CPSC 304 Project Cover Page

Milestone #: ____2____

Date: 2022-10-21

Group Number: 08

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Jia Lu	28829489	b9n2c	lujiaca16@gmail.com
Yingquan Wang	41274853	d7u2x	yingquan@student.ubc.ca
Qiyu Zhou	87141024	v0h2x	zqy2002@student.ubc.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.

University of British Columbia, Vancouver

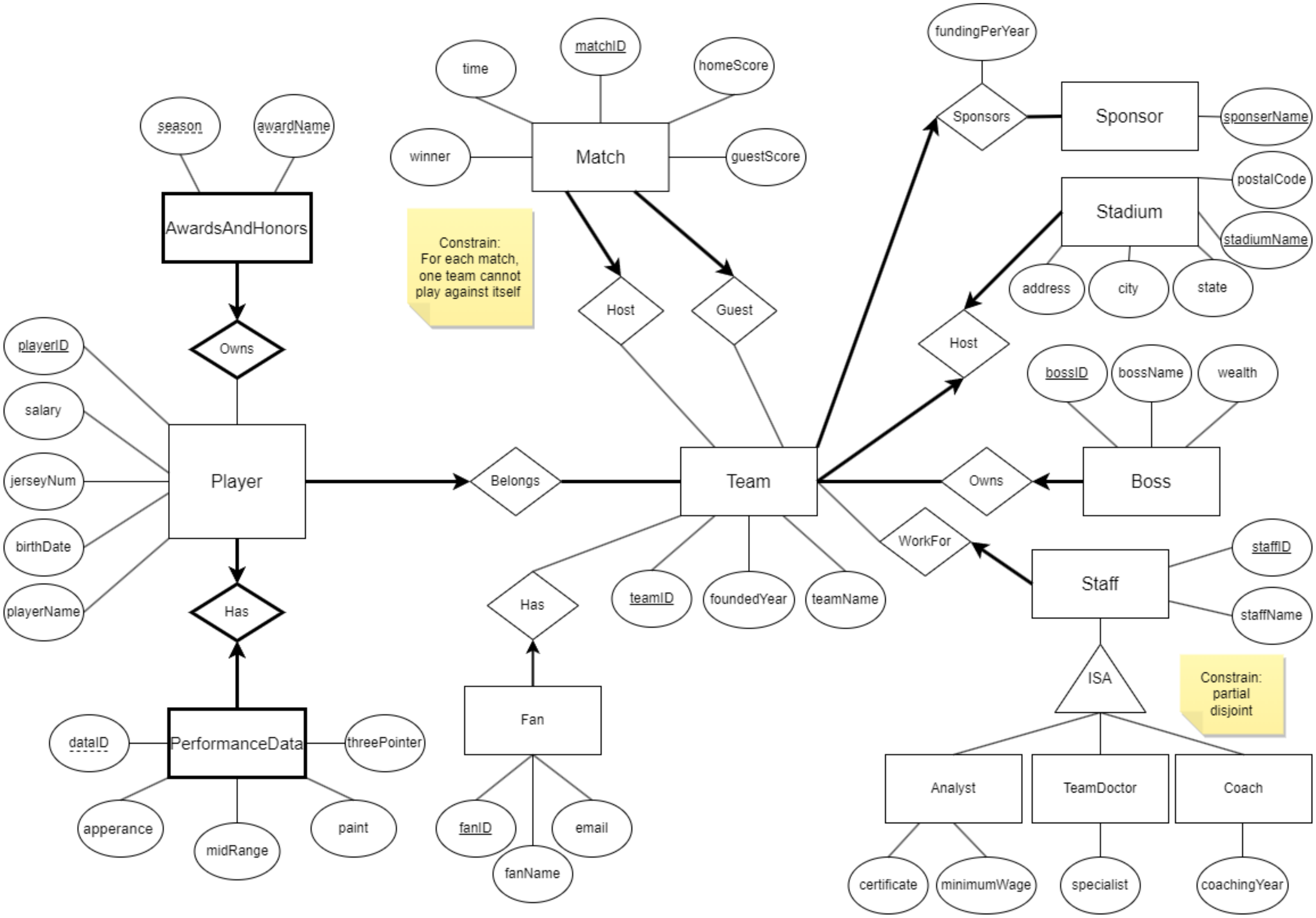
Department of Computer Science

Contents

ER Diagram

Change in ER Diagram:

