



# Basketball League DBMS

CPS 510 - Database Systems I

Professor Abdolreza Abhari

TA: Haytham Qushtom

2021-12-03

Waqas Jalali - 500899545

Omaid Nadi - 500777929

Section 1/7

# Table of Contents

---

Introduction/Description .....	2
Entity Relationship Model .....	4
Schema design .....	5
Views and Simple/Advanced Queries .....	11
Unix shell Implementation .....	16
Functional Dependencies .....	18
Normalization in 3NF .....	19
Graphical User Interface of Database .....	21
Relational Algebra for Queries .....	24
Concluding Remarks .....	26

## Introduction/Description

---

These days, many people resort to sports as a way to pass time, watching and playing. The NBA (national basketball association) provides entertainment to over 15.1 million people across the world. This league consists of 30 teams playing top notch basketball.

For this assignment we will be creating our own basketball league. We will use a database similar to the actual NBA.

**Frontend framework:** Java (JavaFX)

**Database:** SQL/MySQL, Oracle, Visual Studio Code

### **Problem Statement:**

A new basketball league is being formed, and the company needs a system to manage their league.

### **Expected Goals:**

The service will provide a way to manage teams, players, schedules, stadiums, standings, scores and stats. League commissioners will be able to manage every part of the league, while league GMs only have the ability to manage their own team, players, and make trades. Players will be able to register for the league and see their stats as they play games.

### **Priority:**

See league info by collecting the data from the database.

### **Expected Features:**

- General Manager
- Standings
- Scores
- Stats
- Schedules
- Registration

### **System Requirements:**

- Roles for accounts (player, gm, teams)
- Database
  - Users table (players, gms, commissioner)
  - Scores table (results of players game)

- Results table (win/loss of team)
- Teams table (teams and which players belong to)
- Matches table (schedule)

### **Customer/User Perspective:**

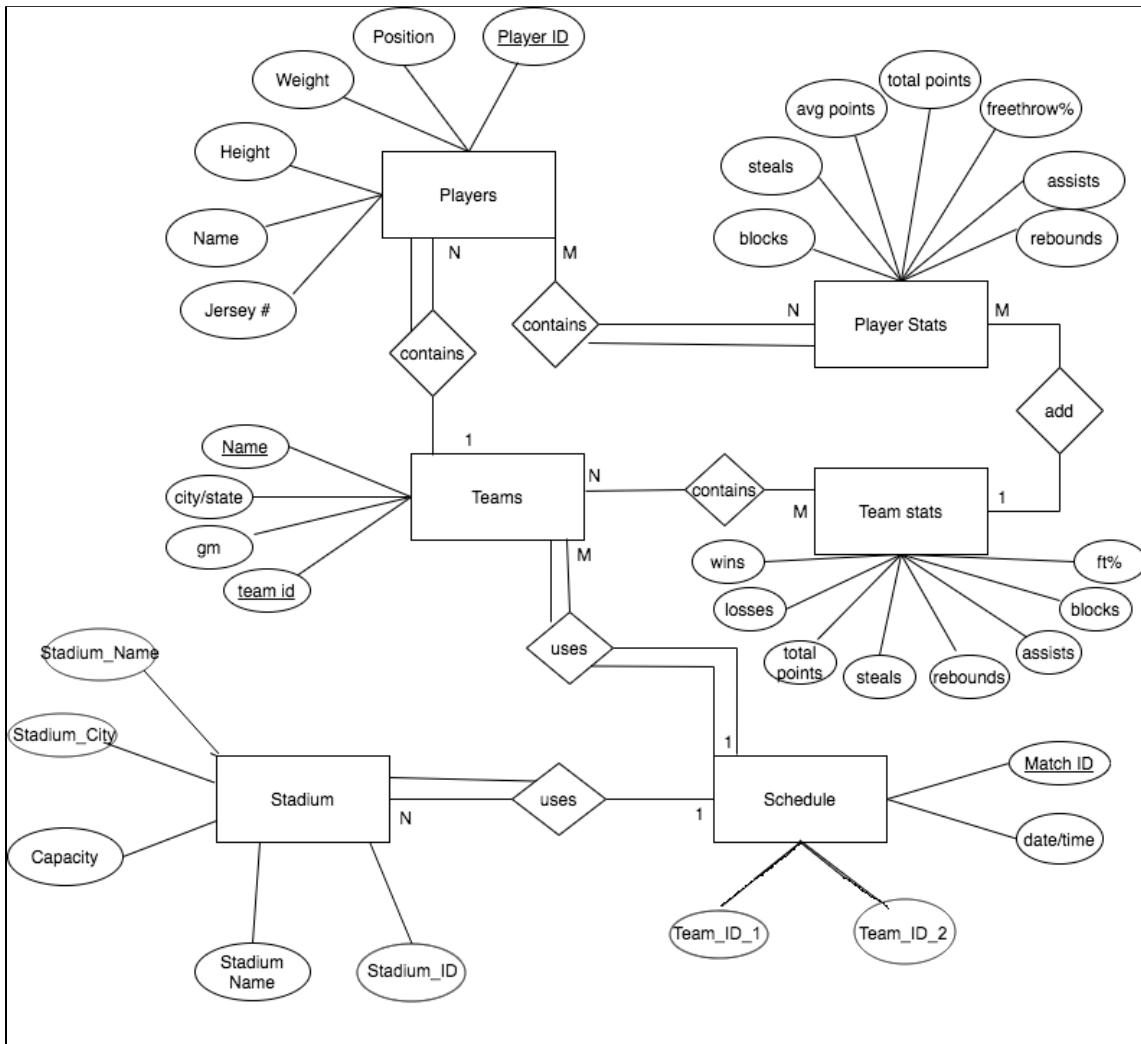
- There will be an authentication system, a control panel for users and an interface for outgoing messages. Depending on the role of the account (player, gm, commissioner) this interface will change
- Users with the commissioner role will be able to edit all of the contents of the league
- Players will be able to see the homepage and edit nothing, only the ability to see the league stats, scores, results, schedule.
- GMs will be able to do the same as players, but have a separate panel for them to make trades, set their team names, and other misc things to manage only their own team.

