CMPT354 – Mini Project 2

Simon Fraser University - Summer 2019

Yiyang Dong

301336929

Bowen Wang

301267523

Brief Overview of Our Mini Project 1:

We create an NBA database for NBA fans to check out information of NBA players and teams. We include the table of players, teams, coaches, sponsors, performance, games, and seasons. There are 9 tables in the database as:

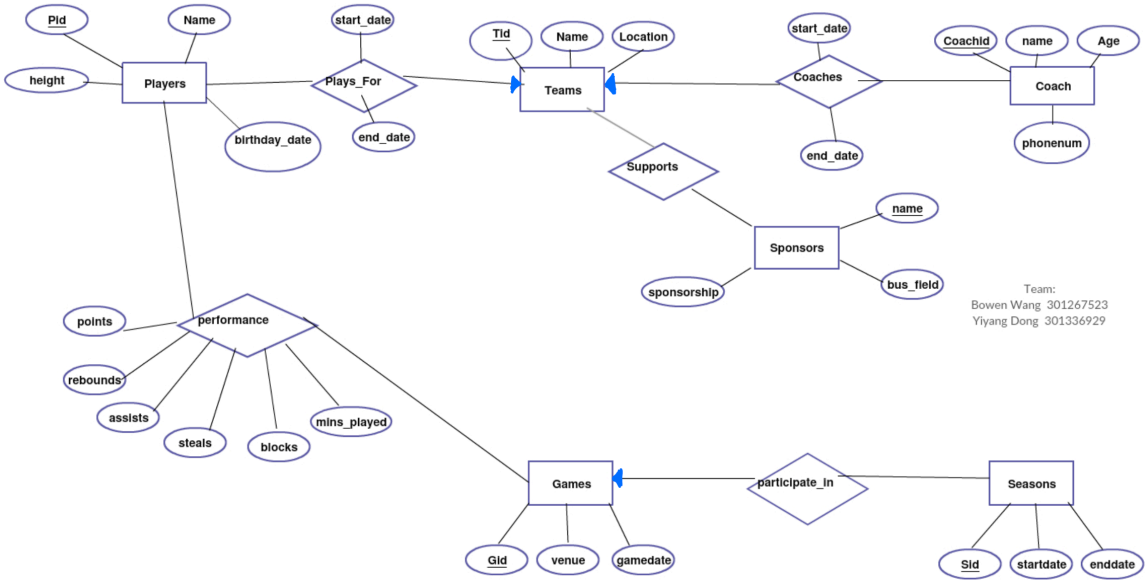
Entity sets:

Players (Pid, height, Name, birthday_date)
Team (Tid, Name, Location)
Coach (Coachid, name, Age, phonenum)
Sponsors (name, sponsorship, bus_field)
Games (Gid, venue, gamedate)
Seasons (Sid, startdate, enddate)

Relationship:

Participate_in (games$^{FK\text{-}games}$, sid$^{FK\text{-}seasons}$) connects games and seasons
Plays_for (pid$^{FK\text{-}players}$, tid$^{FK\text{-}teams}$, start_date, end_date) connects players and teams
Coaches (name$^{FK\text{-}coach}$, tid$^{FK\text{-}team}$, start_date, end_date) connects teams and coach
Performance (pid$^{FK\text{-}players}$, gid$^{FK\text{-}games}$ points, rebounds, assists, steals, blocks, mins_played) connects players and games

E/R Diagram:



Step1: Does your design allow anomalies?

We did a comprehensive review of our design and supposed that there did not occur any serious anomalies. As discussed in class, there are three main anomalies that may happen:

• **Update Anomaly:** one or more instances of duplicated data is updated, but not all.

• **Insertion Anomaly:** certain attributes cannot be inserted into the database without the
presence of other attributes.

• **Deletion Anomaly:** certain attributes are lost because of the deletion of other attributes.

- Coach (Coachid, name, Age, phonenum):
  For convenient, we use C, N, A, P to represent these attributes.
  We record the information of a coach in this entity, the functional dependency is "C => N, A, P" "P=> C, N, A"
  As we assume every coach has only one primary phone number, and in real world scenario phone number has no effect on ages or name or reverse, we think we can ignore the three anomalies in this table.

