NBA User Platform

Wenbin Xuan wx2253 & Angel He ah3354   Postgresql: ah3354

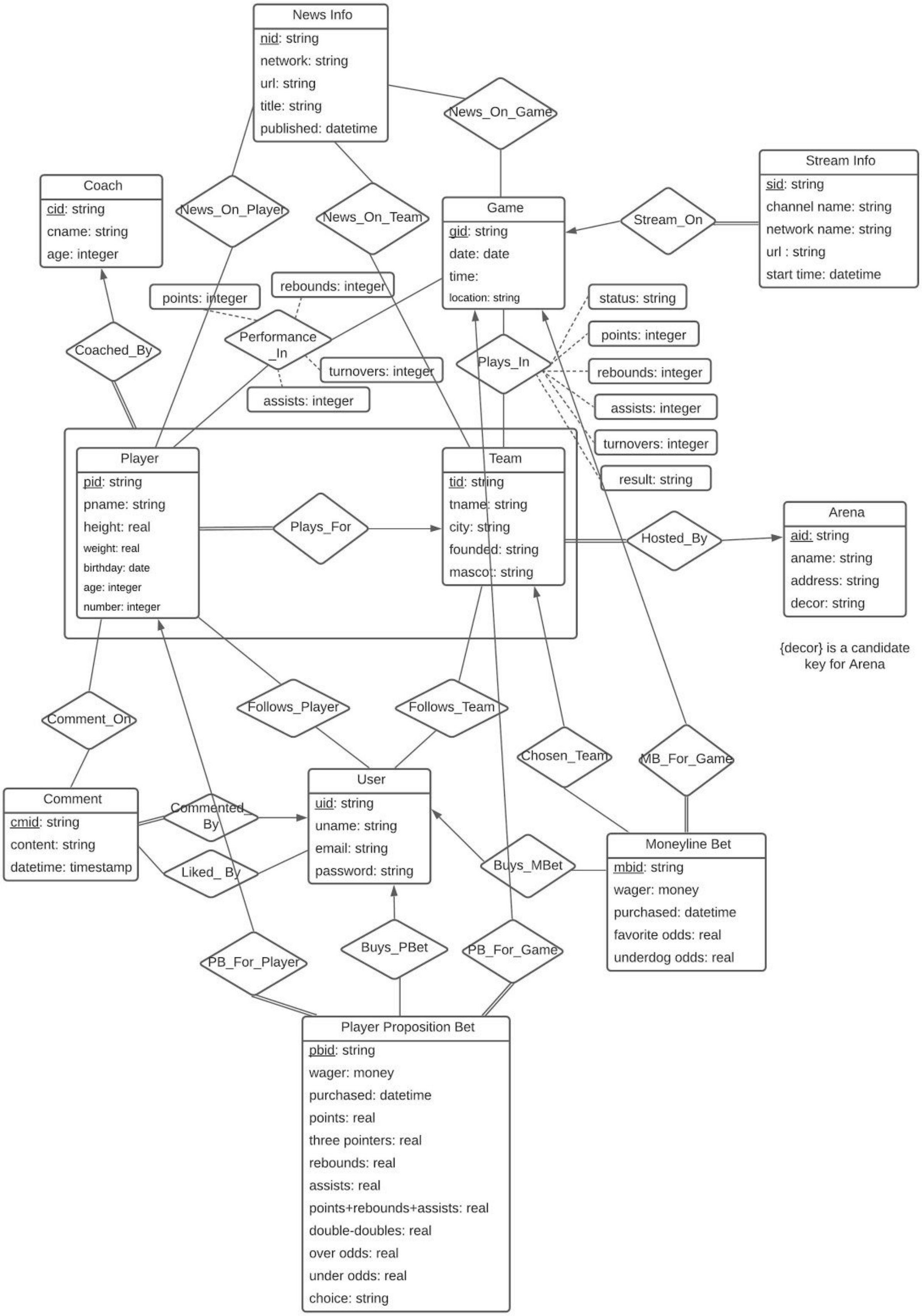
1.A textual description of your extensions on the database design, explaining which entity sets and relationship sets are new, and how you mapped them to SQL statements.