### **Constraints:**

- Security is important because the system handles the input of stats/scores
- Time management (We may need time to review/learn more about the languages we are using to effectively create the product)
- Availability: As time goes on we may realize that we may need to change some features after launch. During this time the app will be down for maintenance as we the update
- Resource: As students we have other classes that require our attention which reduces the number of hours we are able to put into this project. We as well only have 2 members which cuts the number of hours we can put into this by a third.

# Entity Relationship Model

---



## Schema design and Database construction

---

### Create Tables

```
1  -----
2
3  create table Players
4  (
5    Player_ID int,
6    Name varchar(200),
7    Height int,
8    Weight int,
9    Position varchar(10),
10   Jersy_No int,
11   Team_name varchar(200),
12   Primary key (Player_ID)
13 )
14
15  create  table Teams(
16   Team_ID int,
17   Team_Name varchar(200),
18   City_State varchar(100),
19   gm varchar(100),
20   Primary Key(Team_ID)
21 )
22
23  create  table Player_Status
24  (
25    --Player_Status_ID int,      NOT NEEDED
26    Player_ID int,
27    Blocks int,
28    Steals int,
29    Avg_Points int,
30    Total_Points int,
31    Games_Played int,
32    assists int,
33    rebounds int,
34    FreeThrowPercent int,
35    --primary key(Player_Status_ID),
36    Foreign key (Player_ID) references Players(Player_ID)
37 )
```

```
38
39  create table Team_Stats
40  (
41    --Team_Stats_ID int,    NOT NEEDED
42    Team_ID int,
43    Wins int,
44    Losses int,
45    Total_Points int,
46    Steals int,
47    Rebounds int,
48    Assists int,
49    Blocks int,
50    FT_Percentage int,
51    --Primary Key(Team_Stats_ID),
52    Foreign Key(Team_ID) references Teams (Team_ID)
53  )
54
55  create table Stadium
56  (
57    Stadium_ID int,
58    Capacity int,
59    Stadium_Name varchar(100),
60    Stadium_City varchar(100),
61    Primary Key(Stadium_ID)
62  )
63
64
65  create table Schedule
66  (
67    Match_ID int,
68    Stadium_ID int,
69    Team_ID_1 int,
70    Team_ID_2 int,
71    Match_Name varchar(100),
72    DateTime_Schedule date,
73    Primary Key(Match_ID),
74    Foreign Key(Stadium_ID) references Stadium (Stadium_ID)
75  )
```

## Populate Tables

```
79  insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
80  values (1, 'LeBron James', 202, 250, 'SF', 6, 'Lakers');
81  insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
82  values (2, 'Anthony Davis', 208, 253, 'PF', 3, 'Lakers');
83  insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
84  values (3, 'Russel Westbrook', 190, 200, 'PG', 0, 'Lakers');
85  insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
86  values (4, 'Dwight Howard', 208, 265, 'C', 39, 'Lakers');
87  insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
88  values (5, 'Kent Bazemore', 193, 195, 'SG', 9, 'Lakers');
89
90  insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
91  values (6, 'Stephen Curry', 190, 185, 'PG', 30, 'Warriors');
92  insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
93  values (7, 'Klay Thompson', 198, 215, 'SG', 11, 'Warriors');
94  insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
95  values (8, 'Draymond Green', 198, 230, 'PF', 23, 'Warriors');
96  insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
97  values (9, 'Andrew Wiggins', 201, 197, 'PF', 22, 'Warriors');
98  insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
99  values (10, 'Kevon Looney', 206, 222, 'C', 5, 'Warriors');

100 insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
101 values (11, 'Chris Boucher', 206, 200, 'C', 25, 'Raptors');
102 insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
103 values (12, 'Fred VanVleet', 185, 197, 'SG', 23, 'Raptors');
104 insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
105 values (13, 'OG Anunoby', 201, 232, 'SF', 3, 'Raptors');
106 insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
107 values (14, 'Goran Dragic', 190, 190, 'PG', 1, 'Raptors');
108 insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
109 values (15, 'Pascal Siakam', 206, 230, 'PF', 43, 'Raptors');

110 insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
111 values (16, 'Kevin Durant', 208, 240, 'PF', 7, 'Nets');
112 insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
113 values (17, 'James Harden', 196, 220, 'PG', 13, 'Nets');
114 insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
115 values (18, 'Blake Griffin', 206, 250, 'PF', 2, 'Nets');
116 insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
117 values (19, 'LaMarcus Aldridge', 211, 250, 'C', 21, 'Nets');
118 insert into Players (Player_ID, Name, Height, Weight, Position, Jersey_No, Team_name )
119 values (20, 'Kyrie Irving', 188, 195, 'PG', 11, 'Nets');
```

```
124  insert into Teams (Team_ID,Team_Name,City_State,gm)
125  values (1,'Lakers','Los Angeles','Rob Pelinka');
126  insert into Teams (Team_ID,Team_Name,City_State,gm)
127  values (2,'Warriors','San Francisco','Bob Myers');
128  insert into Teams (Team_ID,Team_Name,City_State,gm)
129  values (3,'Raptors','Toronto','Bobby Webster');
130  insert into Teams (Team_ID,Team_Name,City_State,gm)
131  values (4,'Nets','Brooklyn','Sean Marks');
```

```

134 insert into Player_Stats
135 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
136 values (1, 26, 6, 5.5, 1.1, 1.7, 35471, 1314, 76.5);
137 insert into Player_Stats
138 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
139 values (2, 26.2, 2.7, 11.2, 2.5, 0.7, 13620, 570, 73.7);
140 insert into Player_Stats
141 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
142 values (3, 18, 8.7, 8.8, 0.2, 1.7, 21965, 949, 50);
143 insert into Player_Stats
144 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
145 values (4, 4.3, 0.5, 4, 0.5, 1, 19139, 1188, 81.8);
146 insert into Player_Stats
147 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
148 values (5, 7.2, 1.3, 2, 0.5, 1, 5325, 624, 66.7);
149
150 insert into Player_Stats
151 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
152 values (6, 28.7, 6.5, 7.5, 0.3, 1.5, 18606, 768, 97.1);
153 insert into Player_Stats
154 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
155 values (7, 19.5, 2.3, 3.5, 0.6, 1.1, 11995, 615, 84.4);
156 insert into Player_Stats
157 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
158 values (8, 9.3, 7, 8.7, 1.2, 1.3, 5670, 645, 36.8);
159 insert into Player_Stats
160 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
161 values (9, 16.2, 1.7, 4.5, 0.5, 1.2, 10360, 531, 69.2);
162 insert into Player_Stats
163 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
164 values (10, 6.2, 0.7, 5.5, 0.5, 0.4, 1267, 291, 70);
165
166 insert into Player_Stats
167 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
168 values (11, 6.1, 0.3, 4.4, 1, 1, 1365, 158, 62.5);
169 insert into Player_Stats
170 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
171 values (12, 17, 7.3, 5, 0.4, 1.6, 3554, 290, 70.6);
172 insert into Player_Stats
173 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
174 values (13, 18, 2.9, 5.6, 0.7, 1.7, 2448, 260, 87.5);
175 insert into Player_Stats
176 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
177 values (14, 6, 1.5, 2.3, 0.3, 0.8, 12070, 871, 100);
178 insert into Player_Stats
179 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
180 values (15, 14.3, 2.7, 5.8, 0.7, 0.9, 4739, 332, 78.2);