- Coaches (name$^{FK\text{-}coach}$, tid$^{FK\text{-}team}$, start_date, end_date):
  For convenient, we use N, T, S, E to represent these attributes.
  We record the tart date and end date of the coach works in the team in this entity. The functional dependency is "N, T=>S, E".
  There is no functional dependency between attributes other than keys, since there is only one coach contract happen in a certain period. A given start date would have a specific end date, which indicates there is only one tuple exist with certain start date or end date.
  So, this entity set does not have any anomaly at all.

- Games (Gid, venue, gamedate):
  For convenient, we use Gi, V, Ga to represent these 3 attributes.
  We design each game a unique id and record its venue and gamedate in the same tuple. The functional dependency can be shown as "Gi =>V, Ga".
  There is no functional dependency between attributes rather than primary key (Gi), say, a venue can hold games on various date and several games can be hold on the same day.
  So, this entity set does not have any anomaly at all.

- Players (<u>Pid</u>, name, birthday, height):
  - FDs: Pid --> name, birthday, height

  Analysis: Recall that if two tuples of R agree on all of the attributes A_1, A_2, …, An, then they must also agree on all of another list of attributes B_1, B_2, … Bm, then A_1, A_2, …, A_n functionally determine B_1, B_2, …, Bm. We know that each player has unique Pid, which means there are no two players have same Pid. Therefore, it is easy to see that Pid functionally determines all other attributes (the player's name, birthday and height). To prove Players do not allow anomalies, we first confirm that this table is not redundant, every player who get into the NBA exactly is registered with his unique Pid, Pid represents who the player is and player's name, birthday and height. Therefore, this **<u>cannot cause any redundancy</u>**. Then, if a player changes his name or grows taller and we want to update related information for each player, then there is no same information unchanged for any other tuples in table, so it **<u>does not allow Update Anomalies</u>.** Last but not least, each of these attributes is independent of others, so it is **<u>not Deletion Anomalies.</u>**

- Teams (<u>Tid</u>, name, location):
  - FDs: Tid --> name, location

  Analysis: this functional dependency is similar to Players, Tid functionally dependencies team's name and its location. Since it is an important but not complex relation, it can't lead to redundant. Moreover, there is no relationship between team name and location since many teams may locate at same city. Therefore, there is no anomalies in this entity.

- Plays_for (Pid, Tid, start_date, end_date):
  - FDs: Pid, Tid --> start_date, end_date

  Analysis: In Common sense, player and team is kind of "one-on-one" relationship with contract. Therefore, Pid and Tid functionally determine start_date and end_date. And it does not allow each of three anomalies.

- Performance ($pid^{FK\text{-}players}$, $gid^{FK\text{-}games}$ points, rebounds, assists, steals, blocks, mins_played):

  For convenient, we use Pid, Gid, P, R, A, S, B, M to represent these attributes.

  We record the performance of each player into this entity, to check how well a player plays in a game. the functional dependency is Pid, "Gid => P, R, A, S, B, M"

  There is no functional dependency between attributes as every column is independent to others. Say, how much rebounds a player gets in a game has no effect on how much blocks he can get.

So, this entity set does not have any anomaly at all.

- Seasons (Sid, startdate, enddate):
  For convenient, we use Si, St, E to represent these attributes.
  We record the season start date and end date in this entity. The functional dependency can be shown as "Si => St, E".
  There is no functional dependency between attributes rather than primary key, since there is only one season happen in a certain period. A given start date would have a specific end date, which indicates there is only one tuple exist with certain start date or end date.
  So, this entity set does not have any anomaly at all.

- Sponsor (name, sponsorship, bus_field):
  For convenient, we use N, S, B to represent these attributes.
  We record the sponsor's company name, amount, and its business field. The functional dependency can be shown as "N => S, B".
  There is no functional dependency between attributes rather than primary key, since there is no certain relationship between S and B, how much money a company sponsor a team does not relates to what field it operates in.
  So, this entity set does not have any anomaly at all.