As shown in our ER diagram, we added the Stream_Info entity and connected it to our current Game entity via the new relation Stream_On. This would allow us to store and display information on where a particular game can be streamed. We also added the News_Info entity for storing the relevant information of news articles about a player, team, or game, so we have three corresponding relations: News_On_Player, News_On_Team, and News_On_Game. This would let us organize the news entries and link the relevant ones to a player, team, or game, so the user can easily find the news they are looking for. Also new are Follors_Player and Follows_Team relations connecting User to either Player or Team, so the user can follow their favorite players and teams and the website can make the master News section more relevant for the user. Lastly, we added two NBA betting options for the user, which are captured in the entities Moneyline_Bet and Player_Proposition_Bet and all the relations connected to them. Each Moneyline Bet corresponds to exactly one game and at most one user can purchase it, choosing the one team they think will win that game. Each Player_Proposition Bet corresponds to exactly one game and one player in that game, and will specify a value for one and only one of points, three-pointers, rebounds, assists, points+rebounds+assists, and double-doubles, along with the over or under (or both) odds of it. Each Player Proposition Bet can be purchased by at most one user and the user can either bet over or under.

Several changes were made when we mapped our ER-diagram into SQL. Stream is simply a 1-1 constraint, which is the easiest part. For News_info, we firstly tried to create an n-ary format relation. However, there are some overlapped-types news leading to some relation attributes being null, while the others not. This violates the primary key's definition. Therefore, we splitted the relation into several different types of new_info.

The hardest part is mapping two Bet tables into SQL. It's pretty straightforward to connect users to bet tables(purchase lotteries). However, to limit the Moneyline bet into one specific game and correctly illustrate the two corresponding teams is a bit more confusing. As can be seen in our ER-diagram, we drew the bets table by lining them with three different relations. This caused difficulties in expressing some realistic constraints, such as a user can only bet on the teams which will play in that game. This constraint can't be shown with a simple check constraint. Therefore, we decided to reference Plays_In table in Moneyline bet and make (tid, gid) the foreign key. Similarly, we had to limit the player in the player proposition bet to the game they participated in, so we referenced Performance_In table in Player proposition bet and made (pid,gid) the foreign key. Now these real world constraints should hold over the bets.

