**CPSC 304 Project Cover Page**

Milestone #: \_\_\_4\_\_\_\_\_

Date: \_\_November 30th, 2023\_\_\_

Group Number: \_\_\_155\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Student Number** | **CS Alias (Userid)** | **Preferred E-mail Address** |
| Armaan Sawhney | 44616670 | b6x2w | armaansawhney070903@gmail.com |
| Zaid Khan | 23739394 | w9f6u | zkhan1605@gmail.com |
|  |  |  |  |

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

**REPOSITORY LINK**

<https://github.students.cs.ubc.ca/CPSC304-2023W-T1/project_b6x2w_p9m7b_w9f6u>

**PROJECT DESCRIPTION**

- The domain of an application is in sports. The application keeps track of a basketball

league with player information, team information, game information, sponsors, owners,

game locations, etc.

- In real-life situations, the database can be used to search up records of certain game(s), player stats, and team information, etc.

- The database for the basketball application will provide users with a range of

functionalities and capabilities, allowing them to create, modify and delete team members.

- We decided to build up our project using the department provided Oracle servers

as our relational database management system. We used Java (JFrame) along

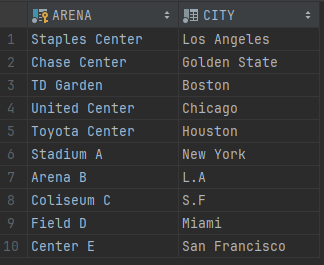
with Oracle as our application technology stack.

**Description of Final Schema**

* Our schema is relatively similar to the original one that we submitted, with basic changes such as:
  + added “ON DELETE CASCADE” constraints for TeamMember references.
  + Added foreign key constraints to Team
  + Added new field and foreign key constraint in Coach
  + Added foreign key constraints to Owner

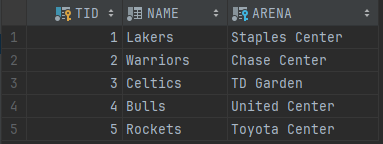
**Current Relation Schema**

Location (arena, city)

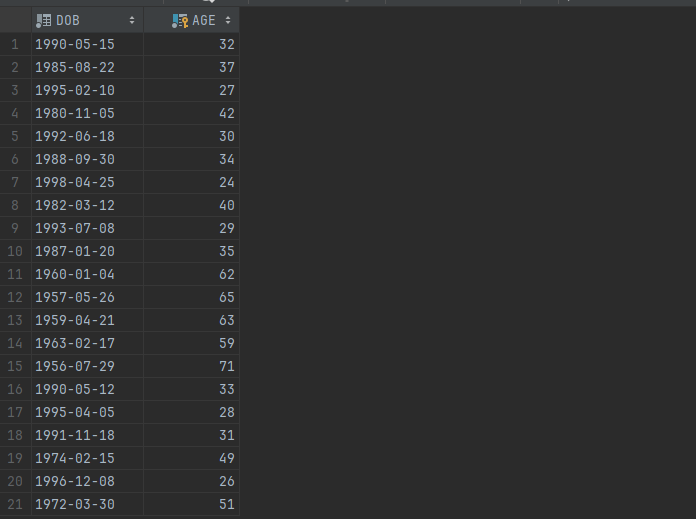
* Primary Key: arena

Team (tid, name, **arena**)

* Primary Key: tid
* Foreign Key: arena (references Location(arena))

