NBA DATABASE PROJECT PROPOSAL

Jeremy Tandjung, Nathan Phan, Aaron Handjojo, and Brandon Posey

University of Washington, Bothell

NBA DATABASE

CSS 475 Winter 2019 – Team Avatar

Jeremy Tandjung, Nathan Phan, Aaron Handjojo, and Brandon Posey

Contents

1. Introduction

In this section we will discuss our team's motivation behind creating the database and the NBA.

1.1. MOTIVATION

Team Avatar intends to create a basketball database based on the properties of the National Basketball Association. The database will track team and player information, statistics, awards, etc within the current NBA season, which is the 2018-19 NBA season.

Team Avatar will implement an update meeting cycle, where every week we meet up to give each other an update on what we have done in the past week.

1.2. ABOUT THE NBA

The National Basketball League [1] is men's professional basketball league that is composed of 30 teams (29 in the United States, 1 in Canada). It is widely accepted that the NBA is the most popular professional basketball league in the world. The NBA was founded in June 6, 1946 as the Basketball Association of America (BAA). The league is divided into two conferences based on the team's location; the Western Conference and the Eastern Conference. Each conference is then divided again into three divisions for each conference. The Western conference has the Pacific, Southwest, and Northwest divisions, whereas the Eastern conference has the Atlantic, Central, and Southeast divisions.

2. APPLICATION

The application is primarily used by NBA teams to determine whether a player is playing well or not. The database could also be used by fantasy league users to see if their current fantasy roster is playing well or not. NBA fans could also look up trivial information of their favorite team and players, such as birthdate, age, height, and many more.

For the UI, we believe that since many NBA enthusiast are familiar with the internet, we decided to make a website that is backed by this database.

3. ENTITIES

In this section, we will discuss all entities and attributes, along with its explanation.

3.1. TEAMS

The 30 Teams that are in the NBA as of the current NBA season; 2018-19 NBA season.

- TeamID : The unique identifier for each team
- TeamName : The team’s name
- CoachID : The unique identifier for the team’s head coach
- ArenaID : The unique identifier for the team’s arena
- Conference : The conference where the team is placed in
- Division : The division where the team is placed in
- Championships : The number of NBA championships the team has won in its history
- Wins : The number of wins the team has achieved this season
- Losses : The number of losses the team has achieved this season

3.2. ARENAS

The current arenas used by the 30 NBA teams.

- ArenaID : Unique identifier for each arena
- ArenaName : The arena’s name
- Capacity : The maximum capacity of the arena
- Address : The address of the arena
- City : The city where the arena is located
- State : The state where the arena is located

3.3. HEAD_COACHES

The current and past head coaches of the 30 NBA teams during the 2018-19 NBA season

- CoachID : Unique identifier for each head coach
- Name : The head coach’s name
- Wins : The total wins the coach has achieved throughout his career
- Losses : The total losses the coach has suffered throughout his career
- DOB : The coach’s date of birth

3.4. PLAYER

The current roster of the 30 NBA teams

- PlayerID : The unique identifier for each player
- TeamID : The unique identifier for the team the player is playing for
- Fname : The player’s first name
- Lname : The player’s last name
- DOB : The player’s date of birth
- Height_ft : The player’s height’s feet value
- Height_inch : The player’s height’s inch value
- Weight : The player’s weight (in lbs)
- DraftYear : The player’s draft year

3.5. PLAYED_FOR

The player’s per game statistics

- PlayerId : Unique identifier of PLAYER
- TeamId : Unique identifier of TEAM
- PPG : The player’s points per game value
- APG : The player’s assists per game value
- RPG : The player’s rebounds per game value
- SPG : The player’s steals per game value
- BPG : The player’s blocks per game value
- FG : The field goal percentage of the player
- 3FG : The three-point field goal percentage of the player
- FT : The free throw percentage of the player

3.6. AWARDS

The awards given to the players for their performance in the regular season

- AwardId : Unique identifier of each award
- Name : The name of the award
- Description : The description of the award

3.7. AWARD_WINNER

The active players that have won an award in the past

- AwardId : Unique identifier for each award
- YearWon : The year in which the player won the award

- PlayerId : The unique identifier for the player who won the award

4. Design

In this section, we will discuss the assumption, design decision, and both ER and RM diagrams that our group have chosen to work it

4.1. ASSUMPTIONS

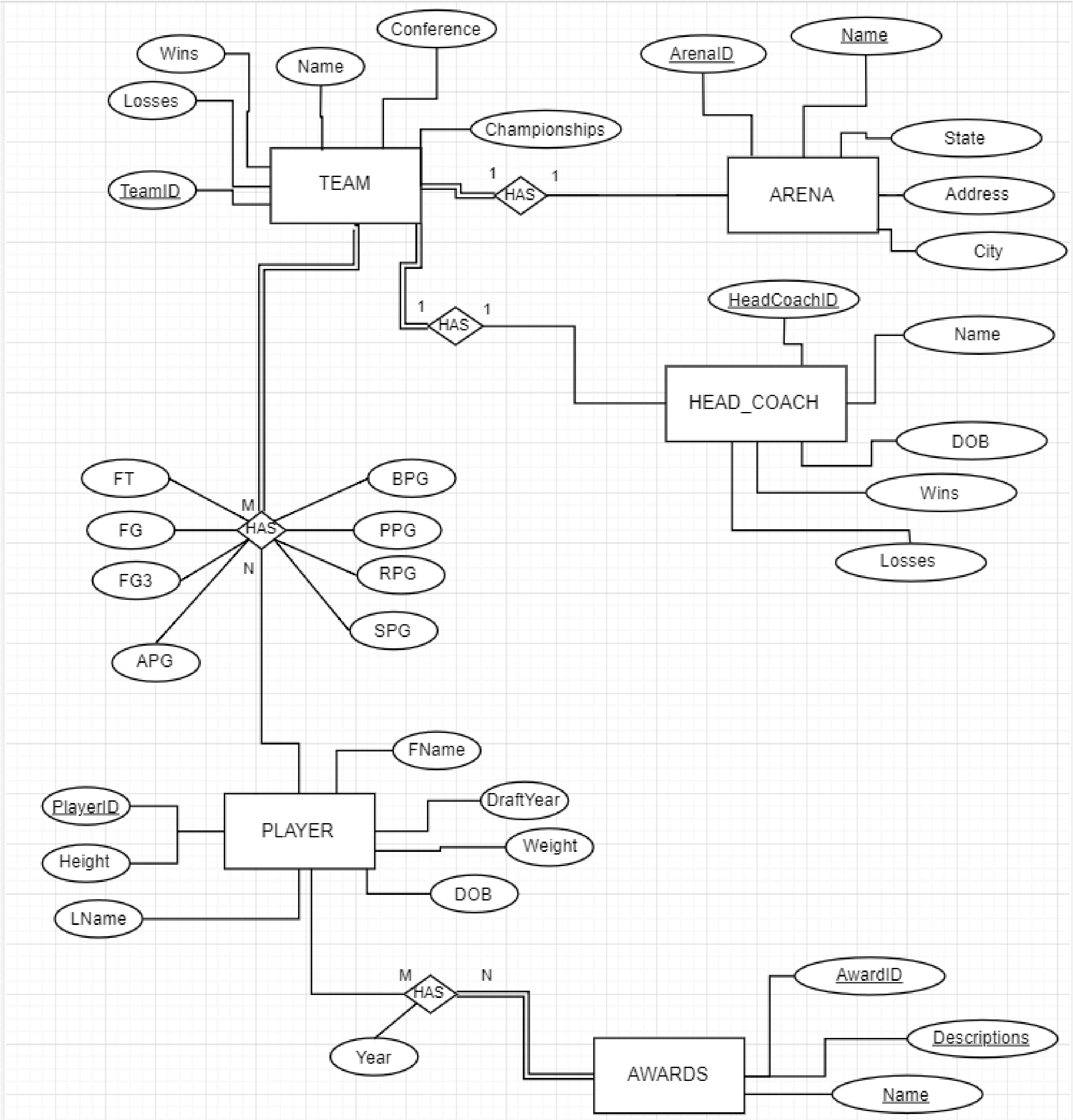
1. This database only contains data from the current NBA season during the making of this project, which is the 2018-19 NBA season.
2. All player statistics and information are up to date until the day before the project presentation which is March 17th, 2019.
3. This database will include all current arenas that are used by the 30 NBA teams.
4. This database will include past and current head coaches that has coached an NBA team. That means, coaches who are currently not coaching an NBA team are included in the database.