2. ER Diagram



{decor} is a candidate key for Arena

3. SQL Schema

```
✔️CREATE TABLE Player-Plays_For-Coached_By  — aggregation & 1-1 constraint
(pid VARCHAR(20),
 pname VARCHAR(50) NOT NULL,
 height REAL CHECK (height >0),
 weight REAL CHECK (weight >0),
 birthday DATE CHECK (birthday <= 2001-10-25),
 age INTEGER CHECK (age >= 19),
 number INTEGER CHECK (number >= 0),
 tid VARCHAR(20) NOT NULL,
 cid VARCHAR(20) NOT NULL,
 PRIMARY KEY (pid)
 FOREIGN KEY (tid) REFERENCES Team-Hosted_By
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
 FOREIGN KEY (cid) REFERENCES Coach
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
```

```
✔️CREATE TABLE Team-Hosted_By              inserted: chicago bulls (tid = aid = 001), miami
heat (tid=aid=002),
            (tid VARCHAR(20),
             tname VARCHAR(50) NOT NULL,
             city VARCHAR(30),
             founded VARCHAR(50),
             mascot VARCHAR(30),
             aid VARCHAR(20) NOT NULL,
             PRIMARY KEY (tid),
             FOREIGN KEY (aid) REFERENCES Arena,
                 ON DELETE NO ACTION
                 ON UPDATE NO ACTION)
```

```
✔️ CREATE TABLE Arena                          -- needed for team-hosted_by
            (aid VARCHAR(20),
             aname VARCHAR(50) NOT NULL,
             address VARCHAR(100),
             decor VARCHAR(20),
             PRIMARY KEY (aid),
             UNIQUE (decor))
```

```
✓ CREATE TABLE Coach              -- needed for coached_by
      (cid VARCHAR(20),
       cname VARCHAR(50) NOT NULL,
       age INTEGER CHECK (age >= 21),
       PRIMARY KEY (cid))


✓ CREATE TABLE Game
      (gid VARCHAR(20),
       date DATE NOT NULL CHECK (date >= 2019-10-22),
       time VARCHAR(20),
       location VARCHAR(50),
       PRIMARY KEY (gid),
       FOREIGN KEY (location) REFERENCES Arena
              ON DELETE NO ACTION,
              ON UPDATE NO ACTION)



✓CREATE TABLE Plays_In
      (tid VARCHAR(20),
       gid VARCHAR(20),
       status VARCHAR(5),
       points REAL CHECK (points >= 0),
       rebounds REAL CHECK(rebounds >= 0),
       assists REAL CHECK (assists >= 0),
       turnovers REAL CHECK (turnovers >= 0),
       PRIMARY KEY (tid,gid),
       FOREIGN KEY (tid) REFERENCES Team-Hosted_By
              ON DELETE NO ACTION
              ON UPDATE NO ACTION,
       FOREIGN KEY (gid) REFERENCES Game
              ON DELETE NO ACTION
              ON UPDATE NO ACTION,
       CHECK (status IN (‘home’,’away’,’Home’,’Away’)) )


✓CREATE TABLE Performance_In
      (pid VARCHAR(20),
       gid VARCHAR(20),
       points REAL CHECK (points >= 0),
       rebounds REAL CHECK(rebounds >= 0),
       assists REAL CHECK(assists >= 0),
       turnovers REAL CHECK(turnovers >= 0),
       PRIMARY KEY (pid, gid),
       FOREIGN KEY (pid) REFERENCES Player-Plays_For-Coached_By
           ON DELETE NO ACTION
           ON UPDATE NO ACTION,
       FOREIGN KEY (gid) REFERENCES Game
           ON DELETE NO ACTION
           ON UPDATE NO ACTION)


✓ CREATE TABLE User   5 tuples inserted
      (uid VARCHAR(20),
       uname VARCHAR(50) NOT NULL,
       email VARCHAR(50) NOT NULL,
       password VARCHAR(30) NOT NULL,
       PRIMARY KEY (uid))


✓ CREATE TABLE Comment-Commented_By
      (cmid VARCHAR(20),
       content VARCHAR(2000),
       datetime TIMESTAMP WITH TIME ZONE CHECK (datetime <= NOW() ),
       uid VARCHAR(20) NOT NULL,
       PRIMARY KEY (cmid),
       FOREIGN KEY (uid) REFERENCES User
              ON DELETE NO ACTION
              ON UPDATE NO ACTION)



✓CREATE TABLE Comment_On
      (cmid VARCHAR(20),
       pid VARCHAR(20),
       PRIMARY KEY (cmid,pid),
       FOREIGN KEY (pid) REFERENCES Player-Plays_For-Coached_By
              ON DELETE NO ACTION
              ON UPDATE NO ACTION,
       FOREIGN KEY (cmid) REFERENCES Comment-Commented_By
              ON UPDATE CASCADE
              ON DELETE CASCADE)


✓CREATE TABLE Liked_By
      (cmid VARCHAR(20),
       uid VARCHAR(20),
PRIMARY KEY (cmid,uid),
FOREIGN KEY (uid) REFERENCES User
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
FOREIGN KEY (cmid) REFERENCES Comment-Commented_By
        ON UPDATE CASCADE
        ON DELETE CASCADE)
```