```

```

182 insert into Player_Stats
183 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
184 values (16, 27.7, 5.3, 8.9, 0.9, 0.9, 24077, 891, 79.5);
185 insert into Player_Stats
186 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
187 values (17, 18.6, 8.6, 7.6, 1.0, 1.0, 22175, 884, 86.5);
188 insert into Player_Stats
189 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
190 values (18, 4.8, 0.7, 4.8, 0.3, 0.5, 14013, 674, 77.8);
191 insert into Player_Stats
192 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
193 values (19, 12.7, 0.3, 5.0, 1.2, 0.3, 20027, 1035, 61.5);
194 insert into Player_Stats
195 (Player_ID, Avg_Points, Assists, Rebounds, Blocks, Steals, Total_Points, Games_Played, FreeThrowPercent)
196 values (20, 22.8, 5.7, 3.8, 0.7, 1.4, 13293, 582, 88.1);

```

```

199  insert into Team_Stats
200  (Team_ID,Wins,Losses,Total_Points,Steals,Rebounds,Assists,Blocks,FT_Percentage)
201  values (1,4,1,900,200,775,1000,80,65);
202  insert into Team_Stats
203  (Team_ID,Wins,Losses,Total_Points,Steals,Rebounds,Assists,Blocks,FT_Percentage)
204  values (2,3,3,870,220,705,1350,50,79);
205  insert into Team_Stats
206  (Team_ID,Wins,Losses,Total_Points,Steals,Rebounds,Assists,Blocks,FT_Percentage)
207  values (3,3,2,790,300,580,1100,70,71);
208  insert into Team_Stats
209  (Team_ID,Wins,Losses,Total_Points,Steals,Rebounds,Assists,Blocks,FT_Percentage)
210  values (4,5,2,905,255,620,1040,90,73);
211
212
213 Insert into Stadium (Stadium_ID, Capacity, Stadium_Name, Stadium_City)
214 Values (1,20000,'Staples Center', 'Los Angeles');
215 Insert into Stadium (Stadium_ID,Capacity,Stadium_Name, Stadium_City)
216 Values (2,18000,'Chase Center', 'San Francisco');
217 Insert into Stadium (Stadium_ID,Capacity,Stadium_Name, Stadium_City)
218 Values (3,19800,'Scotiabank Arena', 'Toronto');
219 Insert into Stadium (Stadium_ID,Capacity,Stadium_Name, Stadium_City)
220 Values (4,19000,'Barclays Center', 'Brooklyn');
221 Insert into Stadium (Stadium_ID,Capacity,Stadium_Name, Stadium_City)
222 Values (5,20800,'Madison Square Garden', 'New York');
223 Insert into Stadium (Stadium_ID,Capacity,Stadium_Name, Stadium_City)
224 Values (6,23500,'United Center', 'Chicago');
225
226
227 Insert into Schedule (Match_ID, Stadium_ID, Team_ID_1,Team_ID_2, Match_Name, DateTime_Schedule)
228 Values (1,1,1,2,'Lakers vs Warriors',TO_DATE('10/08/2021', 'DD/MM/YYYY'));
229 Insert into Schedule (Match_ID, Stadium_ID, Team_ID_1,Team_ID_2, Match_Name, DateTime_Schedule)
230 Values (2,3,3,4,'Raptors vs Nets',TO_DATE('14/08/2021', 'DD/MM/YYYY'));
231 Insert into Schedule (Match_ID, Stadium_ID, Team_ID_1,Team_ID_2, Match_Name, DateTime_Schedule)
232 Values (3,5,3,1,'Raptors vs Lakers',TO_DATE('19/08/2021', 'DD/MM/YYYY'));
233 Insert into Schedule (Match_ID, Stadium_ID, Team_ID_1,Team_ID_2, Match_Name, DateTime_Schedule)
234 Values (4,4,4,2,'Nets vs Warriors',TO_DATE('22/08/2021', 'DD/MM/YYYY'));
235 Insert into Schedule (Match_ID, Stadium_ID, Team_ID_1,Team_ID_2, Match_Name, DateTime_Schedule)
236 Values (5,6,1,4,'Lakers vs Nets',TO_DATE('27/08/2021', 'DD/MM/YYYY'));
237 Insert into Schedule (Match_ID, Stadium_ID, Team_ID_1,Team_ID_2, Match_Name, DateTime_Schedule)
238 Values (6,2,2,3,'Warriors vs Raptors',TO_DATE('02/09/2021', 'DD/MM/YYYY'));

