     rid integer primary key,  
94     name varchar(128) not null,  
95     yoe integer not null  
96 );  
97  
98 create table Games_monitored_by_referees (  
99     winnerTeamId integer,  
100     loserTeamId integer,  
101     gameDate date,  
102     rid integer not null,  
103     primary key (winnerTeamId, loserTeamId, gameDate),  
104     foreign key (winnerTeamId, loserTeamId, gameDate) references Game(winnerTeamId, loserTeamId, gameDate),  
105     foreign key (rid) references Referees(rid)  
106 );
```