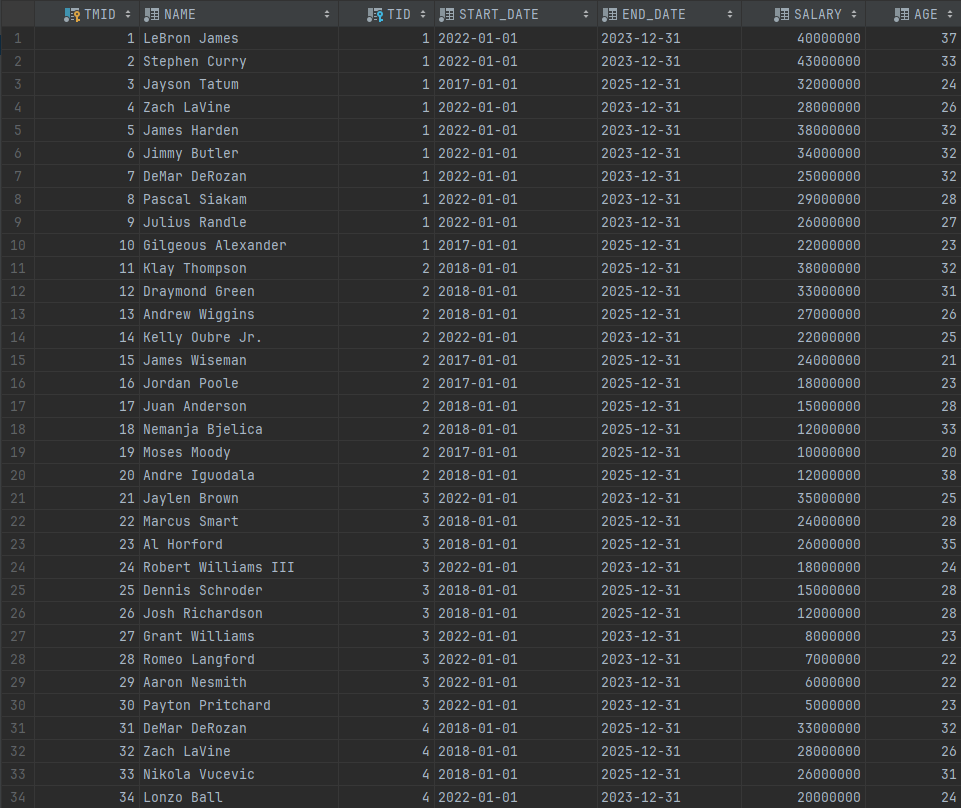


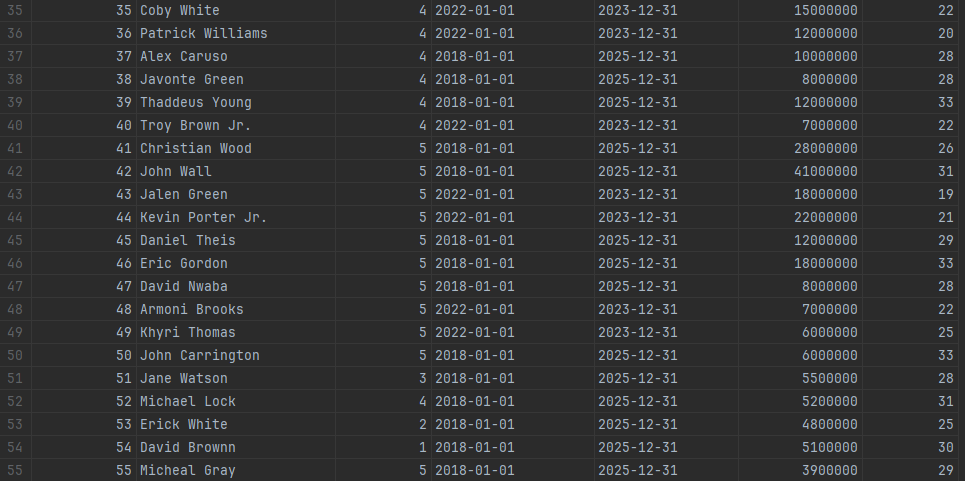
Age (dob, age)

* Primary Key: age

TeamMember (tmid, name, **tid**, start\_date, end\_date, salary, **age**)

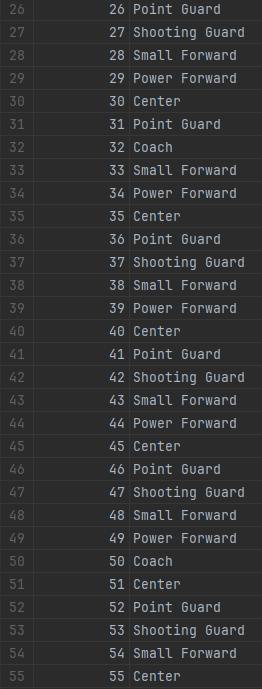
* Primary Key: tmid
* Foreign Keys:
  + tid (references Team(tid) on delete cascade)
  + age (references Age(age))



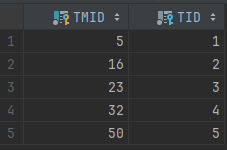


Player (**pid**, position)

* Primary Key: pid
* Foreign Key: pid (references Team Member(tmid))