```

## Views and Simple/Advanced Queries

---

```
SELECT Name FROM Players WHERE height > 200;

SELECT Team_ID, Team_Name, gm FROM Teams WHERE Teams.City_State = 'Toronto';

SELECT Player_Stats.Player_ID, Players.Name, Position, Jersey_No
FROM Player_Stats, Players
WHERE Player_Stats.Player_ID=Players.Player_ID and Player_Stats.Avg_Points > 20;

SELECT Team_Stats.Team_ID, Teams.Team_Name, Losses
FROM Team_Stats, Teams
WHERE Team_Stats.Team_ID=Teams.Team_ID and Team_Stats.Losses >= 2;

SELECT Stadium_ID, Stadium_Name, Capacity
FROM Stadium WHERE Stadium_City = 'New York'

SELECT Schedule.Match_ID, Schedule.Match_Name, Stadium.Stadium_Name, Schedule.DateTime_Schedule
FROM Schedule, Stadium
WHERE Schedule.Stadium_ID = Stadium.Stadium_ID and Team_ID_1 = 2 and Team_ID_2 = 3;

SELECT DISTINCT Height FROM Players;

SELECT Weight, Name, Player_ID FROM Players ORDER BY Weight DESC;

SELECT COUNT(Assists), Player_ID FROM Player_Stats GROUP BY Player_ID ORDER BY COUNT(Assists) DESC;
```

*Overview of all query tables*

1.

```
SELECT Name FROM Players WHERE height > 200;
```

Simple query from ‘Players’ table displaying player names with height over 200 cm

NAME
1 LeBron James
2 Anthony Davis
3 Dwight Howard
4 Andrew Wiggins
5 Kevon Looney
6 Chris Boucher
7 OG Anunoby
8 Pascal Siakam
9 Kevin Durant
10 Blake Griffin
11 LaMarcus Aldridge

2.

```
SELECT Team_ID, Team_Name, gm FROM Teams WHERE Teams.City_State = 'Toronto';
```

Simple query from 'Teams' table displaying Team ID, name of team, and general manager of a team that's home city is Toronto

	TEAM_ID	TEAM_NAME	GM
1	3	Raptors	Bobby Webster

3.

```
SELECT Player_Stats.Player_ID, Players.Name, Position, Jersey_No
```

```
FROM Player_Stats, Players
```

```
WHERE Player_Stats.Player_ID=Players.Player_ID and Player_Stats.Avg_Points > 20;
```

Simple **JOIN** query from 'Player\_Stats' table displaying player ID, player names, player position, and jersey number for players whose average points is greater than 20 points. This query also uses JOIN with 'Player\_Stats.Player\_ID=Players.Player\_ID' which uses 'player\_ID' as primary key and foreign key to join the 'Player\_Stats' and 'Players' table

	PLAYER_ID	NAME	POSITION	JERSEY_NO
1	1	LeBron James	SF	6
2	2	Anthony Davis	PF	3
3	6	Stephen Curry	PG	30
4	16	Kevin Durant	PF	7
5	20	Kyrie Irving	PG	11

4.

```
SELECT Team_Stats.Team_ID, Teams.Team_Name, Losses
```

```
FROM Team_Stats, Teams
```

```
WHERE Team_Stats.Team_ID=Teams.Team_ID and Team_Stats.Losses >= 2;
```

Simple **JOIN** query from 'Player\_Stats' table displaying team ID, team name, and number of losses for teams that have 2 or more losses. This query also uses JOIN clause joining 'Team\_Stats' and 'Teams' tables through primary key and foreign key 'Team\_ID'.

	TEAM_ID	TEAM_NAME	LOSSES
1	2	Warriors	3
2	3	Raptors	2
3	4	Nets	2

5.

```
SELECT Stadium_ID, Stadium_Name, Capacity  
FROM Stadium WHERE Stadium_City = 'New York'
```

Simple query from ‘Stadium’ table displaying stadium ID, stadium name, and stadium capacity where the stadium is located in the city of New York.

STADIUM_ID	STADIUM_NAME	CAPACITY
1	Madison Square Garden	20800

6.

```
SELECT Schedule.Match_ID, Schedule.Match_Name, Stadium.Stadium_Name,  
Schedule.DateTime_Schedule  
FROM Schedule, Stadium  
WHERE Schedule.Stadium_ID = Stadium.Stadium_ID and Team_ID_1 = 2 and Team_ID_2 = 3;
```

Simple **JOIN** query from ‘Schedule’ table displaying match ID, match name, stadium name where match is held, and date of match for Warriors versus Raptors match. ‘Team\_ID\_1 = 2’ refers to Warriors and Team\_ID\_2 = 3 refers to Raptors. This query also uses JOIN clause joining ‘Schedule’ and ‘Stadium’ tables through primary key and foreign key ‘Stadium\_ID’.

MATCH_ID	MATCH_NAME	STADIUM_NAME	DATETIME_SCHEDULE
1	Warriors vs Raptors	Chase Center	21-09-02

## Advanced Queries Using DISTINCT, ORDER BY, GROUP BY, and COUNT

1.

```
SELECT DISTINCT Height FROM Players;
```

This query uses **DISTINCT** statement to display all the distinct heights of players from ‘Players’ table. Therefore there are 11 different distinct heights out of the 20 players.

	HEIGHT
1	198
2	185
3	211
4	193
5	201
6	206
7	188
8	202
9	190
10	196
11	208

2.

```
SELECT Weight, Name, Player_ID FROM Players ORDER BY Weight DESC;
```

This query uses **ORDER BY** statement to display all player's names, weight and ID from the 'Players' table. The weight of each player is ordered by descending weight (most weight to least).