Step 2: SQL Schema

As we don't have any change on our model, we keep the same as original

Entity sets:

Players (Pid, height, Name, birthday_date)
Team (Tid, Name, Location)
Coach (Coachid, name, Age, phonenum)
Sponsors (name, sponsorship, bus_field)
Games (Gid, venue, gamedate)
Seasons (Sid, startdate, enddate)

Relationship:

Participate_in (games$^{FK\text{-}games}$, sid$^{FK\text{-}seasons}$) connects games and seasons
Plays_for (pid$^{FK\text{-}players}$, tid$^{FK\text{-}teams}$, start_date, end_date)
connects players and teams
Coaches (name$^{FK\text{-}coach}$, tid$^{FK\text{-}team}$, start_date, end_date)
connects teams and coach
Performance (pid$^{FK\text{-}players}$, gid$^{FK\text{-}games}$ points, rebounds, assists, steals, blocks, mins_played) connects players and games

Step 3: Populate Tables

As we don't have change on our model, we

```
create table Players (
      Pid integer,
        name char(30),
        birthday date default '0000-00-00',
        height float,
        primary key (Pid),
        foreign key (Pid) references Plays_for (Pid)    on delete restrict,
        CHECK (height > 0)
)

create table Teams (
        Tid int,
        name char(30),
        location char(30),
        primary key (Tid),
        foreign key (Tid) references Plays_for (Tid) on delete restrict,
        foreign key (Tid) references Coaches(Tid)    on delete restrict
)

create table Plays_for (
        Pid integer,
        Tid integer,
        start_date date default '0000-00-00',
        end_date date default '0000-00-00',
        check (start_date <= end_date)
)
```

```
create table Coach (
	Coachid int,
	name char(30),
	age int,
	phonenum char default '0000000000',
	primary key (Coachid),
	foreign key (Coachid) references Coaches(Coachid)    on delete restrict
)

create table Coaches (
	Coachid integer,
	Tid integer,
	start_date date default '0000-00-00',
	end_date date default '0000-00-00',
	check (start_date <= end_date)
)

create table Games (
	Gid int,
	Sid int,
	venue char(30),
	gamedate date default '0000-00-00',
	primary key (Gid)
)
```

```
create table Performance (
	Pid    integer,
	Gid integer,
	points int,
	rebounds int,
	assists int,
	steals int,
	blocks int,
	mins_played int,
	primary key (Pid, Gid),
	foreign key (Pid) references Players(Pid)    on delete restrict,
	foreign key(Gid) references Games(Gid)    on delete restrict,
	check ( points >= 0),
	check (rebounds >= 0),
	check (assists >= 0),
	check (steals >= 0),
	check (blocks >= 0),
	check (mins_played >= 0)
)

create table Seasons (
	Sid int,
	startdate date default '0000-00-00',
	enddate date default '0000-00-00',
	primary key(Sid),
	foreign key (Sid) references Games(Sid) on delete restrict,
	check (startdate < enddate)
)

create table Sponsor (
	name char(30),
	Tid int,
	bus_field char(20),
	sponsorship int,
	primary key (name),
	foreign key (Tid) references Teams(Tid) on delete restrict,
	check (sponsorship >= 0)
)
```

```
INSERT INTO Players
    (Pid, name, birthday, height)
VALUES
    (1, 'a', '1979-08-05', 2.03),
    (2, 'b', '1979-08-05', 2.03),
    (3, 'c', '1979-08-05', 2.03),
    (4, 'd', '1979-08-05', 2.03),
    (5, 'e', '1979-08-05', 2.03),
    (6, 'f', '1979-08-05', 2.03),
    (7, 'g', '1979-08-05', 2.03),
    (8, 'h', '1979-08-05', 2.03),
    (9, 'i', '1979-08-05', 2.03),
    (10, 'j', '1979-08-05', 2.03);

INSERT INTO Coach
    (Coachid, name, age, phonenum)