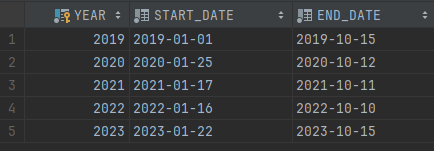


Coach (**tmid**, **tid**)

* Primary Key: tmid
* Foreign Keys:
  + tmid (references TeamMember(tmid) on delete cascade)
  + tid (references Team(tid))

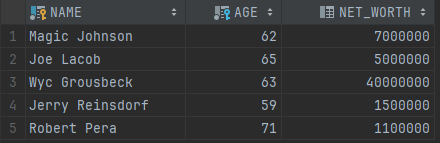
Season (year, start\_date, end\_date)

* Primary Key: year



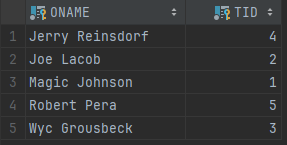
Owner (name, **age**, net\_worth)

* Primary Key: name
* Foreign Key: age (references Age(age))

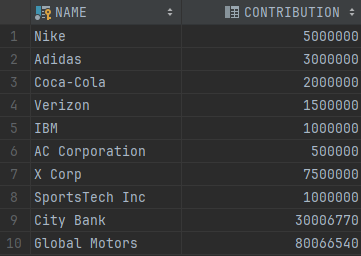


Owns (**oname, tid**)

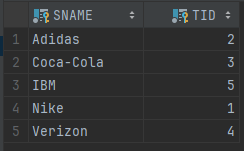
* Primary Key: (oname, tid)
* Foreign Keys:
  + oname (references Owner(name) on delete cascade)
  + tid (references Team(tid))



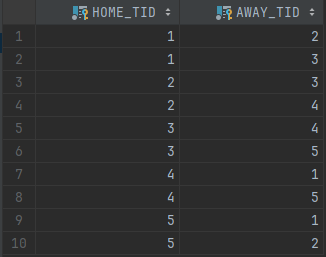
Sponsor (name, contribution)

* Primary Key: name

Sponsors (**sname, tid**)

* Primary Key: (sname, tid)
* Foreign Keys:
  + sname (references Sponsor(name))
  + tid (references Team(tid))

HasPlayed **(home\_tid, away\_tid**)

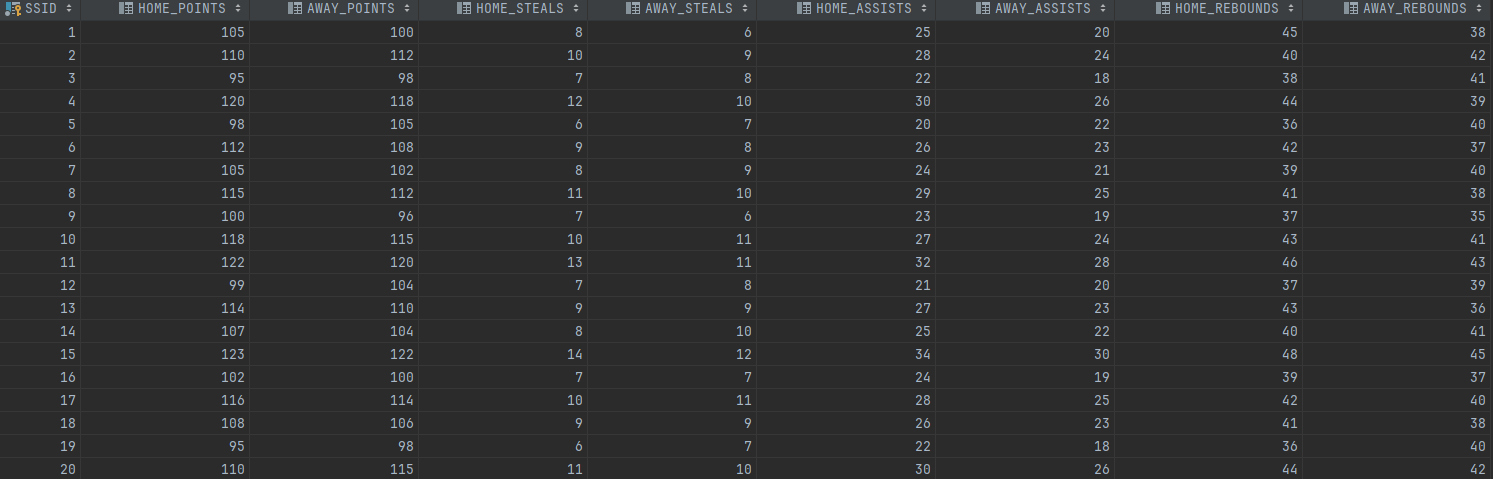
* Primary Key: (home\_tid, away\_tid)
* Foreign Keys:
  + home\_tid (references Team(tid))
  + away\_tid (references Team(tid))