	WEIGHT	NAME	PLAYER_ID
1	265	Dwight Howard	4
2	253	Anthony Davis	2
3	250	LeBron James	1
4	250	LaMarcus Aldridge	19
5	250	Blake Griffin	18
6	240	Kevin Durant	16
7	232	OG Anunoby	13
8	230	Draymond Green	8
9	230	Pascal Siakam	15
10	222	Kevon Looney	10
11	220	James Harden	17
12	215	Klay Thompson	7
13	200	Chris Boucher	11
14	200	Russel Westbrook	3
15	197	Andrew Wiggins	9
16	197	Fred Vanvleet	12
17	195	Kyrie Irving	20
18	195	Kent Bazemore	5
19	190	Goran Dragic	14
20	185	Stephen Curry	6

3.

```
SELECT COUNT(Assists), Player_ID FROM Player_Stats GROUP BY Player_ID ORDER BY COUNT(Assists) DESC;
```

This query uses **COUNT** statement to display the number of 'Assists' values each player has from the 'Player\_Stats' table. Since each player has only one value for 'Assists', the output table

shows '1' for each player. **GROUP BY** is also used to group rows that have the same values into summary rows.

	COUNT(ASSISTS)	PLAYER_ID
1	1	1
2	1	19
3	1	11
4	1	13
5	1	2
6	1	14
7	1	20
8	1	4
9	1	5
10	1	8
11	1	17
12	1	3
13	1	7
14	1	18
15	1	9
16	1	10
17	1	12
18	1	15
19	1	16
20	1	6

# Unix shell Implementation

---

## *Drop Tables*

```
#!/bin/sh
#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib
sqlplus64 "wjalali/02189545@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerson.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF
DROP TABLE Players CASCADE CONSTRAINTS;
DROP TABLE Teams CASCADE CONSTRAINTS;
DROP TABLE Player_Stats CASCADE CONSTRAINTS;
DROP TABLE Team_Stats CASCADE CONSTRAINTS;
DROP TABLE Schedule CASCADE CONSTRAINTS;
DROP TABLE Stadium CASCADE CONSTRAINTS;

exit;
EOF
```

## *Bash Script*

```
#!/bin/sh
MainMenu() {
    while [ "$CHOICE" != "START" ]
    do
        echo " ++++++"
        echo "| Oracle All Inclusive Tool"
        echo "| Main Menu - Select Desired Operation(s):"
        echo "| <CTRL-Z Anytime to Enter Interactive CMD Prompt>"
        echo " ++++++"
        echo ""
        echo " $IS_SELECTEDM M) View Manual"
        echo ""
        echo " $IS_SELECTED1 1) Drop Tables"
        echo " $IS_SELECTED2 2) Create Tables"
        echo " $IS_SELECTED3 3) Populate Tables"
        echo " $IS_SELECTED4 4) Query Tables"
        echo ""
        echo " $IS_SELECTEDX X) Force/Stop/Kill Oracle DB"
        echo ""
        echo " $IS_SELECTEDE E) End/Exit"
        echo ""
        echo "Choose: "

        read CHOICE
        if [ "$CHOICE" = "0" ]
        then
            echo "Nothing Here"
        elif [ "$CHOICE" = "1" ]
        then
            bash drop_tables.sh
            Pause
        elif [ "$CHOICE" = "2" ]
        then
            bash create_tables.sh
            Pause
        elif [ "$CHOICE" = "3" ]
        then
            bash populate_tables.sh
            Pause
        elif [ "$CHOICE" = "4" ]
        then
            bash query_tables.sh
            Pause
        elif [ "$CHOICE" = "E" ]
        then
            exit
        fi
    done
}
```

```
##--COMMENTS BLOCK--  
# Main Program  
##--COMMENTS BLOCK--  
  
ProgramStart()  
{  
    StartMessage  
    while [ 1 ]  
    do  
        MainMenu  
        done  
    }  
  
ProgramStart
```

*Template code provided from D2L*

```
=====|Oracle All Inclusive Tool|  
| Main Menu - Select Desired Operation(s): |  
|<CTRL-Z Anytime to Enter Interactive CMD Prompt> |  
-----  
M) View Manual  
  
1) Drop Tables  
2) Create Tables  
3) Populate Tables  
4) Query Tables  
  
X) Force/Stop/Kill Oracle DB  
  
E) End/Exit  
Choose:
```

*Menu of unix shell ran through VS Code terminal*

# Functional Dependencies

---

## Functional Dependencies

**Players(Player\_ID, Name, Height, Weight, Position, Jersy\_No, Team\_name)**

Player\_ID → Name

Player\_ID → Height

Player\_ID → Weight

Player\_ID → Position

Player\_ID → Jersy\_No

Player\_ID → Team\_name

**Teams(Team\_ID, Team\_Name, City\_State, gm)**

Team\_ID → Team\_Name

Team\_ID → City\_State

Team\_ID → gm

gm → City\_State, Team\_Name (since gm's are unique to each team. Knowing gm would mean you know the City\_State and Team\_Name)

Team\_Name → City\_State, gm (since Team\_Name is also unique to each team. Knowing Team\_Name would mean you know the City\_State and Team\_Name)

**Players\_Stats(Player\_ID, Blocks, Steals, Avg\_Points, Total\_Points, Games\_Played, assists, rebounds, FreeThrowPercent)**

No FD, because table has a many to many relationship.

**Team\_Stats(Team\_ID, Wins, Losses, Total\_Points, Steals, Rebounds, Assists, Blocks, FT\_Percentage)**

No FD, because table has a many to many relationship.