Part3 TABLE

```
👍✔️CREATE TABLE News info
      (nid VARCHAR(20),
       network VARCHAR(100) NOT NULL,
       url VARCHAR(1000),
       title VARCHAR(300) NOT NULL,
       published TIMESTAMP CHECK (published >= ‘2019-10-22 00:00:00’),
       PRIMARY KEY (nid))
```

```
👍✔️CREATE TABLE News_On_Game
      (nid VARCHAR(20),
       gid VARCHAR(20),
       PRIMARY KEY(nid,gid),
       FOREIGN KEY (nid) REFERENCES News info
                ON UPDATE CASCADE
                ON DELETE CASCADE,
       FOREIGN KEY (gid) REFERENCES Game
                ON DELETE NO ACTION
                ON UPDATE NO ACTION)
```

```
👍✔️CREATE TABLE News_On_Team
      (nid VARCHAR(20),
       tid VARCHAR(20),
       PRIMARY KEY(nid,tid),
       FOREIGN KEY (nid) REFERENCES News info
                ON UPDATE CASCADE
                ON DELETE CASCADE,
       FOREIGN KEY (tid) REFERENCES Team-Hosted_By
                ON DELETE NO ACTION
                ON UPDATE NO ACTION)
```

```
👍✔️CREATE TABLE News_On_Player
```

```
     (nid VARCHAR(20),
      pid VARCHAR(20),
      PRIMARY KEY(nid,pid),
      FOREIGN KEY (nid) REFERENCES News info
              ON UPDATE CASCADE
              ON DELETE CASCADE,
      FOREIGN KEY (pid) REFERENCES Player-Plays_For-Coached_By
              ON DELETE NO ACTION
              ON UPDATE NO ACTION)


👍✔️CREATE TABLE Stream info_Stream_On
     (sid VARCHAR(20),
      channel name VARCHAR(20),
      network name VARCHAR(20),
      url VARCHAR(500),
      start time TIMESTAMP CHECK (date >= ‘2019-10-22 00:00:00’),
      gid VARCHAR(20) NOT NULL,
      PRIMARY KEY (sid),
      FOREIGN KEY (gid) REFERENCES Game
             ON UPDATE CASCADE
             ON DELETE CASCADE)


👍✔️CREATE TABLE Follows_Player
     (uid VARCHAR(20),
      pid VARCHAR(20),
      PRIMARY KEY (uid,pid),
      FOREIGN KEY (uid) REFERENCES User
             ON UPDATE CASCADE
             ON DELETE CASCADE),
      FOREIGN KEY (pid) REFERENCES Player-Plays_For-Coached_By
             ON UPDATE NO ACTION
             ON DELETE NO ACTION)


👍✔️CREATE TABLE Follows_Team
     (uid VARCHAR(20),
      tid VARCHAR(20),
      PRIMARY KEY (uid,tid),
      FOREIGN KEY (uid) REFERENCES User
             ON UPDATE CASCADE
             ON DELETE CASCADE),
      FOREIGN KEY (tid) REFERENCES Team-Hosted_By
```

```
                    ON UPDATE NO ACTION
                    ON DELETE NO ACTION)


👍✔️CRATE TABLE Player Proposition Bet
            (pbid VARCHAR(20),
             wager MONEY CHECK (wager > ‘0’),
             purchased TIMESTAMP CHECK (purchased >= ‘2019-10-22 00:00:00’),
            points REAL CHECK (points > 0),
            three_pointers REAL CHECK (three pointers > 0),
            rebounds REAL CHECK (rebounds > 0),
            assists REAL CHECK (assists > 0),
            points_rebounds_assists REAL CHECK ( points_rebounds_assists > 0),
            double_doubles REAL CHECK (double_doubles > 0),
            over_odds REAL,
            under_odds REAL,
            choice VARCHAR(5),
            pid VARCHAR(20) NOT NULL,
            gid VARCHAR(20) NOT NULL,
            uid VARCHAR(20),
            PRIMARY KEY (pbid),
            FOREIGN KEY (pid,gid) REFERENCES Performance_In
                    ON DELETE CASCADE
                    ON UPDATE CASCADE,
             FOREIGN KEY (uid) REFERENCES User
                    ON DELETE NO ACTION
                    ON UPDATE NO ACTION,
            CHECK (choice IN (‘Over’,’over’,’Under’,’under’)),
            CHECK (
                    (over_odds IS NOT NULL OR under_odds IS NOT NULL)
                    AND
                    (  (Points IS NOT NULL AND three_pointers IS NULL AND rebounds IS
NULL AND assists IS NULL AND points_rebounds_assists IS NULL AND
double_doubles IS NULL)
                    OR (Points IS NULL AND three_pointers IS NOT NULL AND rebounds IS
NULL AND assists IS NULL AND points_rebounds_assists IS NULL AND
double_doubles IS NULL)
            OR (Points IS NULL AND three_pointers IS NULL AND rebounds IS NOT NULL
AND assists IS NULL AND points_rebounds_assists IS NULL AND double_doubles IS
NULL)
            OR (Points IS NULL AND three_pointers IS NULL AND rebounds IS NULL AND
assists IS NOT NULL AND points_rebounds_assists IS NULL AND double_doubles IS
NULL)
            OR (Points IS NULL AND three_pointers IS NULL AND rebounds IS NULL AND
assists IS NULL AND points_rebounds_assists IS NOT NULL AND double_doubles IS
NULL)
```