- Added “postalCode” and “state” to “Stadium”.
- Added “minimumWage” to “Analyst”.
- Changed “birth data” to “birthDate” in “Player”.
- Changed primary key of “Stadium” from “name” and “city” to “name” only.
- Changed “3-pointer” to “threePointer”.
- Changed “mid-range” to “midRange”.
- Deleted “home” and “guest” in “Match”.
- Deleted “weight” in “Player”.
- Fixed wrong spelling of “jerseyNum” in “Player”.
- Fixed capitalization of “Coach”.
- Fixed “PerformanceData” as weak entity but not surrounded with think lines.
- Forced each “Player” must play for a “Team”.
- Forced each “Team” must have a “Sponsor”.
- Made ids and names more specific.



Schema – Relational Model

Primary Key: underlined
Foreign Key: **bolded**
Candidate Key: listed in Table 1.1

Player(playerID: integer, **teamID**: integer, playerName: string, birthdate: date, jerseyNum: integer, salary: integer, appearance: integer, threePointer: integer, midRange: integer, paint: integer)

AwardsAndHonors(**playerID**: integer, awardName: string, season: string)

Fan(fanID: integer, **teamID**: integer, fanName: string, email: string)

Boss (bossID: integer, **teamID**: integer, bossName: string, wealth: integer)

Staff (staffID: integer, **teamID**: integer, staffName: string)
[Constrain: partial, disjoint.]

Coach (**staffID**: integer, coachingYear: integer)

Analyst (**staffID**: integer, certificate: string)

TeamDoctor (**staffID**: integer, specialist: string)

Team_Sponsors_Stadium(teamID: integer, **stadiumName**: string, foundedYear: integer, teamName: string, sponserName: string, fundingPerYear: integer)

Stadium(stadiumName: string, **teamID**: integer, postalCode: string, state: string, city: string , address: string)

Match_Home_Guest(matchID: integer, **homeTeamID**: integer, **guestTeamID**: integer, winner: string, time: timestamp, homeScore: integer, guestScore: integer)
[Constrain: For each match, one team cannot play against itself.]

Table 1.1 Candidate Keys for Relational Models

Relational Model	Candidate keys
Player	Key 1: playerID
AwardsAndHonors	Key 1: **playerID**, season, awardName
Fan	Key 1: fanID
Team_Sponsors_Stadium	Key 1: teamID
Match_Home_Guest	Key 1: matchID Key 2: time, homeTeamID, guestTeamID Key 3: time, winner
Stadium	Key 1: stadiumName
Boss	Key 1: bossID
Staff	Key 1: staffID
Coach	Key 1: **staffID**
Analyst	Key 1: **staffID**
TeamDoctor	Key 1: **staffID**

Functional Dependencies

Player

- playerID → teamID, playerName, birthdate, jerseyNum, salary, appearance, threePointer, paint, midRange

AwardsAndHonors

- none

Fan

- fanID → teamID, fanName, email

Boss

- BossID → teamID, bossName, wealth

Staff

- staffID → teamID, staffName

Coach

- staffID → coachingYear

Analyst (2NF)

- staffID → certificate, minimumWage
- certificate → minimumWage

TeamDoctor

- staffID → specialist

Team_Sponsors_Stadium

- teamID → stadiumName, foundedYear, name, sponserName, fundingPerYear

Stadium (2NF)

- stadiumName → address, city, province, postalCode
- postalCode → state, city
- address, city, province → postalCode

Match_Home_Guest

- matchID → homeTeamID, guestTeamID, winner, time, homeScore, guestScore
- time, homeTeamID, guestTeamID → matchID, winner, homeScore, guestScore
- winner, time → homeTeamID, guestTeamID, matchID, homeScore, guestScore

Normalization

Stadium (2NF)

R(stadiumName, address, city, state, postalCode)

Finding minimal covers of FDs

Step 1: Put FDs in standard form

- stadiumName → address
- stadiumName → city
- stadiumName → state
- stadiumName → postalCode
- postalCode → state
- postalCode → city
- address, city, state → postalCode