4.2. ENTITIY RELATIONSHIP (ER) DIAGRAM

The Entity Relationship Diagram provides an abstract overview of the various relationships between the entities derived from the requirements of the NBA Database. The strong entities include the following entities: TEAM, ARENA, AWARD, HEAD_COACH, and PLAYER. The attributes that correspond with each entity are clearly connected. In addition, attributes that serve as keys are underlined to indicate their status as candidate keys, or partially underlined to indicate their status as partial keys. Each relationship between entities has its cardinality and participation status clearly labeled. Refer to Figure 1 for further details regarding the diagram.

Team Avatar chose to create “HAS” relationships between TEAM and the entities of ARENA, HEAD_COACH, and PLAYER. The reasoning for this decision is that TEAMS in the NBA have direct relationships with those entities wherein the latter entities are identified as belonging to the TEAM. Since an NBA team needs to have a relationship with those entities, it was decided to impose total participation on the TEAM side of those relationships. In addition, the cardinality ratio of the relationships are 1:1 in the case of TEAM and ARENA, and TEAM, and HEAD_COACHES

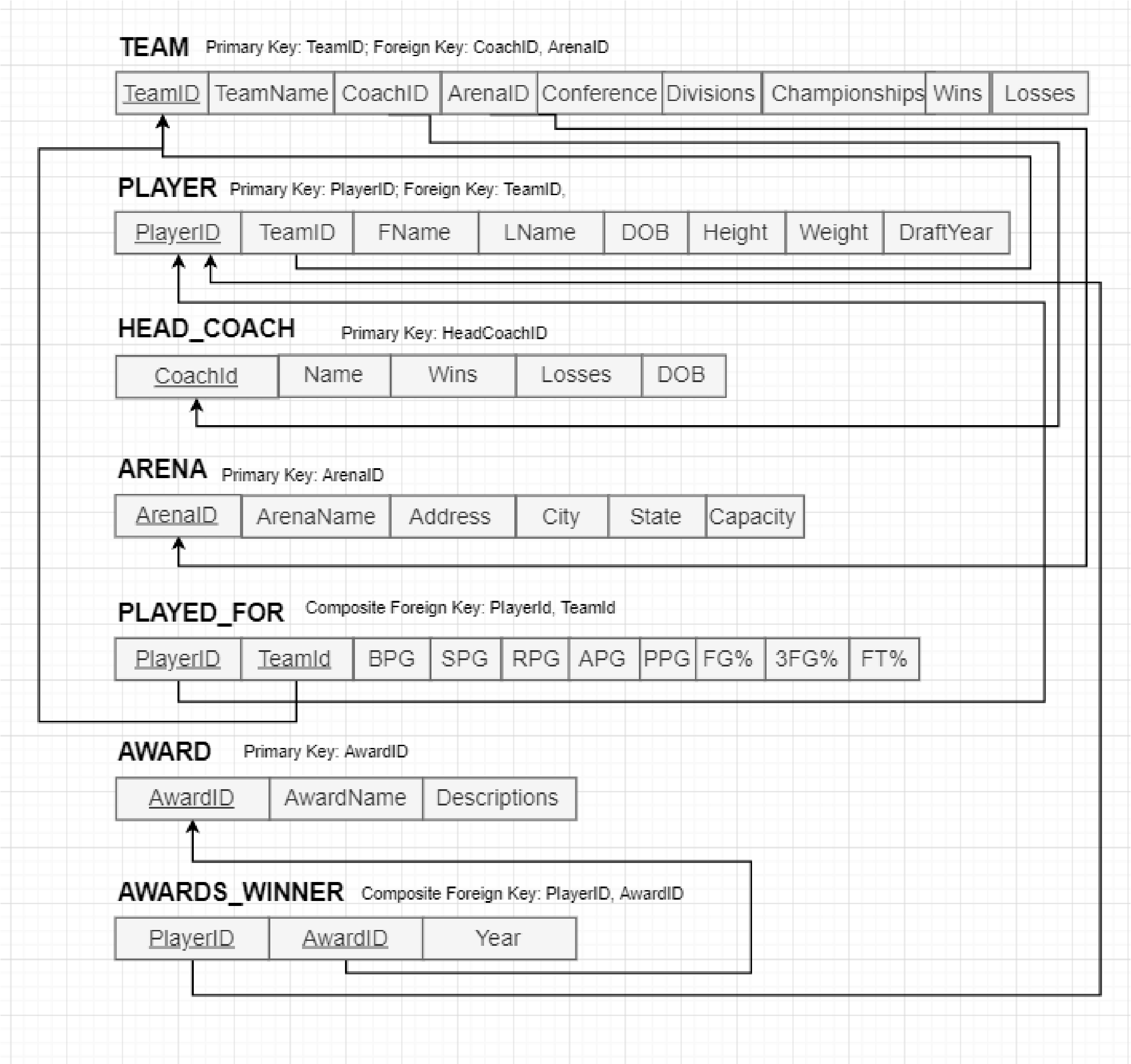
Figure 1: NBA ER Diagram



4.3. RELATIONAL MODEL

The Relational Model provides an abstract overview of the different relations derived from the requirements. Each relation will include its entity's attributes including keys such as primary keys, foreign keys, and partial keys. The keys that are referenced from other relations are modeled using arrows to demonstrate the origin of the foreign key. In this case, the relation AWARDS_WON represents the relationship between PLAYER and AWARDS. The relation itself will have no distinctive key as the relationship between PLAYER and AWARDS with the foreign keys of PlayerID, AwardID, and the relation's attribute of Year providing meaning and uniqueness for the AWARDS_WON relation. Refer to Figure 2 for further details regarding the Relational Model schema.

