University of British Columbia
Department of Computer Science

CPSC 304 2020

Summer Term1

Group Project - Implementation of a Relational Database

Project Title:	NBA Database
Project Milestone:	Milestone 4a - Implementation

#	Student Name	Student Number	Email Address
1	Mike Powar	81430993	mike.powar@gmail.com
2	Daniel Au	78934767	ubc.danielau@gmail.com
3	Justin Sohnne	77575520	justin.sohn08@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 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

SOURCE CODE

https://github.com/MikePowar/nbadb

PROJECT DESCRIPTION

The National Basketball Association's popularity spans the globe and it generates billions in revenue to be distributed amongst its players, the sports franchises that employ them, the industries connected to them, and the hard working people employed by them.

While the race to harness the power of data seems all the rage in this moment, the importance of it for the sport goes back to the initial days of the leagues formation. The statistics generated by the play on the court has been tracked from its inception and makes for great conversations amongst fans comparing the records of players in the past with those in the present.

With this project, our group wanted to understand the work that goes into building the databases that tell the stories of the league through data.

How is fourty-eight minutes of battling on a hardwood floor for eighty-two games a year, by thirty plus teams, played by hundreds of players each year represented in a database ? How is that data made useable and searchable for fans and players of the game to understand what happens on the court in ways the eye cannot see ?

We built an active database with an administrative feature to allow the entry of new players as they enter the league and edit their information as they change over time. The database provides high-level views of what happens in a season from league, team, and individual player perspectives. Users are provided with overall data for team and player categories, and then are given options to interact with that data to answer a range of questions fans of the league would have using the queries we learned to build in the course:

1. *Selection* - What teams played in the league during a particular season ?
2. *Projection* - Who lead the league in Points, Rebounds, Minutes played, 3-pointers, Steals, Assists ?
3. *Join* - Can I see the names of the full roster for my favorite team in a particular year ? What about the coach ? Who was the owner of the team that year ?
4. *Nested Aggregation* - Who are the elite of the elite ? Can I get a list of the players who perform above average for a particular statistical category ?

The queries above are highlights our team enjoyed applying the course material to. Given the scope of the project, we focused our data on one season, 2018-2019, five teams, their full rosters, and five games they played against one another.

This project is definitely one we hope to return to, add more data, and increase its complexity in hopes of building on the relational database skills we developed over the few weeks.

SCHEMA COMPARISON

Going into the project, we knew there could be many directions in which to focus our database on related to the NBA. We decided to focus on capturing what the general public would care about the most - packaging game summaries as well as player and team performance along with some useful details into our application on a per season basis. After our normalized schema was completed, our initial goal hadn't made any pivots, thus not much changed going into our implementation. The main notable change was updating domain types to strengthen our input constraints. Features such as age or height share the same number of decimal places among all examples, thus we make specific this constraint.

In addition, we also originally had the match schema containing a result attribute, which would store the number of points scored by both teams (e.g. '125-115'). We identified that formatting of the score wasn't well defined, and anomalies could arise from storing the data this way. As a solution, we decided to split this into two integer attributes, one for the score of one team, and one for the score of the other team. In doing this, we were able to come up with a rather interesting query involving division that observed which teams have defeated all other teams at least once in a season (At home or as a visiting team).

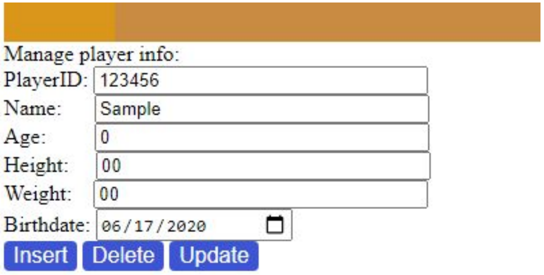
SQL QUERIES

Static Table Queries

```
SELECT * FROM Player;
SELECT * FROM Season;
SELECT * FROM Team;
SELECT * FROM playerstatistic;
```

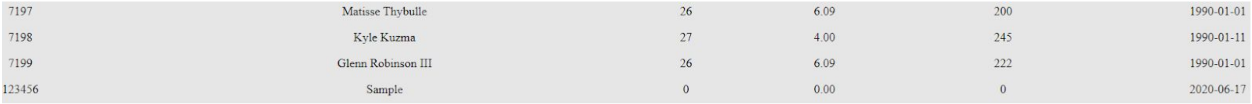
Insertion Query

```
INSERT INTO player (playerID, name, age, height, weight, dob)
VALUES ($id, '$name', $age, $height, $weight, '$birthdate');
```