Step2: Minimize LHS of each FD

- Already minimized

Step3: Delete Redundant FDs

- stadiumName → address
- stadiumName → city (deleted)
- stadiumName → state (deleted)
- stadiumName → postalCode
- postalCode → state
- postalCode → city
- address, city, state → postalCode

Decomposition into 3NF Method 1: Lossless-joint method

To make it easy for decomposition, denote:

stadiumName	N
postalCode	P
state	S
city	C
address	A

1. The minimal cover has been found as:

   N → A (BCNF)
   N → P (BCNF)
   P → S (Violation)
   P → C (Violation)
   ACS → P (Violation)

2. Decompose the relation which violates 3NF/BCNF
   It is found that P→S, P→C, and ACS→P violate 3NF, and need to be decomposed. Starting from R1(NCAP), R2(PS). In R1, since P→C is neither 3NF nor BCNF , continue decomposing R1 into R3(NAP), R4(PC). Now we have three relations: R2(PS), R3(NAP), R4(PC).,
3. Identify the set of dependencies that are not preserved by the decomposition:
   Note that we still miss one FD, which is ACS → P.
4. We create a relation R5(ACSP) and add it to the composition. Now we have R2(PS), R3(NAP), R4(PC), R5(ACSP).
5. Note that R2(PS) and R4(PC) are covered in R5(ACSP), so we remove R2 and R4. The final results of lossless-joint decomposition: R5(ACSP), R3(NAP).

Analyst (2NF)

R(staffID, certificate, minimumWage)

Finding minimal covers of FDs

Step 1: Put FDs in standard form

- staffID → certificate
- staffID → minimumWage
- certificate → minimumWage

Step2: Minimize LHS of each FD

- Already Minimized

Step3: Delete Redundant FDs

- staffID → minimumWage (delete)

Decomposition into 3NF Method #2: Using a Minimal Cover and 3NF Synthesis

The final result: R1(staffID, certificate), R2(certificate, minimumWage)

SQL Statements

We find that Oracle does not support ON UPDATE CASCADE, so we decide our table does not allow changes on PKs, and it is reasonable in this case.

Player(playerID: integer, **teamID**: integer, playerName: string, birthdate: date, jerseyNum: integer, salary: integer, appearance: integer, threePointer: integer, midRange: integer, paint: integer)

```
CREATE TABLE Player (
        playerID INT PRIMARY KEY,
        teamID INT,
        playerName CHAR(100),
        birthdate DATE,
        jerseyNum INT NOT NULL,
        salary INT,
        appearance INT,
        threePointer INT,
        midRange INT,
        paint INT,
        UNIQUE (teamID, jerseyNum),
        FOREIGN KEY (teamID) REFERENCES Team_Sponsors_Stadium (teamID) ON DELETE
        CASCADE
);
```

AwardsAndHonors(**playerID**: integer, awardName: string, season: string)

```
CREATE TABLE AwardsAndHonors (
        playerID INT,
        awardName CHAR(100),
        season CHAR(100),
        PRIMARY KEY (playerID, awardName, season),
        FOREIGN KEY (playerID) REFERENCES Player(playerID) ON DELETE CASCADE
);
```

Fan(fanID: integer, **teamID**: integer, fanName: string, email: string)

```
 CREATE TABLE Fan (
        fanID INT PRIMARY KEY,
        teamID INT,
        fanName CHAR(100),
        email CHAR(100) UNIQUE,
        FOREIGN KEY (teamID) REFERENCES Team_Sponsors_Stadium (teamID) ON DELETE
        CASCADE
);
```

Boss (bossID: integer, **teamID**: integer, bossName: string, wealth: integer)

```
CREATE TABLE Boss (
        bossID INT PRIMARY KEY,
        teamID INT,
        bossName CHAR(100),
        wealth INT,
        FOREIGN KEY (teamID) REFERENCES Team_Sponsors_Stadium (teamID) ON DELETE
        CASCADE
);
```

Staff (staffID: integer, **teamID**: integer, staffName: string)

```
CREATE TABLE Staff (
        staffID INT PRIMARY KEY,
        teamID INT,
        staffName CHAR(100),
        FOREIGN KEY (teamID) REFERENCES Team(teamID) ON DELETE CASCADE
);
```

Coach (**staffID**: integer, coachingYear: integer)

```
CREATE TABLE Staff (
        staffID INT PRIMARY KEY,
        coachingYear INT,
        FOREIGN KEY (staffID) REFERENCES Staff(staffID) ON DELETE CASCADE
);
```