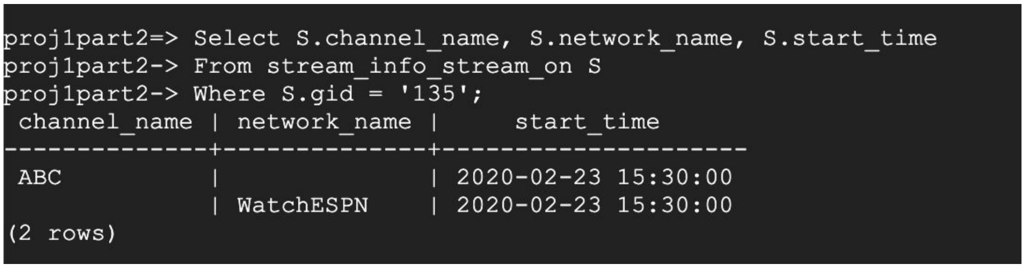
```
        OR (Points IS NULL AND three_pointers IS NULL AND rebounds IS NULL AND assists IS NULL AND points_rebounds_assists IS NULL AND double_doubles = 0.5)
        )
        )
)
```

```
👍✔️CREATE TABLE Moneyline Bet
        (mbid VARCHAR(20),
         wager MONEY CHECK (wager > ‘0’),
         purchased TIMESTAMP CHECK (purchased >= ‘2019-10-22 00:00:00’),
         favorite_team VARCHAR(20) NOT NULL,
         favorite_odds REAL NOT NULL,
        underdog_team VARCHAR(20) NOT NULL,
         underdog_odds REAL NOT NULL,
         gid VARCHAR(20) NOT NULL,
         uid VARCHAR(20),
        chosen_team VARCHAR(20),
         PRIMARY KEY (mbid),
         FOREIGN KEY (favorite_team,gid) REFERENCES Plays_In
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (underdog_team,gid) REFERENCES Plays_In
                ON DELETE CASCADE
                ON UPDATE CASCADE,
         FOREIGN KEY (uid) REFERENCES User
                ON DELETE NO ACTION
                ON UPDATE NO ACTION
        CHECK ( (chosen_team = favorite_team) OR (chosen_team = underdog_team) )
        )
```

4. Queries

Query 1: find all the ways to stream the 2/23 game LA Lakers vs. Boston Celtics (gid = 135), return the channel/network name and start time

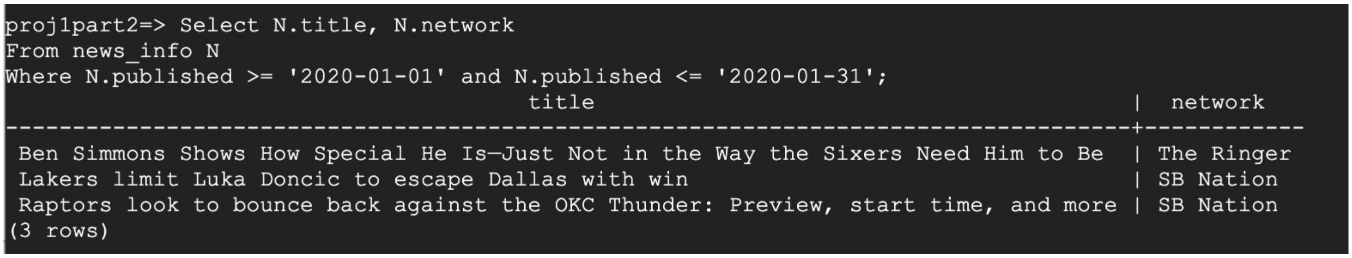
Select S.channel_name, S.network_name, S.start_time
From stream_info_stream_on S
Where S.gid = '135';



```
proj1part2=> Select S.channel_name, S.network_name, S.start_time
proj1part2-> From stream_info_stream_on S
proj1part2-> Where S.gid = '135';
 channel_name | network_name |     start_time
--------------+--------------+---------------------
 ABC          |              | 2020-02-23 15:30:00
              | WatchESPN    | 2020-02-23 15:30:00
(2 rows)
```