Figure 2: NBA RM Diagram


TEAM Primary Key: TeamID; Foreign Key: CoachID, ArenaID
TeamID
TeamName
CoachID
ArenaID
Conference
Divisions
Championships
Wins
Losses
PLAYER Primary Key: PlayerID; Foreign Key: TeamID,
PlayerID
TeamID
FName
LName
DOB
Height
Weight
DraftYear
HEAD_COACH Primary Key: HeadCoachID
CoachId
Name
Wins
Losses
DOB
ARENA Primary Key: ArenaID
ArenaID
ArenaName
Address
City
State
Capacity
PLAYED_FOR Composite Foreign Key: PlayerId, TeamId
PlayerID
TeamId
BPG
SPG
RPG
APG
PPG
FG%
3FG%
FT%
AWARD Primary Key: AwardID
AwardID
AwardName
Descriptions
AWARDS_WINNER Composite Foreign Key: PlayerID, AwardID
PlayerID
AwardID
Year

4.4. CONSTRAINTS

- **TEAM**
  - TeamId:
    - PRIMARY KEY
    - Unsigned integer
    - Not NULL
    - Domain: All positive integers

    - TeamName:
        - Not NULL
        - Domain: VarChar(40)
    - Conference:
        - Not NULL
        - Domain: Enum (‘West’, ‘East’)
    - Division
        - NOT NULL
        - Domain: Enum (‘Pacific’, ‘Northwest’, ‘Southwest’, ‘Atlantic’, ‘Central’, ‘Southeast’)
    - Championship:
        - NULL
        - Domain: All non-negative integers
    - Wins:
        - Not NULL
        - Domain: All non-negative integers
    - Loses:
        - Not NULL
        - Domain: All non-negative integers
    - CoachID:
        - FOREIGN KEY from HEAD_COACHES
        - Not NULL
        - Domain: All non-negative integers
    - ArenaID:
        - FOREIGN KEY from ARENA
        - Not NULL
        - Domain: All non-negative integers
- **PLAYER**
    - PlayerID:
        - PRIMARY KEY
        - Not NULL

        - Domain: All positive integers
    - Fname:
        - Not NULL
        - Domain: All non-numerical VarChar(40)
    - Lname:
        - Not NULL
        - Domain: All non-numerical VarChar(40)
    - DOB:
        - Not NULL
        - Domain: integers in the format mm/dd/yy where: 1 <= mm <= 12, 1 <= dd<= 31 (for months with 31 days) or 1 <= dd <= 30 (for months with 30 days) or 1 <= dd <= 28 (for February in a non-leap year) or 1 <= dd <=29 (for February in a leap year), and yy being the last two digits of years between 1900 and 2099.
    - Height_ft:
        - Not NULL
        - Domain: All non-negative integers
    - Height_inch
        - Not NULL
        - Domain: All unsigned integer
    - Weight:
        - Not NULL
        - Domain: All non-negative integers
    - DraftYear:
        - Not NULL
        - Domain: 4-digit positive integer between 1900 and 2099
    - TeamID:
        - FOREIGN KEY from TEAM
        - Not NULL
        - Domain: All non-negative integers
- **HEAD_COACH**
    - CoachID:
        - PRIMARY KEY

    - Not NULL
    - Domain: All positive integers
  - Name:
    - Not NULL
    - Domain: All non-numerical VarChar(40)
  - Wins:
    - Not NULL
    - Domain: All non-negative integers
  - Loses:
    - Not NULL
    - Domain: All non-negative integers
  - DOB:
    - Not NULL
    - Domain: integers in the format mm/dd/yy where: 1 <= mm <= 12, 1 <= dd<= 31 (for months with 31 days) or 1 <= dd <= 30 (for months with 30 days) or 1 <= dd <= 28 (for February in a non-leap year) or 1 <= dd <=29 (for February in a leap year), and yy being the last two digits of years between 1900 and 2099.

- **ARENA**
  - ArenaID:
    - PRIMARY KEY
    - Not NULL
    - Domain: All positive integers
  - Name:
    - Not NULL
    - Domain: All non-numerical VarChar(40)
  - Address:
    - Not NULL
    - Domain: VarChar(40)
  - City:
    - Not NULL

        - Domain: VarChar(40) of a name of a valid city in the US within the State.
    - State:
        - Not NULL
        - Domain: Char(2) of a name of a US State in abbreviation
    - Capacity:
        - Not NULL
        - Domain: All non-negative integers
- **PLAYED_FOR**
    - PlayerId:
        - Foreign Key from PLAYER
        - Not NULL
        - Domain: All positive integers
    - TeamId:
        - Foreign Key from TEAM
        - Not NULL
        - Domain: All positive integers
    - BPG:
        - Not NULL
        - Domain: All non-negative double
    - SPG:
        - Not NULL
        - Domain: All non-negative double
    - RPG:
        - Not NULL
        - Domain: All non-negative double
    - APG:
        - Not NULL
        - Domain: All non-negative double
    - PPG:
        - Not NULL

        - Domain: All non-negative double
    - FG:
        - Not NULL
        - Domain: All non-negative double
    - 3FG:
        - Not NULL
        - Domain: All non-negative double
    - FT:
        - Not NULL
        - Domain: All non-negative double
- **AWARD**
    - AwardID:
        - PRIMARY KEY
        - Not NULL
        - Domain: All positive integers
    - Name:
        - Not NULL
        - Domain: VarChar(40)
    - Description:
        - Not NULL
        - Domain: VarChar(400)
- **AWARD_WINNER**
    - AwardID:
        - PARENT KEY from AWARDS
        - Not NULL
        - Domain: All positive integers
    - PlayerID:
        - PARENT KEY from PLAYER
        - Not NULL
        - Domain: All positive integers
    - Year:

- Not NULL
- Domain: 4-digit positive integer between 1900 and 2099

5. QUERIES

In this section we will discuss the scope of all possible queries and the queries that are used in making the website.

The NBA database will allow any interested party, namely NBA fantasy league players, to query information about players and teams to see trends and make predictions about team, and individual performance for the 2018-2019 season.

For example, an NBA fantasy player may query the database to search for every player who currently plays for the Boston Celtics. The Fantasy player may then search for players who are not listed as out or suspended for the season.

The Fantasy player can then search the database with the existing filters to determine which of the players have a higher fantasy point average. The fantasy player can also narrow down the results by querying the database to search for players who have a higher than average projected points for the year.

Besides fantasy players, casual fans can also enjoy following their favorite players and teams for the current season and see how they are doing so far. From fun facts about a team or player to fun statistics of a certain group of players are some queries that this database can output to casual fans.

5.1. SAMPLE QUERIES

The table below showcases sample queries that can be pulled from the database and their purpose. Some queries below will show some php keywords, as those queries are used in our website.

SQL statement	Purpose

SELECT CONCAT (Fname, ' ', Lname) AS PlayerName, APG
FROM PLAYED_FOR
JOIN PLAYER USING (PlayerId)
ORDER BY APG DESC
LIMIT 10;
	Displays the names of the top ten players who lead the NBA in the average assists per game category of statistics. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know who

	performed best in this statistical category in the past decade.
SELECT TeamName, ArenaName, Arena.Address AS Arena_Address, City, Arena.State AS Arena_State, Capacity, BuiltIn FROM TEAM JOIN ARENA USING (ArenaId) WHERE Team.ArenaId = Arena.ArenaId;	Displays the name, address, city, state, capacity, year built, of each team in the NBA. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know the arena information for their favorite team.

SELECT CONCAT(Fname, ' ', Lname) AS
PlayerName, BPG
FROM PLAYED_FOR
JOIN PLAYER USING (PlayerId)
ORDER BY BPG DESC
LIMIT 10;
	Displays the names of the top ten players who lead the NBA in the average blocks per game category of statistics. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know who performed best in this statistical category in the past decade.

SELECT TEAM.TeamName,
HEAD_COACH.Name
FROM HEAD_COACH
JOIN TEAM USING (CoachId);
	Displays the names of all the coaches in the NBA corresponding with the NBA team that they are coaching this season. This query was included with the NBA database system because Team Avatar believes that an average NBA fan would want to know who is currently coaching their favorite team.