Analyst (**staffID**: integer, **certificate**: string)

```
CREATE TABLE Analyst (
        staffID INT PRIMARY KEY,
        certificate CHAR(100),
        FOREIGN KEY (staffID) REFERENCES Staff(staffID) ON DELETE CASCADE,
        FOREIGN KEY (certificate) REFERENCES Certificate_Wage(certificate)
);
```

Certificate_Wage (certificate: string, minimumWage: integer)

```
CREATE TABLE Certificate_Wage (
        certificate CHAR(100) PRIMARY KEY,
        minimumWage INT
);
```

TeamDoctor(**staffID**: integer, specialist: string)

```
CREATE TABLE  TeamDoctor (
        staffID  INT PRIMARY KEY,
        specialist CHAR(100),
        FOREIGN KEY (staffID) REFERENCES Staff(staffID) ON DELETE CASCADE
);
```

Stadium_Location(stadiumName: string, **postalCode**: string, address: string)

```
CREATE TABLE Stadium_Location(
        stadiumName CHAR(100) PRIMARY KEY,
        postalCode CHAR(100),
        address CHAR(100),
        FOREIGN KEY (postalCode) REFERENCES PostalCode_Location(postalCode)
);
```

PostalCode_Location(postalCode: string, state: string, city: string address: string)

```
CREATE TABLE PostalCode_Location (
        postalCode CHAR(100) PRIMARY KEY,
        state CHAR(100),
        city CHAR(100),
        address CHAR(100)
);
```

Team_Sponsors_Stadium(teamID: integer, **stadiumName**: string, foundedYear: integer, teamName: string, sponserName: string, fundingPerYear: integer)

```
CREATE TABLE Team_Sponsors_Stadium (
        teamID INT PRIMARY KEY,
        stadiumName CHAR(100),
        foundedYear INT,
        teamName CHAR(100),
        sponserName CHAR(100),
        fundingPerYear INT,
        FOREIGN KEY (stadiumName) REFERENCES Stadium_Location(stadiumName)
);
```

Match_Home_Guest(matchID: integer, **homeTeamID**: integer, **guestTeamID**: integer, winner: string, time: timestamp, homeScore: integer, guestScore: integer)

```
CREATE TABLE Match_Home_Guest (
        matchID INT PRIMARY KEY,
        homeTeamID INT,
        guestTeamID INT,
        winner CHAR(100),
        time TIMESTAMP,
        homeScore INT,
        guestScore INT,
        FOREIGN KEY (homeTeamID) REFERENCES Team_Sponsors_Stadium (teamID),
        FOREIGN KEY (guestTeamID) REFERENCES Team_Sponsors_Stadium (teamID),
        UNIQUE (time, homeTeamID, guestTeamID),
        UNIQUE (time, winner)
);
```

[Constrain: For each match, one team cannot play against itself]

INSERT Statements

Player(playerID: integer, **teamID**: integer, playerName: string, birthdate: date, jerseyNum: integer, salary: integer, appearance: integer, threePointer: integer, midRange: integer, paint: integer);

```
INSERT INTO Player VALUES(1, 1, “LeBron James”, 1984-12-30, 6, 100, 1354, 312, 5984, 416);
INSERT INTO Player VALUES(2, 2, “Kawhi Leonard”, 1991-6-29, 6, 90, 984, 562, 648, 354);
INSERT INTO Player VALUES(3, 3, ‘Giannis Antetokounmpo’, ‘1994-Dec-6’, 34, 2300000,126,212,2134,352);
INSERT INTO Player VALUES(4, 4, ‘Stephen Curry, 1988-3-14, 30, 230,652,121,331,101);
INSERT INTO Player VALUES(5, 5, “Kevin Durant”, 1988-9-29, 5, 90, 212, 120, 92, 79);
```

AwardsAndHonors(**playerID**: integer, awardName: string, season: string)

```
INSERT INTO AwardsAndHonors VALUES(1, ‘ROY’, ‘2001-2002’);
INSERT INTO AwardsAndHonors VALUES(2, ‘MVP’, ‘2007-2008’);
INSERT INTO AwardsAndHonors VALUES(3, ‘DPOY’, ‘2009-2010’);
INSERT INTO AwardsAndHonors VALUES(4, ‘MIP’, ‘2011-2012’);
INSERT INTO AwardsAndHonors VALUES(5, ‘Scoring Champ’, ‘2001-2002’);
```