Manage player info:

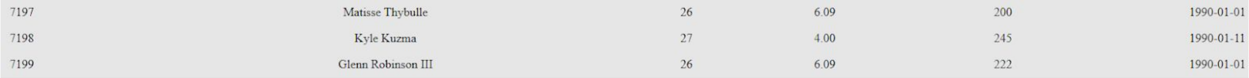
PlayerID: 123456
Name: Sample
Age: 0
Height: 00
Weight: 00
Birthdate: 06/17/2020



7197	Matisse Thybulle	26	6.09	200	1990-01-01
7198	Kyle Kuzma	27	4.00	245	1990-01-11
7199	Glenn Robinson III	26	6.09	222	1990-01-01
123456	Sample	0	0.00	0	2020-06-17

Delete Query

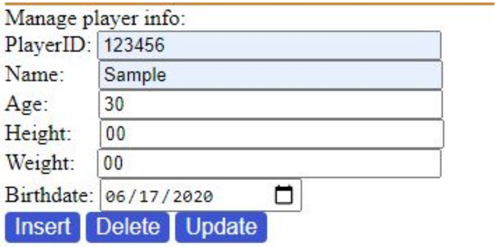
DELETE FROM player
WHERE playerID = $id;



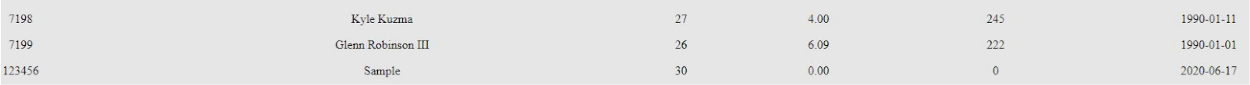
7197	Matisse Thybulle	26	6.09	200	1990-01-01
7198	Kyle Kuzma	27	4.00	245	1990-01-11
7199	Glenn Robinson III	26	6.09	222	1990-01-01

Update Query

UPDATE player SET playerID = $id, name = '$name', age = $age, height = $height, weight = $weight, dob = '$birthdate' WHERE playerID = $id;



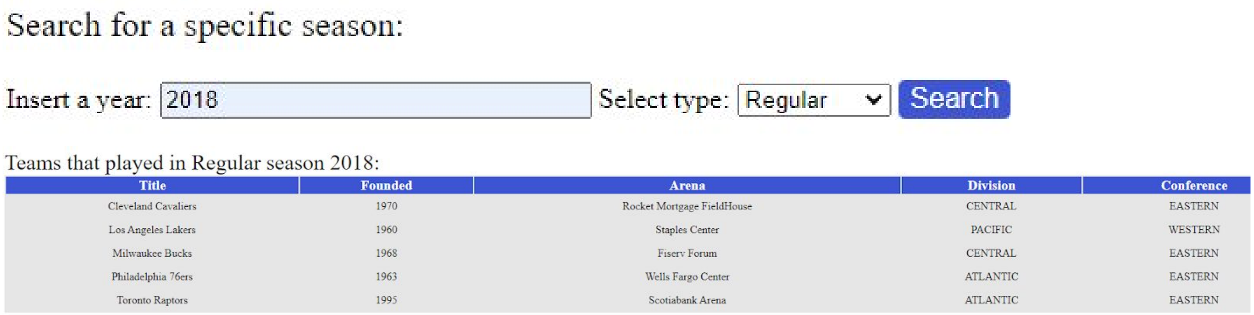
Manage player info:
PlayerID: 123456
Name: Sample
Age: 30
Height: 00
Weight: 00
Birthdate: 06/17/2020



7198	Kyle Kuzma	27	4.00	245	1990-01-11
7199	Glenn Robinson III	26	6.09	222	1990-01-01
123456	Sample	30	0.00	0	2020-06-17

Selection Query

```
SELECT * FROM team
WHERE EXISTS
(SELECT ts.tname FROM TeamSeasons ts, Season s
WHERE $year = year(ts.startDate)
AND '$type' = s.seasonType);
```



Search for a specific season:

Insert a year: 2018 Select type: Regular

Teams that played in Regular season 2018:

Title	Founded	Arena	Division	Conference
Cleveland Cavaliers	1970	Rocket Mortgage FieldHouse	CENTRAL	EASTERN
Los Angeles Lakers	1960	Staples Center	PACIFIC	WESTERN
Milwaukee Bucks	1968	Fiserv Forum	CENTRAL	EASTERN
Philadelphia 76ers	1963	Wells Fargo Center	ATLANTIC	EASTERN
Toronto Raptors	1995	Scotiabank Arena	ATLANTIC	EASTERN

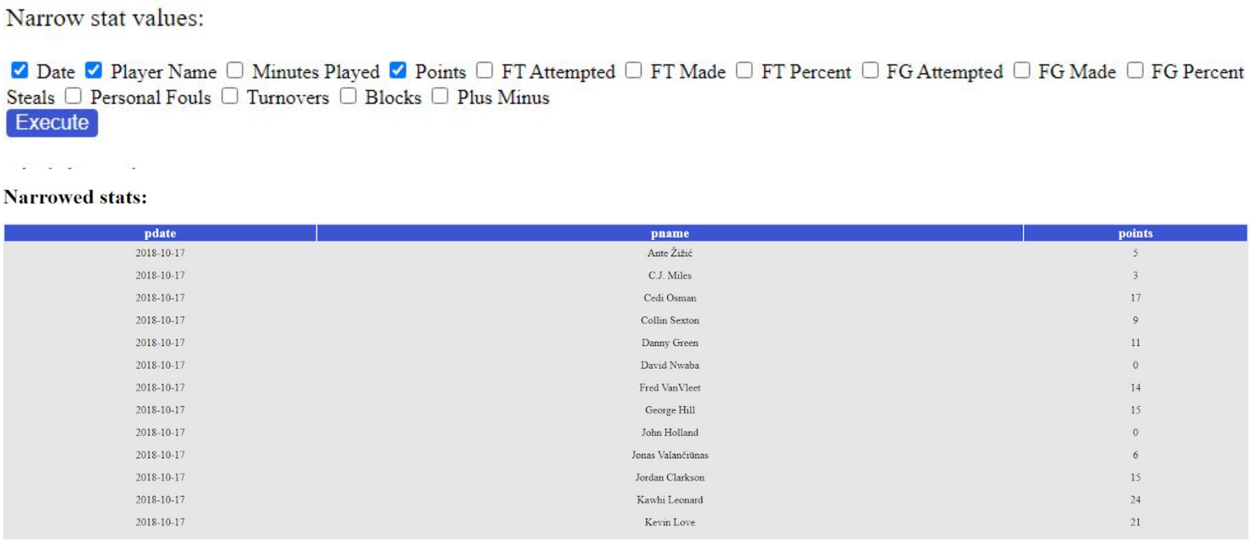
```
SELECT * FROM game
WHERE year(mdate) = 2018 or year(mdate) = 2019;

SELECT * FROM playerStatistic
WHERE pname = '$player';

SELECT * FROM teamstatistic
WHERE tname = '$team';
```

Projection Query

```
SELECT " . $check_list . " FROM playerstatistic
```



Narrow stat values:

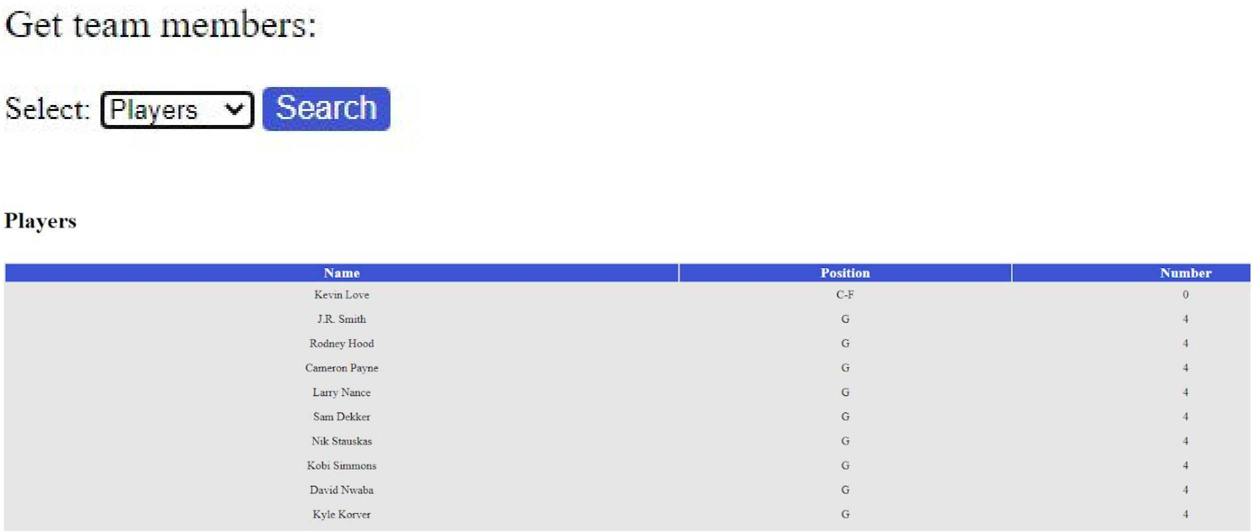
☑ Date ☑ Player Name □ Minutes Played ☑ Points □ FT Attempted □ FT Made □ FT Percent □ FG Attempted □ FG Made □ FG Percent
Steals □ Personal Fouls □ Turnovers □ Blocks □ Plus Minus