SELECT AWARD.Description,
AWARD_WINNER.YearWon,
PLAYER.Fname, PLAYER.Lname
FROM AWARD
JOIN AWARD_WINNER USING (AwardId)
JOIN PLAYER USING (PlayerId)
WHERE Name = 'Defensive Player of the
Year';
	Displays the description of the ‘Defensive Player of the Year’ award, the year that the award was won, and the first and last name of the player who won it that year. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know which players won this award recently.

SELECT CONCAT(Height_ft, '\'',
Height_inch, '\"') AS Height, AVG(PPG) AS
AveragePPG
FROM PLAYER
JOIN PLAYED_FOR USING (PlayerId)
GROUP BY Height
ORDER BY Height_ft DESC, Height_inch
DESC;
	Displays the height in feet and inches and their correlation with the average points per game category. This query was used as an instance where the NBA Database could be used to discover correlations and patterns between different attributes in the NBA. In this case it was used to discover how height correlated with average points per game.

SELECT TeamName, MAX(PPG) AS
HighestPPG, Wins
FROM TEAM
JOIN PLAYED_FOR USING (TeamId)
JOIN PLAYER USING (PlayerId)
GROUP BY (TeamName)
	Displays the name of teams, the average points per game value of their highest scoring player, and the team’s number of wins. This query was used as an instance where the NBA Database could be used to discover correlations and patterns between different

ORDER BY Wins DESC;	attributes in the NBA. In this case it was used to discover the relationship between a team possessing high scoring players and the team number of wins.

SELECT TeamName, Championships
FROM TEAM
ORDER BY Championships DESC;
	Displays the name of teams and their corresponding number of championships. This query was used to show interesting information that a casual viewer of the NBA would want to know from an NBA Database.

SELECT CONCAT(Fname, ' ', Lname) AS PlayerName, FG
FROM PLAYED_FOR
JOIN PLAYER USING (PlayerId)
WHERE FG < 0.66
ORDER BY FG DESC
LIMIT 10;
	Displays the names of the top ten players who lead the NBA in the field goal percentage per game category of statistics. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know who performed best in this statistical category in the past decade.

SELECT CONCAT(Fname, ' ', Lname) AS PlayerName, FG3
FROM PLAYED_FOR
JOIN PLAYER USING (PlayerId)
WHERE FG3 < 0.47
ORDER BY FG3 DESC
LIMIT 10;
	Displays the names of the top ten players who lead the NBA in the field goal 3 percentage per game category of statistics. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know who performed best in this statistical category in the past decade.

SELECT CONCAT(Fname, ' ', Lname) AS PlayerName, FT
FROM PLAYED_FOR
JOIN PLAYER USING (PlayerId)
WHERE FT < 0.904
ORDER BY FT DESC
LIMIT 10;
	Displays the names of the top ten players who lead the NBA in the free throw percentage per game category of statistics. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know who performed best in this statistical category in the past decade.

SELECT TeamName AS Team, Wins, Losses
FROM TEAM
WHERE TEAM.Conference = 'East'
ORDER BY TEAM.Wins DESC;
	Shows the team names, wins and losses of the teams that play in the Eastern Conference. This was included in the NBA database system because NBA fans would be able to use the NBA database system to find their team’s performance relative to other teams in their team’s conference. This information is essential as part of a functioning NBA database because a team’s current win-loss record influences their seeding in the playoffs or their odds in the draft lottery.

SELECT TeamName AS Team, Wins, Losses
FROM TEAM
	Shows the team names, wins and losses of the teams that play in the Western Conference.

WHERE TEAM.Conference = 'West'
ORDER BY TEAM.Wins DESC;
	This was included in the NBA database system because NBA fans would be able to use the NBA database system to find their team’s performance relative to other teams in their team’s conference. This information is essential as part of a functioning NBA database because a team’s current win-loss record influences their seeding in the playoffs or their odds in the draft lottery.

SELECT TeamName AS Team, Wins, Losses
FROM TEAM
WHERE TEAM.Division = 'Atlantic'
ORDER BY TEAM.Wins DESC;
	Shows the team names, wins and losses of the teams that play in the Atlantic Division. This was included in the NBA database system because NBA fans would be able to use the NBA database system to find their team’s performance relative to other teams in their team’s division. This information is essential as part of a functioning NBA database because a team’s current win-loss record influences their seeding in the playoffs or their odds in the draft lottery.

SELECT TeamName AS Team, Wins, Losses
FROM TEAM
WHERE TEAM.Division = 'Central'
ORDER BY TEAM.Wins DESC;
	Shows the team names, wins and losses of the teams that play in the Central Division. This was included in the NBA database system because NBA fans would be able to use the NBA database system to find their team’s performance relative to other teams in their team’s division. This information is essential as part of a functioning NBA database because a team’s current win-loss record influences their seeding in the playoffs or their odds in the draft lottery.

SELECT TeamName AS Team, Wins, Losses
FROM TEAM
WHERE TEAM.Division = 'Southeast'
ORDER BY TEAM.Wins DESC;
	Shows the team names, wins and losses of the teams that play in the Southeast Division. This was included in the NBA database system because NBA fans would be able to use the NBA database system to find their team’s performance relative to other teams in their team’s division. This information is essential as part of a functioning NBA database because a team’s current win-loss record influences their seeding in the playoffs or their odds in the draft lottery.

SELECT TeamName AS Team, Wins, Losses
FROM TEAM
WHERE TEAM.Division = 'Northwest'
ORDER BY TEAM.Wins DESC;
	Shows the team names, wins and losses of the teams that play in the Northwest Division. This was included in the NBA database system because NBA fans would be able to

	use the NBA database system to find their team’s performance relative to other teams in their team’s division. This information is essential as part of a functioning NBA database because a team’s current win-loss record influences their seeding in the playoffs or their odds in the draft lottery.

SELECT TeamName AS Team, Wins, Losses
FROM TEAM
WHERE TEAM.Division = 'Pacific'
ORDER BY TEAM.Wins DESC;
	Shows the team names, wins and losses of the teams that play in the Pacific Division. This was included in the NBA database system because NBA fans would be able to use the NBA database system to find their team’s performance relative to other teams in their team’s division. This information is essential as part of a functioning NBA database because a team’s current win-loss record influences their seeding in the playoffs or their odds in the draft lottery.

SELECT TeamName AS Team, Wins, Losses
FROM TEAM
WHERE TEAM.Division = 'Southwest'
ORDER BY TEAM.Wins DESC;
	Shows the team names, wins and losses of the teams that play in the Southwest Division. This was included in the NBA database system because NBA fans would be able to use the NBA database system to find their team’s performance relative to other teams in their team’s division. This information is essential as part of a functioning NBA database because a team’s current win-loss record influences their seeding in the playoffs or their odds in the draft lottery.

SELECT AWARD.Description, AWARD_WINNER.YearWon, PLAYER.Fname, PLAYER.Lname FROM AWARD JOIN AWARD_WINNER USING (AwardId) JOIN PLAYER USING (PlayerId) WHERE Name = 'Most Improved Player';	Displays the description of the 'Most Improved Player' award, the year that the award was won, and the first and last name of the player who won it that year. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know which players won this award recently.
SELECT AWARD.Description, AWARD_WINNER.YearWon, PLAYER.Fname, PLAYER.Lname FROM AWARD JOIN AWARD_WINNER USING (AwardId) JOIN PLAYER USING (PlayerId) WHERE Name = 'Most Valuable Player';	Displays the description of the 'Most Valuable Player' award, the year that the award was won, and the first and last name of the player who won it that year. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know which players won this award recently.