Fan(fanID: integer, **teamID**: integer, fanName: string, email: string)

```
INSERT INTO Fan VALUES(1, 1, ‘Qiyu Zhou’, ‘qiyuzhou@gmail.com’);
INSERT INTO Fan VALUES(2, 4, ‘Jia Lu’, ‘jialu@gmail.com’);
INSERT INTO Fan VALUES(3, 6, ‘Yingquan Wang’, ‘yingquanwang@gmail.com’);
INSERT INTO Fan VALUES(4, 8, ‘Hibah Rasmussen’, ‘hibahrasmussen @gmail.com’);
INSERT INTO Fan VALUES(5, 10, ‘Linda Dunn’, ‘lindadunn@gmail.com’);
```

Boss (bossID: integer, **teamID**: integer, bossName: string, wealth: integer)

```
INSERT INTO Boss VALUES(1, 1, ‘Dougie Calvert’, 67000000);
INSERT INTO Boss VALUES(2, 2, ‘Jarrad Seymour’, 92300000);
INSERT INTO Boss VALUES(3, 3, ‘Riccardo Farley’, 20000000);
INSERT INTO Boss VALUES(4, 4, ‘Remi Jacobson’, 52000000);
INSERT INTO Boss VALUES(5,5, ‘Flora Fenton’, 21000000);
```

Staff (staffID: integer, **teamID**: integer, staffName: string)

```
INSERT INTO Staff VALUES(1, 1, ‘Torin Hodson’);
INSERT INTO Staff VALUES(2, 1, ‘Geoffrey Lloyd);
INSERT INTO Staff VALUES(3, 2, ‘Beau Conner’);
INSERT INTO Staff VALUES(4, 2, ‘Aditya Kim);
INSERT INTO Staff VALUES(5, 3, ‘Rohan Whelan’);
INSERT INTO Staff VALUES(6, 1, ‘Vincent Quintero’);
INSERT INTO Staff VALUES(7, 1, ‘Kevin Andrade’);
INSERT INTO Staff VALUES(8, 2, ‘Vlad Boyd’);
INSERT INTO Staff VALUES(9, 2, ‘Jordanne Green’);
INSERT INTO Staff VALUES(10, 3, ‘Taran Roth’);
INSERT INTO Staff VALUES(11, 1, ‘Haleemah Callaghan’);
INSERT INTO Staff VALUES(12, 1, ‘Daisy Brown’);
INSERT INTO Staff VALUES(13, 2, ‘Nelly Bate’);
INSERT INTO Staff VALUES(14, 2, ‘Alexandre Gross’);
INSERT INTO Staff VALUES(15, 3, ‘Daniyal Rose’);
```

Coach (**staffID**: integer, coachingYear: integer)

```
INSERT INTO Coach VALUES(1, 8);
INSERT INTO Coach VALUES(2, 2);
INSERT INTO Coach VALUES(3, 1);
INSERT INTO Coach VALUES(4, 15);
INSERT INTO Coach VALUES(5, 3);
```

Analyst (**staffID**: integer, **certificate**: string)

```
INSERT INTO Analyst VALUES(6, ‘Expert’);
INSERT INTO Analyst VALUES(7, ‘Proficient’);
INSERT INTO Analyst VALUES(8, ‘Advanced’);
INSERT INTO Analyst VALUES(9, ‘Intermediate’);
INSERT INTO Analyst VALUES(10, ‘Entry’);
```

Certificate_Wage (certificate: string, minimumWage: integer)

```
INSERT INTO Certificate_Wage VALUES(‘Entry’, 100000);
INSERT INTO Certificate_Wage VALUES(‘Intermediate’, 200000);
INSERT INTO Certificate_Wage VALUES(‘Advanced’, 400000);
INSERT INTO Certificate_Wage VALUES(‘Proficient’, 800000);
INSERT INTO Certificate_Wage VALUES(‘Expert’, 1600000);
```