Narrowed stats:

pdate	pname	points
2018-10-17	Ante Žižić	5
2018-10-17	C.J. Miles	3
2018-10-17	Cedi Osman	17
2018-10-17	Collin Sexton	9
2018-10-17	Danny Green	11
2018-10-17	David Nwaba	0
2018-10-17	Fred VanVleet	14
2018-10-17	George Hill	15
2018-10-17	John Holland	0
2018-10-17	Jonas Valančiūnas	6
2018-10-17	Jordan Clarkson	15
2018-10-17	Kawhi Leonard	24
2018-10-17	Kevin Love	21

Join Query

```
SELECT p.name,pt.position,pt.number FROM playerteams pt
INNER JOIN player p ON pt.tname = '$team'
AND pt.playerID = p.playerID;
```



Get team members:

Select: Players

Players

Name	Position	Number
Kevin Love	C-F	0
J.R. Smith	G	4
Rodney Hood	G	4
Cameron Payne	G	4
Larry Nance	G	4
Sam Dekker	G	4
Nik Stauskas	G	4
Kobi Simmons	G	4
David Nwaba	G	4
Kyle Korver	G	4

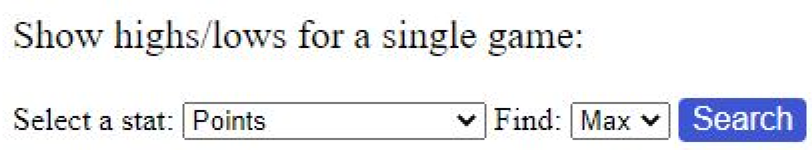
```
SELECT s.name,o.net_worth FROM `owner` o
INNER JOIN staff s ON s.tname = '$team'
AND o.staffID = s.staffID;
```

```
SELECT s.name,c.position,c.wins,c.losses
FROM `coach` c INNER JOIN staff s ON s.tname = '$team'
AND c.staffID = s.staffID;
```

Aggregate Query

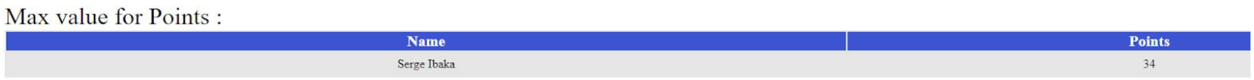
```
SELECT Avg($stat) as avg FROM playerstatistic;

SELECT pname,$stat FROM playerstatistic
WHERE $stat = (SELECT  $extrema($stat) FROM playerstatistic);
```



Show highs/lows for a single game:

Select a stat: Points Find: Max

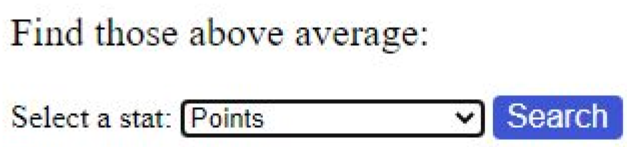


Max value for Points :