SELECT CONCAT(Fname, ' ', Lname) AS PlayerName, PPG
FROM PLAYED_FOR
JOIN PLAYER USING (PlayerId)
ORDER BY PPG DESC
LIMIT 20;
	Displays the names of the top twenty players who lead the NBA in the average points per game category of statistics. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know who performed best in this statistical category in the past two decades.

SELECT AWARD.Description, AWARD_WINNER.YearWon, PLAYER.Fname, PLAYER.Lname FROM AWARD JOIN AWARD_WINNER USING (AwardId) JOIN PLAYER USING (PlayerId) WHERE Name = 'Rookie of the Year';	Displays the description of the 'Rookie of the Year' award, the year that the award was won, and the first and last name of the player who won it that year. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know which players won this award recently.

SELECT CONCAT(Fname, ' ', Lname) AS PlayerName, RPG
FROM PLAYED_FOR
JOIN PLAYER USING (PlayerId)
ORDER BY RPG DESC
LIMIT 20;
	Displays the names of the top twenty players who lead the NBA in the average rebounds per game category of statistics. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know who performed best in this statistical category in the past two decades.

SELECT Fname, Lname, PPG, APG, RPG, BPG, SPG, FG, FG3, FT
FROM PLAYED_FOR
JOIN PLAYER USING (PlayerId)
WHERE Fname LIKE '%" . $fname_kw . "%'
AND Lname LIKE '%" . $lname_kw . "%'";
	Displays the first and last name of a player with their corresponding statistical performance in the categories of “Points Per Game,” “Assists Per Game,” “Rebounds Per Game,” “Blocks Per Game,” “Steals Per Game,” “Field Goal Percentage Per Game,” “Field Goal 3 Percentage Per Game,” and “Free Throw Percentage Per Game.” The '%" . $fname_kw . "%' and the '%" . $lname_kw . "%' segments are used for PHP to recognize the requested values of Player first name and Player last name and parse it into a manner that MySQL can use to conduct the query. This query was included as part of the ‘PLAYERS’ section of the NBA Database system GUI in order to show the statistical performance of a player that was searched for by an end user.

SELECT AWARD.Name,
AWARD_WINNER.YearWon
FROM AWARD_WINNER
JOIN PLAYER USING (PlayerId)
	Displays the all of the awards that a player has won including the award name and the year the player won that award. The '%" . $fname_kw . "%' and the '%" . $lname_kw .

JOIN AWARD USING (AwardId)
WHERE Fname LIKE '%" . $fname_kw . "%'
AND Lname LIKE '%" . $lname_kw . "%'";
	"%' segments are used for PHP to recognize the requested values of Player first name and Player last name and parse it into a manner that MySQL can use to conduct the query. This query was included as part of the 'PLAYERS' section of the NBA Database system GUI in order to show the achievements of a player that was searched for by a typical end user.

SELECT TeamName, TEAM.Wins AS teamWins, TEAM.Losses AS teamLosses, ArenaName, HEAD_COACH.Name AS coachName, Championships FROM TEAM JOIN ARENA USING (ArenaId) JOIN HEAD_COACH USING (CoachId) WHERE TeamName LIKE '%" . $team_kw . "%'";	Displays the wins, losses, home stadium, and current head coach of a team that is requested on the "TEAMS" section of the website. The '%" . $team_kw . "%'" is used for PHP to recognize the requested value and parse it into a manner that MySQL can use to conduct the query of Team record, home stadium, and current head coach. This query was included as part of the 'TEAMS' section of the NBA Database system GUI in order to show the overview of a team that was searched for by a typical end user.

SELECT CONCAT(Fname, ' ', Lname) AS PlayerName, DOB, CONCAT(Height_ft, '\'', Height_inch, '\"') AS Height, Player.Weight, DraftYear
FROM PLAYER
JOIN TEAM USING (TeamId)
WHERE TeamName LIKE '%" . $team_kw . "%'";
	Displays the first and last name of the player on the roster of the team requested, their date of birth, their accompanying height in feet and inches, their weight, and the year the player was drafted. If the player was not drafted then the system will display "Undrafted" through a PHP echo. The '%" . $team_kw . "%'" is used for PHP to recognize the requested value and parse it into a manner that MySQL can use to conduct the query. This query was included as part of the 'TEAMS' section of the NBA Database system GUI in order to show the overview of players on the roster of the team that was searched for by a typical end user.

SELECT TeamId, Team.Name,
FROM TEAM";
	This query shows all the teams in the NBA. It was included as a sample SQL query in one of the system's PHP files as an exercise of using MySQL with PHP.

SELECT AWARD.Description, AWARD_WINNER.YearWon, PLAYER.Fname, PLAYER.Lname FROM AWARD JOIN AWARD_WINNER USING (AwardId) JOIN PLAYER USING (PlayerId)	Displays the description of the 'Sixth Man of the Year' award, the year that the award was won, and the first and last name of the player who won it that year. This query was included in the NBA database system because Team Avatar believes that an

WHERE Name = 'Sixth Man of the Year';	average NBA fan would want to know which players won this award recently.

SELECT CONCAT(Fname, ' ', Lname) AS PlayerName, SPG
FROM PLAYED_FOR
JOIN PLAYER USING (PlayerId)
ORDER BY SPG DESC
LIMIT 10;
	Displays the names of the top ten players who lead the NBA in the average steals per game category of statistics. This query was included in the NBA database system because Team Avatar believes that an average NBA fan would want to know who performed best in this statistical category in the past decade.

6. DATA GENERATION

Team Avatar will use public information provided by http://basketball-reference.com [2] and the Canadian Broadcasting Corporation [3] to populate the database with NBA teams and players. The statistics that are available for each team and player will populate the team and player records. Since the nature of the data is dynamic, the data the we are using is up to date until the day before the presentation, which is March 17th, 2019.

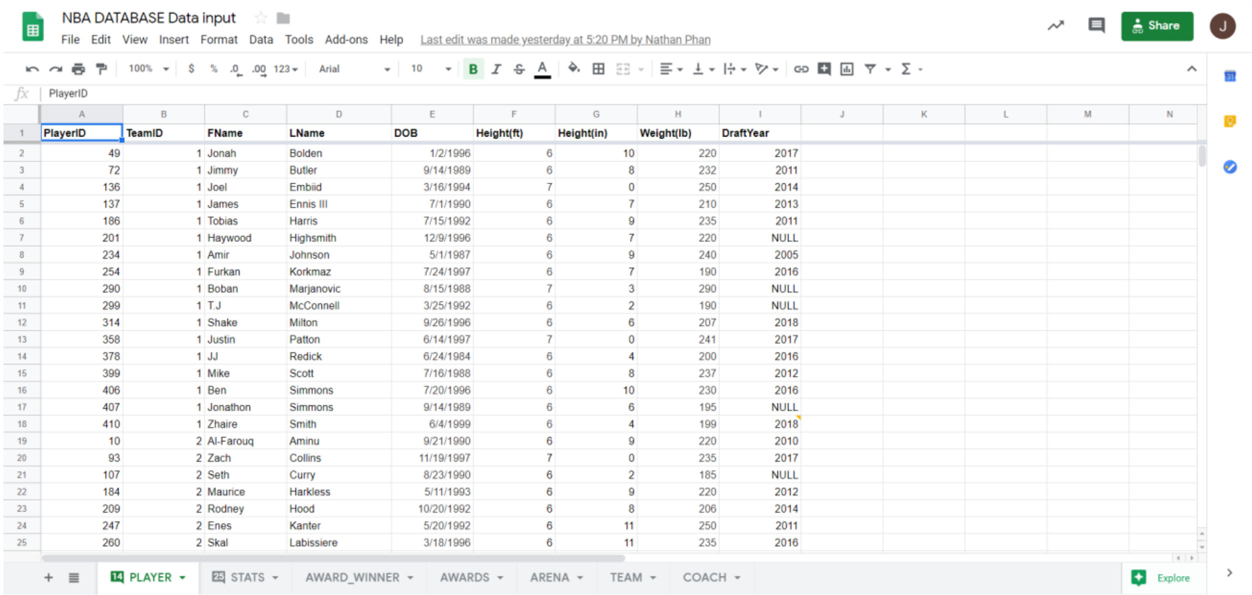
7. METHODOLOGY

For this project, basically the process can be separated into four sections, getting data, creating the database, inputting the data, and website building.