**Stadium(Stadium\_ID, Capacity, Stadium\_Name, Stadium\_City)**

Stadium\_ID → Capacity

Stadium\_ID → Stadium\_Name

Stadium\_ID → Stadium\_City

**Schedule(Match\_ID, Stadium\_ID, Team\_ID\_1, Team\_ID\_2, Match\_Name, DateTime\_Schedule)**

Match\_ID → Team\_ID\_1

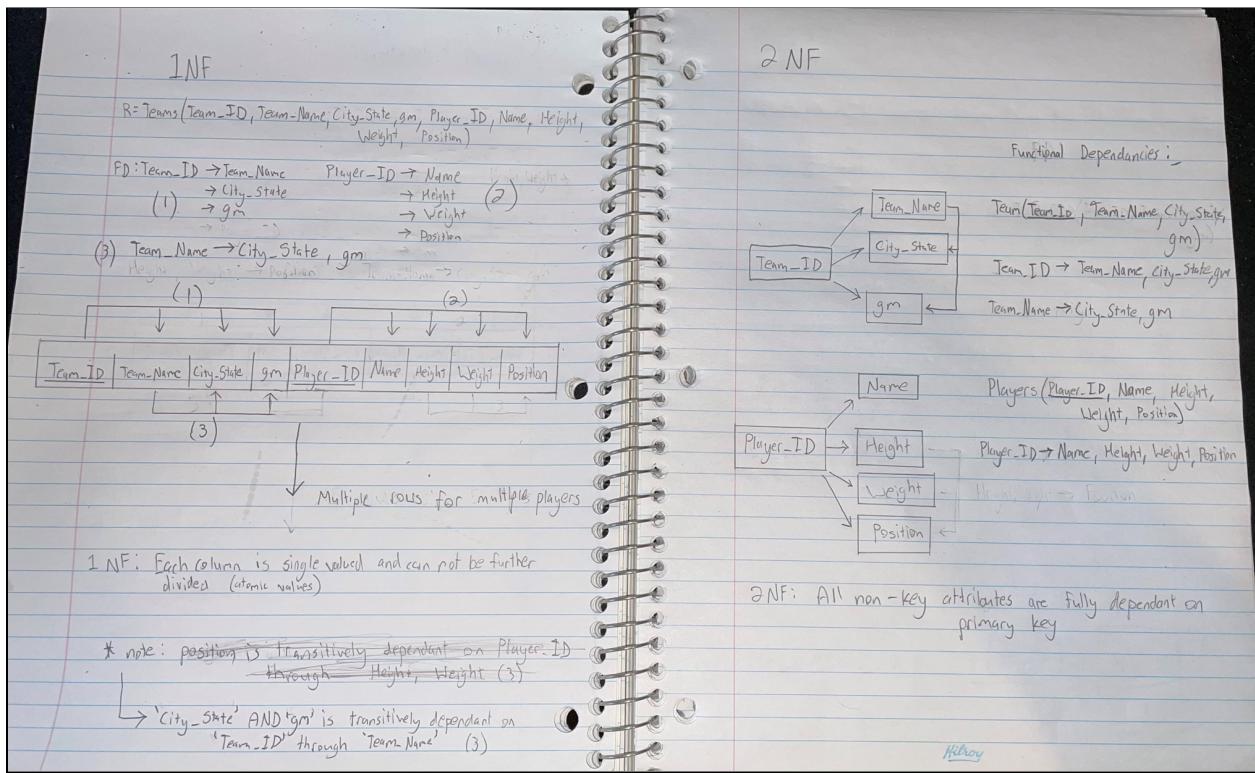
Match\_ID → Team\_ID\_2

Match\_ID → Match\_Name

Match\_ID → DateTime\_Schedule

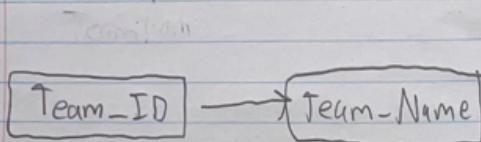
Match\_ID → Stadium\_ID (since the match determines which stadium game was played at)

## Normalization in 3NF

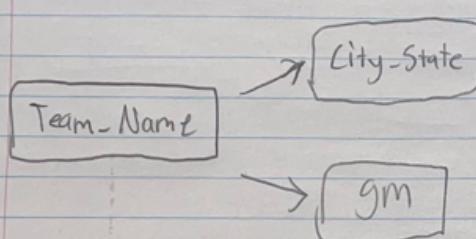


3NF

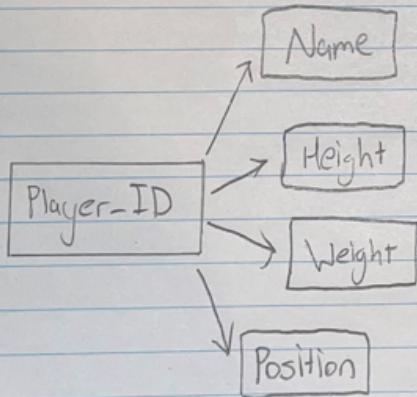
Functional Dependencies:



TeamName (Team-ID, Team-Name)  
TeamName (Team-ID, Team-Name)  
 $\text{Team-ID} \rightarrow \text{Team-Name}$   
 $\text{Team-ID} \rightarrow \text{Team-Name}$



TeamInfo (Team-Name, City-State,  
gm)



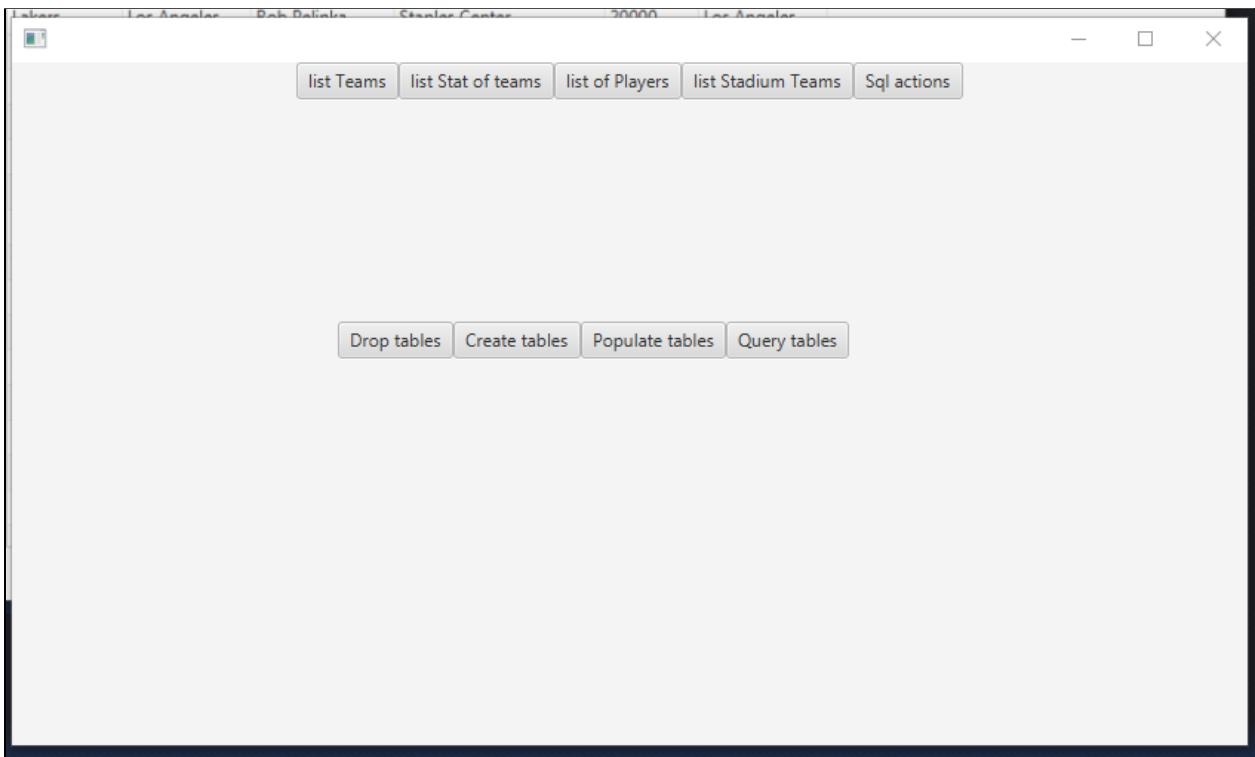
Players (Player-ID, Name, Height,  
Weight, Position)  
 $\text{Player-ID} \rightarrow \text{Name, Height, Weight, Position}$