Name	Points
Serge Ibaka	34

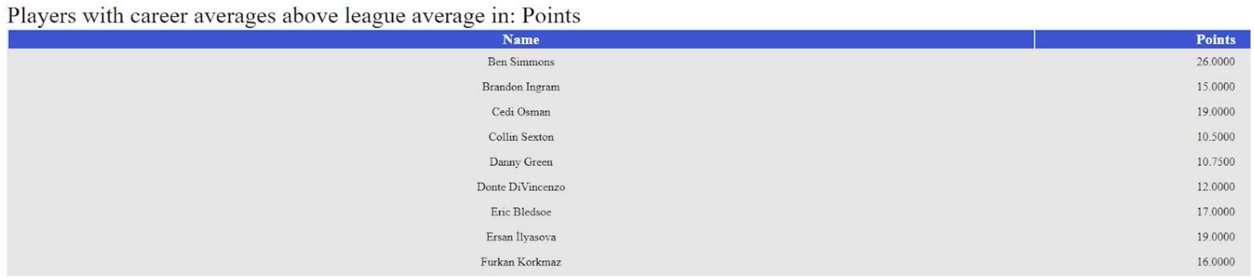
Nested Aggregation

```
SELECT pname, avg($stat) as stat
FROM playerstatistic
GROUP BY pname
HAVING stat > (SELECT avg($stat) FROM playerstatistic);
```



Find those above average:

Select a stat: Points

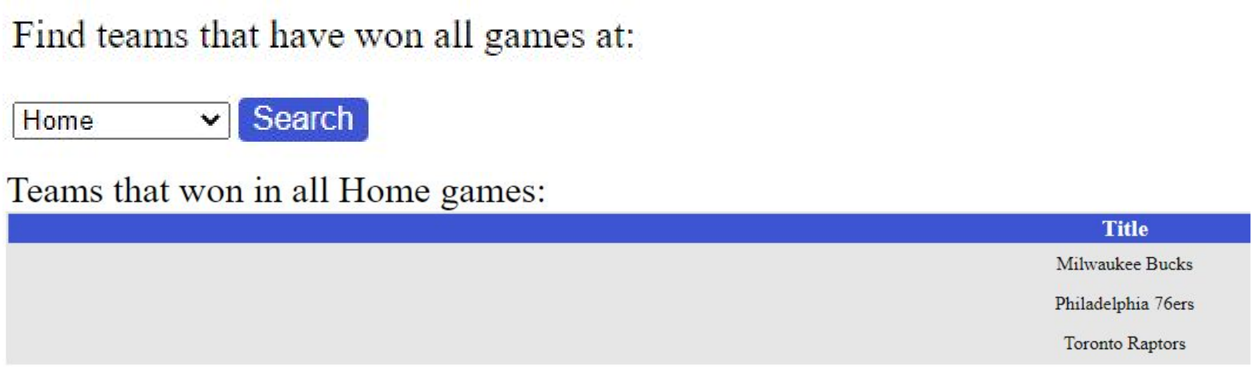


Players with career averages above league average in: Points

Name	Points
Ben Simmons	26.0000
Brandon Ingram	15.0000
Cedi Osman	19.0000
Collin Sexton	10.5000
Danny Green	10.7500
Donte DiVincenzo	12.0000
Eric Bledsoe	17.0000
Ersan İlyasova	19.0000
Furkan Korkmaz	16.0000

Division Query

```
SELECT t.title FROM Team t
WHERE NOT EXISTS
((SELECT g.awayTeamName FROM game g WHERE g.homeTeamName = t.title)
EXCEPT (SELECT h.awayTeamName FROM game h
WHERE h.homePTS > h.awayPTS
AND h.homeTeamName = t.title));
```



Find teams that have won all games at:

Home

Teams that won in all Home games:

Title
Milwaukee Bucks
Philadelphia 76ers
Toronto Raptors

```
SELECT t.title FROM Team t
WHERE NOT EXISTS
((SELECT g.homeTeamName FROM game g WHERE g.awayTeamName = t.title)
EXCEPT (SELECT h.homeTeamName FROM game h
WHERE h.homePTS < h.awayPTS
AND h.awayTeamName = t.title));
```

```
SELECT t.title FROM Team t
WHERE NOT EXISTS
((SELECT g.awayTeamName FROM game g WHERE g.homeTeamName = t.title)
EXCEPT (SELECT h.awayTeamName FROM game h
WHERE h.homePTS > h.awayPTS
AND h.homeTeamName = t.title))
INTERSECT
SELECT t.title FROM Team t
WHERE NOT EXISTS
((SELECT g.homeTeamName FROM game g WHERE g.awayTeamName = t.title)
EXCEPT (SELECT h.homeTeamName FROM game h
WHERE h.homePTS < h.awayPTS
AND h.awayTeamName = t.title));
```