7.1. GETTING DATA

As mentioned before, we have chosen http://basketball-reference.com as our source for the latest data for this database. Since the data is dynamic, we decided to get the data and not input them immediately onto the database. Instead, we put them in a spreadsheet at Google Spreadsheet as it is a collaboration-friendly platform. Figure 3 is a screenshot of the sheet that we use for our data input.

Figure 3: Google Spreadsheet for NBA Database input


NBA DATABASE Data input
File Edit View Insert Format Data Tools Add-ons Help
Last edit was made yesterday at 5:20 PM by Nathan Phan
Share
100%
Arial
10
PlayerID
PLAYER
STATS
AWARD_WINNER
AWARDS
ARENA
TEAM
COACH
Explore

PlayerID	TeamID	FName	LName	DOB	Height(ft)	Height(in)	Weight(lb)	DraftYear
49	1	Jonah	Bolden	1/2/1996	6	10	220	2017
72	1	Jimmy	Butler	9/14/1989	6	8	232	2011
136	1	Joel	Embiid	3/16/1994	7	0	250	2014
137	1	James	Ennis III	7/1/1990	6	7	210	2013
186	1	Tobias	Harris	7/15/1992	6	9	235	2011
201	1	Haywood	Highsmith	12/9/1996	6	7	220	NULL
234	1	Amir	Johnson	5/1/1987	6	9	240	2005
254	1	Furkan	Korkmaz	7/24/1997	6	7	190	2016
290	1	Boban	Marjanovic	8/15/1988	7	3	290	NULL
299	1	T.J	McConnell	3/25/1992	6	2	190	NULL
314	1	Shake	Milton	9/26/1996	6	6	207	2018
358	1	Justin	Patton	6/14/1997	7	0	241	2017
378	1	JJ	Redick	6/24/1984	6	4	200	2016
399	1	Mike	Scott	7/16/1988	6	8	237	2012
406	1	Ben	Simmons	7/20/1996	6	10	230	2016
407	1	Jonathon	Simmons	9/14/1989	6	6	195	NULL
410	1	Zhaire	Smith	6/4/1999	6	4	199	2018
10	2	Al-Farouq	Aminu	9/21/1990	6	9	220	2010
93	2	Zach	Collins	11/19/1997	7	0	235	2017
107	2	Seth	Curry	8/23/1990	6	2	185	NULL
184	2	Maurice	Harkless	5/11/1993	6	9	220	2012
209	2	Rodney	Hood	10/20/1992	6	8	206	2014
247	2	Enes	Kanter	5/20/1992	6	11	250	2011
260	2	Skal	Labissiere	3/18/1996	6	11	235	2016

The spreadsheet is composed of multiple sheets that represents each relation in our database.

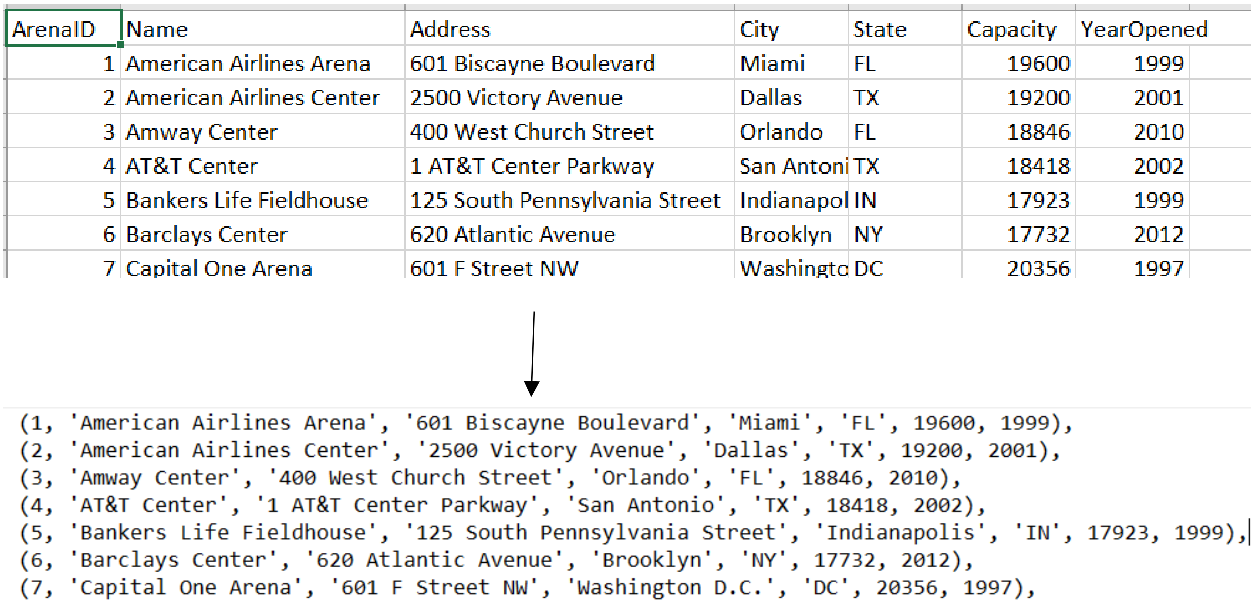
7.2. CREATING THE DATABASE

Team Avatar used MySQL Workbench to create the database. We applied many SQL syntax that we learned in CSS 475, including creating a table, inserting values into a table, and basic queries.

7.3. INPUTTING THE DATA

Since the amount of data inputs is enormous, it would be ineffective if we typed all of them by hand as SQL input values. Fortunately, one of our members created a python program that converted a csv file to a text file that follows the syntax of a SQL value input. Figure 4 showcases the conversion from csv to SQL syntax text file.

