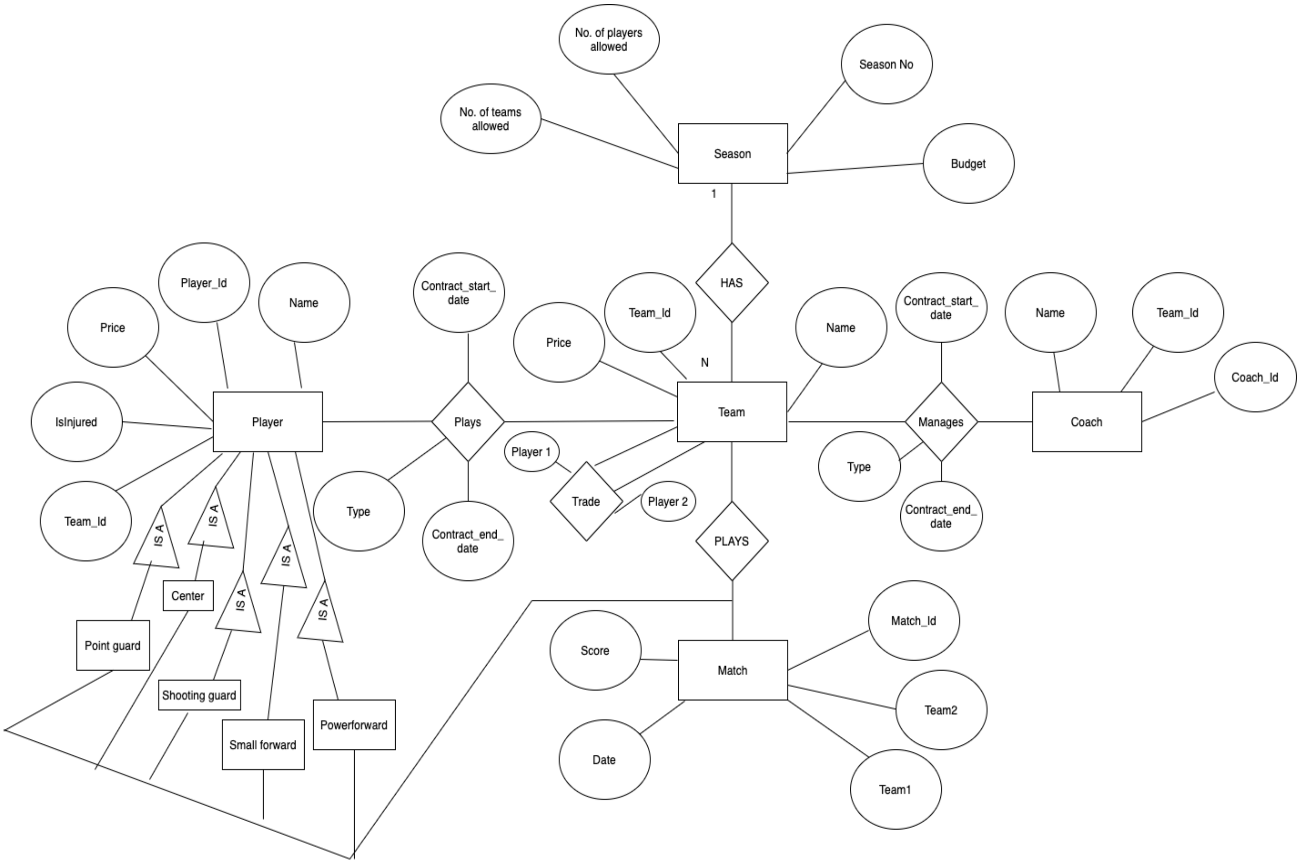
Assignment 2

Q1.