Referee (rid, name, experience\_years)

* Primary Key: rid

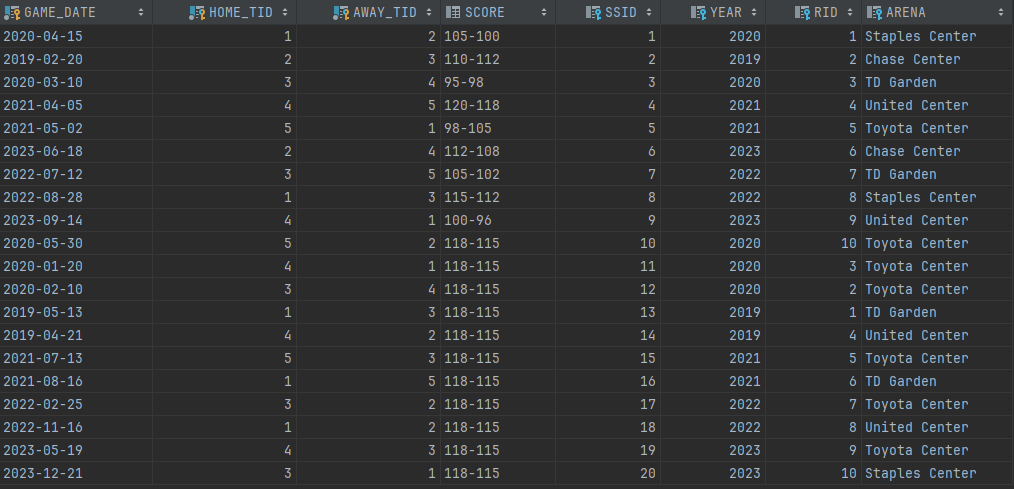
StatSheet (ssid, home\_points, away\_points, home\_steals, away\_steals, home\_assists, away\_assists, home\_rebounds, away\_rebounds)

* Primary Key: ssid



Game (**game\_date, home\_tid, away\_tid**, score, **ssid, year, rid, arena**)

* Primary Key: (game\_date, home\_tid, away\_tid)
* Foreign Keys:
  + home\_tid (references Team(tid))
  + away\_tid (references Team(tid))
  + ssid (references StatSheet(ssid))
  + year (references Season(year))
  + rid (references Referee(rid))
  + arena (references Location(arena))



**REQUIRED SQL QUERIES**

All queries listed before are in class DatabaseConnectionHandler in package database. Screenshots for before / during / after executing the queries for each case is listed below.

1. **INSERT QUERY**

443. INSERT INTO TeamMember VALUES (?,?,?,?,?,?,?);

465. INSERT INTO Player VALUES (?,?);

1. **DELETE QUERY**

210. DELETE FROM TeamMember WHERE tmid = ?;

1. **UPDATE QUERY**

230. UPDATE TeamMember

SET tid = ?, age = ?, salary = ?, start\_date = ?, end\_date = ?

WHERE tmid = ?;

1. **SELECTION QUERY**

86. SELECT t1.name as name1, t2.name as name2

FROM game g, team t1, team t2

WHERE g.home\_tid = t1.tid AND g.AWAY\_TID = t2.TID

AND (t1.name = ? OR t2.name = ?);

1. **PROJECTION**

370. SELECT ssid, home\_points, home\_steals, home\_assists, home\_rebounds

FROM Statsheet WHERE ssid = ?;

396. SELECT ssid, away\_points, away\_steals, away\_assists, away\_rebounds

FROM Statsheet WHERE ssid = ?;

1. **JOIN QUERY**

112. SELECT s.name as sponsor, s.CONTRIBUTION FROM Sponsors ss

JOIN Sponsor s ON s.name = ss.sname

JOIN Team t ON t.TID = ss.tid

WHERE t.name = ?;

1. **AGGREGATION WITH GROUP BY QUERY**

485. SELECT SUM(salary) AS total

FROM TeamMember WHERE tid = ?;