3NF: No non-candidate key attribute is transitively dependent  
on any candidate key

Hilroy

## Graphical User Interface of Database

---



This screenshot shows a graphical user interface displaying a table of NBA team data. The table has columns: teamName, teamCity, gm, stadium, capacity, stadiumCity, and a final column which appears to be a summary or calculated value. The data rows are as follows:

teamName	teamCity	gm	stadium	capacity	stadiumCity	
Lakers	Los Angeles	Rob Pelinka	Staples Center	20000	Los Angeles	
Warriors	San Francisco	Bob Myers	Staples Center	20000	Los Angeles	
Raptors	Toronto	Bobby Webster	Scotiabank Arena	19800	Toronto	
Nets	Brooklyn	Sean Marks	Scotiabank Arena	19800	Toronto	
Raptors	Toronto	Bobby Webster	Madison Square Garden	20800	New York	
Lakers	Los Angeles	Rob Pelinka	Madison Square Garden	20800	New York	
Nets	Brooklyn	Sean Marks	Barclays Center	19000	Brooklyn	
Warriors	San Francisco	Bob Myers	Barclays Center	19000	Brooklyn	
Lakers	Los Angeles	Rob Pelinka	United Center	23500	Chicago	
Nets	Brooklyn	Sean Marks	United Center	23500	Chicago	
Warriors	San Francisco	Bob Myers	Chase Center	18000	San Francisco	
Raptors	Toronto	Bobby Webster	Chase Center	18000	San Francisco	

Lakers 4 1 900 200 80 1000 775 65

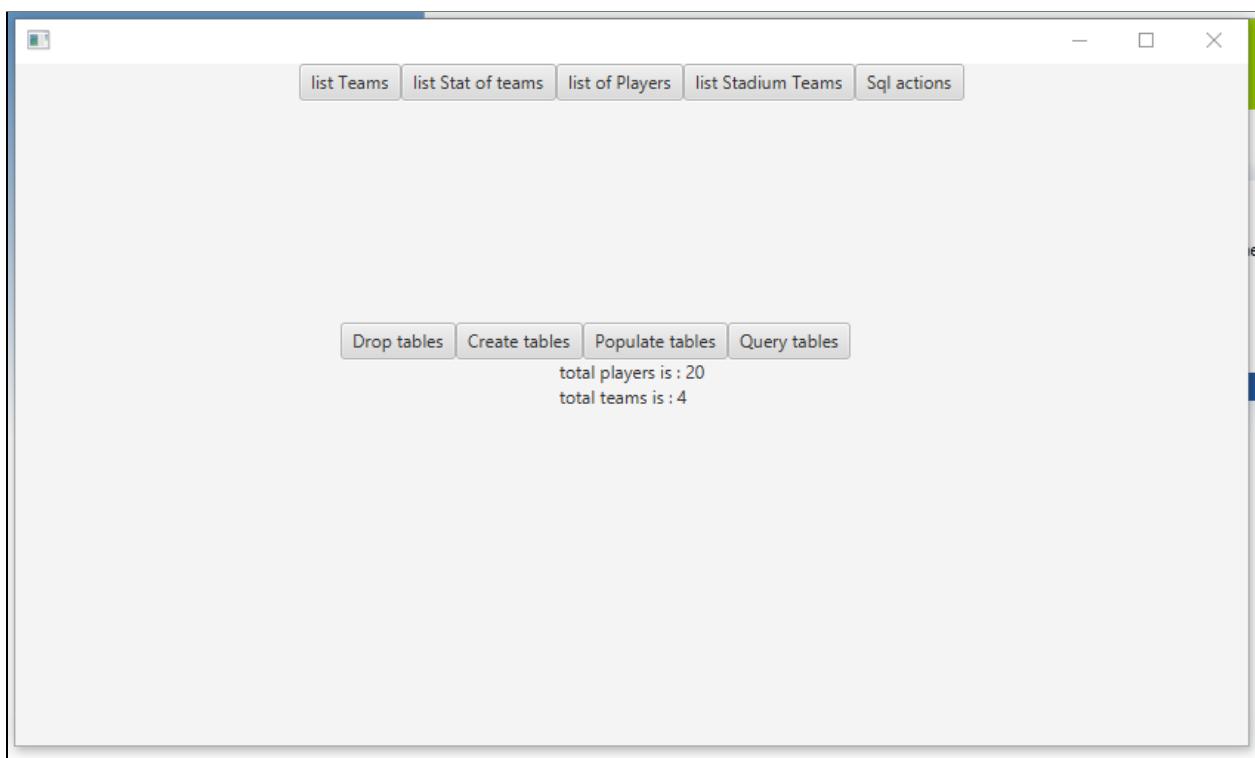
RANK	id	playerName	playerTeam	avgPoints	totalPoints	gamesPlayed	steals	blocks	assists	rebounds	ftPercent
1	4	Dwight Howard	Lakers	19139	1188	4	1	1	1	4	82
2	14	Goran Dragic	Raptors	12070	871	6	1	0	2	2	100
3	18	Blake Griffin	Nets	14013	674	5	1	0	1	5	78
4	5	Kent Bazemore	Lakers	5325	624	7	1	1	1	2	67
5	19	LaMarcus Aldridge	Nets	20027	1035	13	0	1	0	5	62
6	8	Draymond Green	Warriors	5670	645	9	1	1	7	9	37
7	3	Russel Westbrook	Lakers	21965	949	18	2	0	9	9	50
8	1	LeBron James	Lakers	35471	1314	26	2	1	6	6	77
9	10	Kevon Looney	Warriors	1267	291	6	0	1	1	6	70
10	17	James Harden	Nets	22175	884	19	1	1	9	8	87
11	9	Andrew Wiggins	Warriors	10360	531	16	1	1	2	5	69
12	16	Kevin Durant	Nets	24077	891	28	1	1	5	9	80
13	7	Klay Thompson	Warriors	11995	615	20	1	1	2	4	84
14	6	Stephen Curry	Warriors	18606	768	29	2	0	7	8	97
15	11	Chris Boucher	Raptors	1365	158	6	1	1	0	4	63
16	20	Kyrie Irving	Nets	13293	582	23	1	1	6	4	88