Query 2: find all the news entries published in January 2020, return their titles and news networks

Select N.title, N.network
From news_info N
Where N.published >= ‘2020-01-01’ and N.published <= ‘2020-01-31’

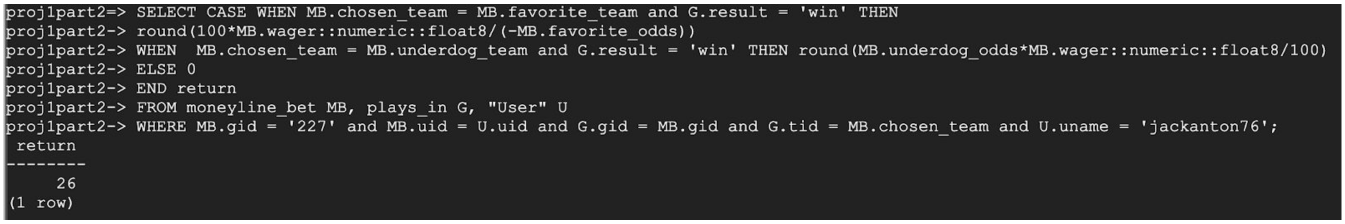


```
proj1part2=> Select N.title, N.network
From news_info N
Where N.published >= '2020-01-01' and N.published <= '2020-01-31';
                                         title                                     |  network
-----------------------------------------------------------------------------------+------------
 Ben Simmons Shows How Special He Is—Just Not in the Way the Sixers Need Him to Be  | The Ringer
 Lakers limit Luka Doncic to escape Dallas with win                                 | SB Nation
 Raptors look to bounce back against the OKC Thunder: Preview, start time, and more | SB Nation
(3 rows)
```

Query 3: find the return value for ‘jackanton76’(uname) in game 2/28/2020 Mavs vs Heat(gid = ‘227’) in his moneyline lottery.

SELECT CASE WHEN MB.chosen_team = MB.favorite_team and G.result = 'win' THEN round(100*MB.wager::numeric::float8/(-MB.favorite_odds))
WHEN MB.chosen_team = MB.underdog_team and G.result = 'win' THEN round(MB.underdog_odds*MB.wager::numeric::float8/100)
ELSE 0
END return
FROM moneyline_bet MB, plays_in G, User U

WHERE MB.gid = ‘227’ and MB.uid = U.uid and G.gid = MB.gid and G.tid = MB.chosen_team and U.uname = ‘jackanton76’;



```
proj1part2=> SELECT CASE WHEN MB.chosen_team = MB.favorite_team and G.result = 'win' THEN
proj1part2-> round(100*MB.wager::numeric::float8/(-MB.favorite_odds))
proj1part2-> WHEN  MB.chosen_team = MB.underdog_team and G.result = 'win' THEN round(MB.underdog_odds*MB.wager::numeric::float8/100)
proj1part2-> ELSE 0
proj1part2-> END return
proj1part2-> FROM moneyline_bet MB, plays_in G, "User" U
proj1part2-> WHERE MB.gid = '227' and MB.uid = U.uid and G.gid = MB.gid and G.tid = MB.chosen_team and U.uname = 'jackanton76';
 return
--------
     26
(1 row)
```