Figure 4: CSV to SQL input



ArenaID	Name	Address	City	State	Capacity	YearOpened
1	American Airlines Arena	601 Biscayne Boulevard	Miami	FL	19600	1999
2	American Airlines Center	2500 Victory Avenue	Dallas	TX	19200	2001
3	Amway Center	400 West Church Street	Orlando	FL	18846	2010
4	AT&T Center	1 AT&T Center Parkway	San Antoni	TX	18418	2002
5	Bankers Life Fieldhouse	125 South Pennsylvania Street	Indianapol	IN	17923	1999
6	Barclays Center	620 Atlantic Avenue	Brooklyn	NY	17732	2012
7	Capital One Arena	601 F Street NW	Washingto	DC	20356	1997

```
(1, 'American Airlines Arena', '601 Biscayne Boulevard', 'Miami', 'FL', 19600, 1999),
(2, 'American Airlines Center', '2500 Victory Avenue', 'Dallas', 'TX', 19200, 2001),
(3, 'Amway Center', '400 West Church Street', 'Orlando', 'FL', 18846, 2010),
(4, 'AT&T Center', '1 AT&T Center Parkway', 'San Antonio', 'TX', 18418, 2002),
(5, 'Bankers Life Fieldhouse', '125 South Pennsylvania Street', 'Indianapolis', 'IN', 17923, 1999),
(6, 'Barclays Center', '620 Atlantic Avenue', 'Brooklyn', 'NY', 17732, 2012),
(7, 'Capital One Arena', '601 F Street NW', 'Washington D.C.', 'DC', 20356, 1997),
```

Even though there are still some bugs for the python program, it reduces the amount of time we have to use in order to input the data. In total, there were 1105 tuples of data input.

7.4. BUILDING THE WEBSITE

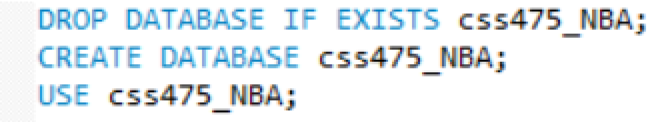
The final step to this project is building the website as a mean of user interaction with the database. For the website, we used USBWebserver, which is a combination of popular webserver software; PHP, Apache, MySQL, and phpMyAdmin. For publishing the website, we used ngrok.io as our web host.

8. SQL SCRIPT

As mentioned before, team Avatar uses MySQL as our DBMS to support our NBA database. We have attached a .sql file for all the script below in another separate file.

8.1. DATABASE CREATION

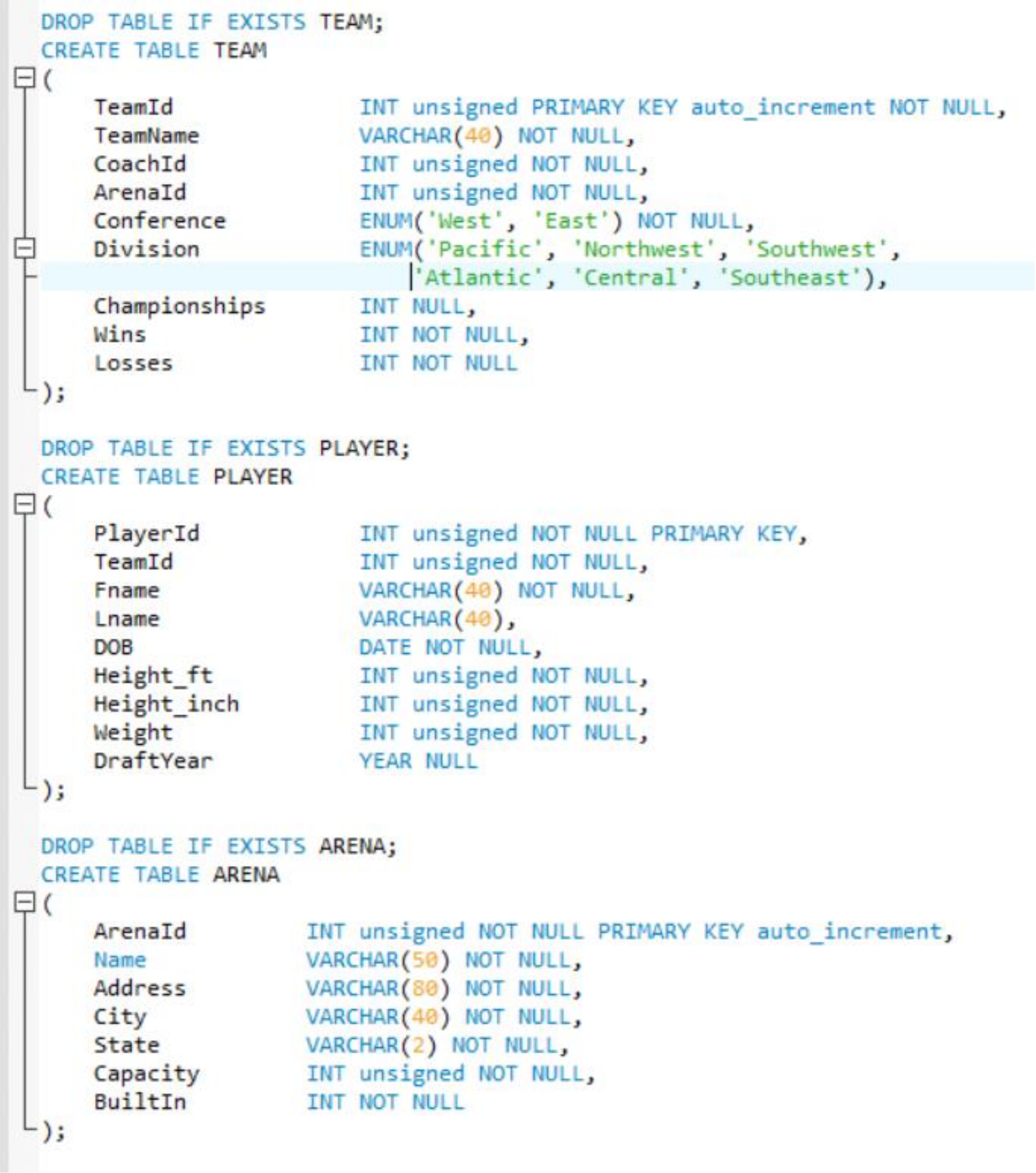
Since the size of our database is not that large, we have decided that we will regenerate our database whenever we run the script and want to apply some changes to the database.



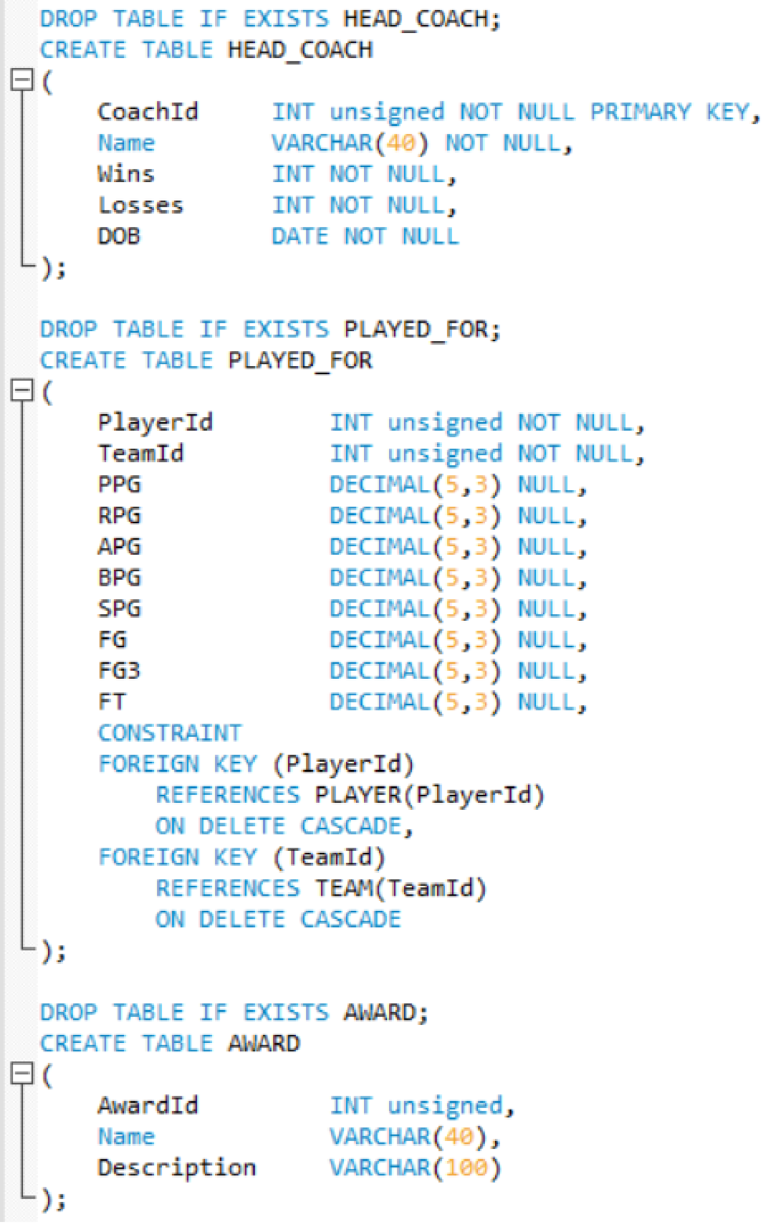
```
DROP DATABASE IF EXISTS css475_NBA;
CREATE DATABASE css475_NBA;
USE css475_NBA;
```