VALUES
    (1001, 'a', 37, '1234432112'),
    (1002, 'b', 37, '1234432113'),
    (1003,   'c', 37, '1234432114'),
    (1004,   'd', 37, '1234432115'),
    (1005,   'e', 37, '1234432116'),
    (1006,   'f', 37, '1234432117'),
    (1007,   'g', 37, '1234432118'),
    (1008,   'h', 37, '1234432119'),
    (1009,   'i', 37, '1234432110'),
    (1010,   'j', 37, '1234432111'),
    (1011,   'k', 37, '1234432121'),
    (1012,   'l', 37, '1234432122');
```

```
INSERT INTO Teams
    (Tid, name, location)
VALUES
    (1001, 'Lakers', 'LA'),
    (1002, 'Ducks', 'BJ'),
    (1003, 'Rockets', 'H'),
    (1004, 'Warriors', 'GS'),
    (1005, 'Bulls', 'CH'),
    (1006, 'Heat', 'Miami'),
    (1007, 'Raptors', 'Tor'),
    (1008, 'Nets', 'Brooklyn'),
    (1009, 'Knicks', 'NY'),
    (1010, 'Bucks', 'Mil');


INSERT INTO Coaches
    (Coachid, Tid, start_date, end_date)
VALUES
    (1001, 1001, '2010-01-01', '2011-01-01'),
    (1002, 1002, '2010-01-01', '2011-01-01'),
    (1003, 1003, '2010-01-01', '2011-01-01'),
    (1004, 1004, '2010-01-01', '2011-01-01'),
    (1005, 1005, '2010-01-01', '2011-01-01'),
    (1006, 1006, '2010-01-01', '2011-01-01'),
    (1007, 1007, '2010-01-01', '2011-01-01'),
    (1008, 1008, '2010-01-01', '2011-01-01'),
    (1009, 1009, '2010-01-01', '2011-01-01'),
    (1010, 1010, '2010-01-01', '2011-01-01');
```

```
INSERT INTO Plays_for
    (Pid, Tid, start_date, end_date)
VALUES
    (1, 1001, '2010-01-01', '2011-01-01'),
    (2, 1002, '2010-01-01', '2011-01-01'),
    (3, 1003, '2010-01-01', '2011-01-01'),
    (4, 1004, '2010-01-01', '2011-01-01'),
    (5, 1005, '2010-01-01', '2011-01-01'),
    (6, 1006, '2010-01-01', '2011-01-01'),
    (7, 1007, '2010-01-01', '2011-01-01'),
    (8, 1008, '2010-01-01', '2011-01-01'),
    (9, 1009, '2010-01-01', '2011-01-01'),
    (10, 1010, '2010-01-01', '2011-01-01');

INSERT INTO Seasons
    (Sid, startdate, enddate)
VALUES
    (1, '2010-01-01','2011-01-01'),
    (2, '2009-01-01','2010-01-01'),
    (3, '2008-01-01','2009-01-01'),
    (4, '2007-01-01','2008-01-01'),
    (5, '2006-01-01','2007-01-01'),
    (6, '2005-01-01','2006-01-01'),
    (7, '2004-01-01','2005-01-01'),
    (8, '2003-01-01','2004-01-01'),
    (9, '2002-01-01','2003-01-01'),
    (10, '2001-01-01','2002-01-01');

INSERT INTO Games
    (Gid, venue, gamedate, Sid)
VALUES
    (1,'a', '2010-03-05', 1),
    (2,'b', '2010-03-06', 1),
    (3,'c', '2010-03-07', 1),
    (4,'d', '2010-03-08', 1),
    (5,'e', '2010-03-09', 1),
    (6,'f', '2010-03-10', 1),
    (7,'g', '2010-03-11', 1),
    (8,'h', '2010-03-12', 1),
    (9,'j', '2010-03-13', 1),
    (10,'k', '2010-03-14', 1);
```

```
INSERT INTO Performance
    (Pid, Gid, points, rebounds, assists, steals, blocks, mins_played)
VALUES
    (1, 1, 19, 4, 5, 1, 2, 30),
    (2, 2, 19, 4, 5, 1, 2, 30),
    (3, 3, 19, 4, 5, 1, 2, 30),
    (4, 4, 19, 4, 5, 1, 2, 30),
    (5, 5, 19, 4, 5, 1, 2, 30),
    (6, 6, 19, 4, 5, 1, 2, 30),
    (7, 7, 19, 4, 5, 1, 2, 30),
    (8, 8, 19, 4, 5, 1, 2, 30),
    (9, 9, 19, 4, 5, 1, 2, 30),
    (10, 10, 19, 4, 5, 1, 2, 30);

INSERT INTO Sponsor
    (name, Tid, bus_field, sponsorship)
VALUES
    ('a', 1001, 'q', 900),
    ('b', 1002, 'w', 900),
    ('c', 1003, 'e', 900),
    ('d', 1004, 'r', 900),
    ('e', 1005, 't', 900),
    ('f', 1006, 'y', 900),
    ('g', 1007, 'u', 900),
    ('h', 1008, 'i', 900),
    ('i', 1009, 'o', 900),
    ('j', 1010, 'p', 900);
```