TeamDoctor (**staffID**: integer, specialist: string)

```
INSERT INTO TeamDoctor VALUES(11, ‘Cardiology’);
INSERT INTO TeamDoctor VALUES(12, ‘Orthopedic surgery’);
INSERT INTO TeamDoctor VALUES(13, ‘Orthopedic surgery’’);
INSERT INTO TeamDoctor VALUES(14, ‘Psychiatry’);
INSERT INTO TeamDoctor VALUES(15, ‘Neurology’);
```

Stadium_Location(stadiumName: string, **postalCode**: string, address: string)

```
INSERT INTO Stadium_Location VALUES(‘Chase Center’, ‘94158’, ‘1 Warriors Way’);
INSERT INTO Stadium_Location VALUES(‘Crypto.com Arena’, ‘90015’, ‘1111 S Figueroa St’);
INSERT INTO Stadium_Location VALUES(‘TD Garden’, ‘02114’, ‘100 Legends Way’);
INSERT INTO Stadium_Location VALUES(‘AT&T Center’, ‘78219’, ‘1 AT&T Center Parkway’);
```

```
    INSERT INTO Stadium_Location VALUES(‘FTX Arena’, ‘33132’, ‘601 Biscayne Blvd’);
    INSERT INTO Stadium_Location VALUES(‘Fiserv Forum’, ‘53203’, ‘1111 Vel R. Phillips Ave’);
```

PostalCode_Location(postalCode: string, state: string, city: string address: string)

```
    INSERT INTO PostalCode_Location VALUES(‘94158’, ‘CA’, ‘San Francisco’, ‘1 Warriors Way’);
    INSERT INTO PostalCode_Location VALUES(‘90015’, ‘CA’, ‘Los Angeles’, ‘1111 D Figueroa St’);
    INSERT INTO PostalCode_Location VALUES(‘02114’, ‘MA’, ‘Boston’, ‘100 Legends Way’);
    INSERT INTO PostalCode_Location VALUES(‘78219’, ‘TX’, ‘San Antonio’, ‘1 AT&T Center Parkway’);
    INSERT INTO PostalCode_Location VALUES(‘33132’, ‘FL’, ‘Miami’, ‘601 Biscayne Blvd’);
    INSERT INTO PostalCode_Location VALUES(‘53203’, ‘WI’, ‘Milwaukee’, ‘1111 Vel R. Phillips Ave’);
```

Team_Sponsors_Stadium(teamID: integer, **stadiumName**: string, foundedYear: integer, teamName: string, sponserName: string, fundingPerYear: integer)

```
    INSERT INTO Team_Sponsors_Stadium VALUES(4, ‘Chase Center’, 1946, ‘Warrior’, ‘Rakuten’, 1000);
    INSERT INTO Team_Sponsors_Stadium VALUES(1, ‘Crypto.com Arena’, 1947, ‘Lakers’, ‘Bibigo’, 1000);
    INSERT INTO Team_Sponsors_Stadium VALUES(2, ‘AT&T Center’, 1967, ‘Spurs’, ‘Self Financial’, 900);
    INSERT INTO Team_Sponsors_Stadium VALUES(3, ‘Fiserv Forum’, 1968, ‘Bucks’, ‘Motorola’, 900);
    INSERT INTO Team_Sponsors_Stadium VALUES(5, ‘Barclays Center’, 1967, ‘Nets’, ‘Webull’, 800);
```

Match_Home_Guest(matchID: integer, **homeTeamID**: integer, **guestTeamID**: integer, winner: string, time: timestamp, homeScore: integer, guestScore: integer)

```
    INSERT INTO Match_Home_Guest VALUES(1, 1, 2, ‘Lakers’, ‘2022-10-15 19:00:00.00’’, 101, 98);
    INSERT INTO Match_Home_Guest VALUES(2, 2, 3, ‘Spurs’, ‘2022-10-16 19:00:00.00’, 97, 88);
    INSERT INTO Match_Home_Guest VALUES(3, 3, 4, ‘Bucks’, ‘2022-10-17 19:00:00.00’, 83, 78);
    INSERT INTO Match_Home_Guest VALUES(4, 4, 5, ‘Warrior’, ‘2022-10-18 19:00:00.00’, 121, 100);
    INSERT INTO Match_Home_Guest VALUES(5, 5, 1, ‘Nets’, ‘2022-10-19 19:00:00.00’, 96, 85);
```