Q2.

```
CREATE DATABASE basketball;

USE basketball;

CREATE TABLE IF NOT EXISTS league(

   season_no int NOT NULL AUTO_INCREMENT,

   no_of_teams int NOT NULL CHECK(no_of_teams > 0),

   no_of_players int NOT NULL CHECK(no_of_players > 6),

   budget double int NULL CHECK(budget > 0),
   PRIMARY KEY (season_no)
);
CREATE TABLE IF NOT EXISTS team(
   team_id int NOT NULL,
   team_name varchar(255) NOT NULL ,
   team_price int NOT NULL,
   team_coach int NOT NULL,
   PRIMARY KEY (id),
   FOREIGN KEY (team_price) REFERENCES player(player_price),
   FOREIGN KEY (team_coach) REFERENCES coach(coach_id)
);


CREATE TABLE IF NOT EXISTS player(
   player_id int NOT NULL ,
   player_name varchar(255) NOT NULL,
   player_price double NOT NULL,
   team_id INT,
   is_injured boolean NOT NULL DEFAULT 0,
   contract_start_date DATE NOT NULL,
   contract_end_date DATE NOT NULL,
   contract_type varchar(12) NOT NULL,
   position int position_id NOT NULL,
   PRIMARY KEY (player_id),
   FOREIGN KEY (team_id) REFERENCES team(team_id)
);
```

```
CREATE TABLE IF NOT EXISTS position(
  id int NOT NULL,
  name varchar(255) NOT NULL,
  PRIMARY KEY (id),
 );


CREATE TABLE IF NOT EXISTS contract(
  contract_id int NOT NULL,
  team_id int NOT NULL,
  player_id int NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (team_id) REFERENCES team(team_id),
  FOREIGN KEY (player_id) REFERENCES player(player_id)
 );


CREATE TABLE IF NOT EXISTS coach(
  coach_id int NOT NULL,
  coach_name varchar(255) NOT NULL ,
  team_id int NOT NULL,
  contract_start_date DATE NOT NULL,
  contract_end_date DATE NOT NULL,
  contract_type varchar(12) NOT NULL,
  PRIMARY KEY (coach_id),
  FOREIGN KEY (team_id) REFERENCES team(team_id),
 );
```

```
CREATE TABLE IF NOT EXISTS matches(
   match_id int NOT NULL ,
   team1_name varchar(255) NOT NULL,
   team2_name varchar(255) NOT NULL,
   score_team1 int NOT NULL,
   score_team2 int NOT NULL.
   PRIMARY KEY (match_id),
   FOREIGN KEY (team1_name) REFERENCES team(team_name),
   FOREIGN KEY (team2_name) REFERENCES team(team_name)
);
```

Q3.

```
INSERT INTO league(season_no, no_of_teams, no_of_players, budget) SELECT
FLOOR(RAND()*(25-10+1))+10,
FLOOR(RAND()*(10-5+1))+5,
FLOOR(RAND()*(15-6+1))+6,
1000000 + (CONVERT(INT, CRYPT_GEN_RANDOM(3)) % 1000000);
```

Q4.

```
a. CREATE FUNCTION add_injury_list(plyr_id int) RETURNS void AS $$
    DECLARE
    BEGIN
UPDATE player SET is_injured = true WHERE player_id = plyr_id
END;
    $$ LANGUAGE plpgsql;
CREATE FUNCTION remove_injury_list(plyr_id int) RETURNS void AS $$
    DECLARE
    BEGIN
UPDATE player SET is_injured = false WHERE player_id = plyr_id
END;
 $$ LANGUAGE plpgsql;
```

```
b. CREATE FUNCTION player_trade(team1_id int, plyr1_id int, team2_id int, plyr2_id int)
RETURNS void AS $$
    DECLARE
    BEGIN
UPDATE player SET team_id = team2_id WHERE player_id = plyr1_id;
UPDATE player SET team_id = team1_id WHERE player_id = plyr2_id;
             UPDATE contract SET player_id = player2_id WHERE team_id = team1_id;
UPDATE contract SET player_id = player1_id WHERE team_id = team2_id;
END;
```

```
    $$ LANGUAGE plpgsql;
```

```
c. Select

   (SELECT player_id FROM player WHERE team_id=1 AND position=1 ORDER by price DESC
LIMIT 1) AS “PointGuard”

    (SELECT player_id FROM player WHERE team_id=1 AND position=2 ORDER by price DESC
LIMIT 1) AS “ShootingGuard”

    (SELECT player_id FROM player WHERE team_id=1 AND position=3 ORDER by price DESC
LIMIT 1) AS “SmallForward”

     (SELECT player_id FROM player WHERE team_id=1 AND position=4 ORDER by price DESC
LIMIT 1) AS “PowerForward”

     (SELECT player_id FROM player WHERE team_id=1 AND position=5 ORDER by price DESC
LIMIT 1) AS “Center”
```

```
d. SELECT * FROM team ORDER BY team_price DESC

SELECT * FROM player ORDER BY player_price DESC
```