Lakers 4 1 900 200 80 1000 775 65

RANK	name	wins	losses	totalPoints	steals	blocks	assists	rebounds	ftPercent	
1	Lakers	4	1	900	200	80	1000	775	65	
2	Raptors	3	2	790	300	70	1100	580	71	
3	Nets	5	2	905	255	90	1040	620	73	
4	Warriors	3	3	870	220	50	1350	705	79	

A screenshot of a Windows application window titled "NBA Statistics". The window has a standard title bar with minimize, maximize, and close buttons. Below the title bar is a menu bar with five items: "list Teams", "list Stat of teams", "list of Players", "list Stadium Teams", and "Sql actions". The main area of the window contains a table with the following data:

RANK	NAME	W	L	PCT
1	Lakers	4	1	80.0
2	Nets	5	2	71.42857
3	Raptors	3	2	60.000004
4	Warriors	3	3	50.0



## Relational Algebra Notation

---

1. SELECT Name FROM Players WHERE height > 200;

**Π Name (σ height > 200(Players))**

2. SELECT Team\_ID, Team\_Name, gm FROM Teams WHERE Teams.City\_State = 'Toronto';

**Π Team\_ID, Team\_Name, gm (σ Teams.City\_State = 'Toronto'(Teams))**

3. SELECT Player\_Stats.Player\_ID, Players.Name, Position, Jersy\_NO  
FROM Player\_Stats, Players  
WHERE Player\_Stats.Player\_ID=Players.Player\_ID and  
Player\_Stats.Avg\_Points > 20;

**Π Player\_Stats.Player\_ID, Players.Name, Position, Jersy\_NO (σ Player\_Stats.Player\_ID = Players.Player\_ID AND Player\_Stats.Avg\_Points > 20 (Player\_Stats << Players))**

4. SELECT Team\_Stats.Team\_ID, Teams.Team\_Name, Losses  
FROM Team\_Stats, Teams  
WHERE Team\_Stats.Team\_ID=Teams.Team\_ID and Team\_Stats.Losses >= 2;

**Π Team\_Stats.Team\_ID, Teams.Team\_Name, Losses (σ Team\_Stats.Team\_ID = Teams.Team\_ID AND Team\_Stats.Losses >= 2 (Team\_Stats << Teams))**

5. SELECT Stadium\_ID, Stadium\_Name, Capacity  
FROM Stadium WHERE Stadium\_City = 'New York'

**Π Stadium\_ID, Stadium\_Name, Capacity (σ Stadium\_City = 'New York'(Stadium))**

6. SELECT Schedule.Match\_ID, Schedule.Match\_Name, Stadium.Stadium\_Name,  
Schedule.DateTime\_Schedule  
FROM Schedule, Stadium

WHERE Schedule.Stadium\_ID = Stadium.Stadium\_ID and Team\_ID\_1 = 2 and Team\_ID\_2 = 3;

**II** Schedule.Match\_ID, Schedule.Match\_Name, Stadium.Stadium\_Name,  
Schedule.DateTime\_Schedule ( $\sigma$  Schedule.Stadium\_ID = Stadium.Stadium\_ID AND  
Team\_ID\_1 = 2 and Team\_ID\_2 = 3; (Schedule >< Stadium ))

7. SELECT DISTINCT Height FROM Players;

**II** Height (Players)

8. SELECT Weight, Name, Player\_ID FROM Players ORDER BY Weight DESC;

**II** Weight, Name, Player\_ID (Players)

9. SELECT COUNT(Assists), Player\_ID FROM Player\_Stats GROUP BY Player\_ID ORDER BY COUNT(Assists) DESC;

Player\_ID F COUNT Assists (Player\_Stats)

## Concluding Remarks

---

Working on this basketball league management database has provided us with a solid foundation in database design and implementation. With the theories we learned regarding entity-relationship diagrams, relational schema design, functional dependencies, normalization, etc., we were able to turn various pieces of data into a useful and accessible database that anyone can access.

On the technical side, we became accustomed to using SQL and the services provided by Oracle. Through this, we learned what it was like to create tables, drop tables, insert data, and query information. Making a GUI using Java also familiarized us with how a front-end interface connects and interacts with a back-end database.

All in all, it was very informative working on this basketball league database. It allowed us to apply the knowledge and skills acquired in this course into a real project that's application can be applied in real life. We see databases used in our everyday lives and we now have technical insight on how these systems work.