8.2. TABLE CREATION

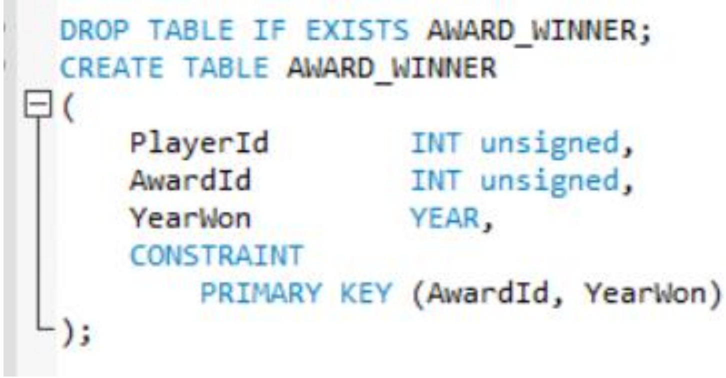
Like the database itself, we have decided to regenerate the tables whenever we run the script and want to apply some changes to the database.



```
DROP TABLE IF EXISTS TEAM;
CREATE TABLE TEAM
(
    TeamId                  INT unsigned PRIMARY KEY auto_increment NOT NULL,
    TeamName                VARCHAR(40) NOT NULL,
    CoachId                 INT unsigned NOT NULL,
    ArenaId                 INT unsigned NOT NULL,
    Conference              ENUM('West', 'East') NOT NULL,
    Division                ENUM('Pacific', 'Northwest', 'Southwest',
                               'Atlantic', 'Central', 'Southeast'),
    Championships           INT NULL,
    Wins                    INT NOT NULL,
    Losses                  INT NOT NULL
);

DROP TABLE IF EXISTS PLAYER;
CREATE TABLE PLAYER
(
    PlayerId                INT unsigned NOT NULL PRIMARY KEY,
    TeamId                  INT unsigned NOT NULL,
    Fname                   VARCHAR(40) NOT NULL,
    Lname                   VARCHAR(40),
    DOB                     DATE NOT NULL,
    Height_ft               INT unsigned NOT NULL,
    Height_inch             INT unsigned NOT NULL,
    Weight                  INT unsigned NOT NULL,
    DraftYear               YEAR NULL
);

DROP TABLE IF EXISTS ARENA;
CREATE TABLE ARENA
(
    ArenaId             INT unsigned NOT NULL PRIMARY KEY auto_increment,
    Name                VARCHAR(50) NOT NULL,
    Address             VARCHAR(80) NOT NULL,
    City                VARCHAR(40) NOT NULL,
    State               VARCHAR(2) NOT NULL,
    Capacity            INT unsigned NOT NULL,
    BuiltIn             INT NOT NULL
);
```



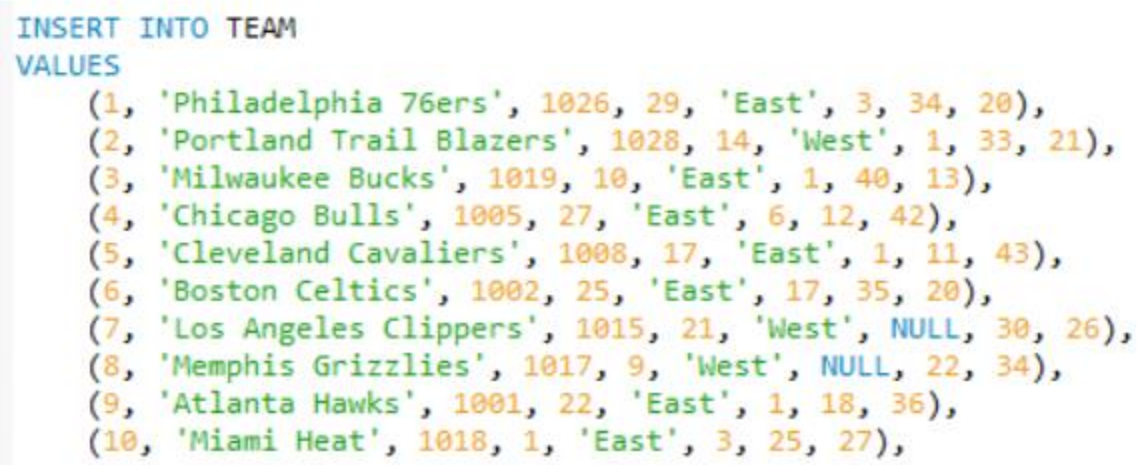
```
DROP TABLE IF EXISTS HEAD_COACH;
CREATE TABLE HEAD_COACH
(
    CoachId      INT unsigned NOT NULL PRIMARY KEY,
    Name         VARCHAR(40) NOT NULL,
    Wins         INT NOT NULL,
    Losses       INT NOT NULL,
    DOB          DATE NOT NULL
);

DROP TABLE IF EXISTS PLAYED_FOR;
CREATE TABLE PLAYED_FOR
(
    PlayerId        INT unsigned NOT NULL,
    TeamId          INT unsigned NOT NULL,
    PPG             DECIMAL(5,3) NULL,
    RPG             DECIMAL(5,3) NULL,
    APG             DECIMAL(5,3) NULL,
    BPG             DECIMAL(5,3) NULL,
    SPG             DECIMAL(5,3) NULL,
    FG              DECIMAL(5,3) NULL,
    FG3             DECIMAL(5,3) NULL,
    FT              DECIMAL(5,3) NULL,
    CONSTRAINT
    FOREIGN KEY (PlayerId)
        REFERENCES PLAYER(PlayerId)
        ON DELETE CASCADE,
    FOREIGN KEY (TeamId)
        REFERENCES TEAM(TeamId)
        ON DELETE CASCADE
);

DROP TABLE IF EXISTS AWARD;
CREATE TABLE AWARD
(
    AwardId         INT unsigned,
    Name            VARCHAR(40),
    Description     VARCHAR(100)
);
```



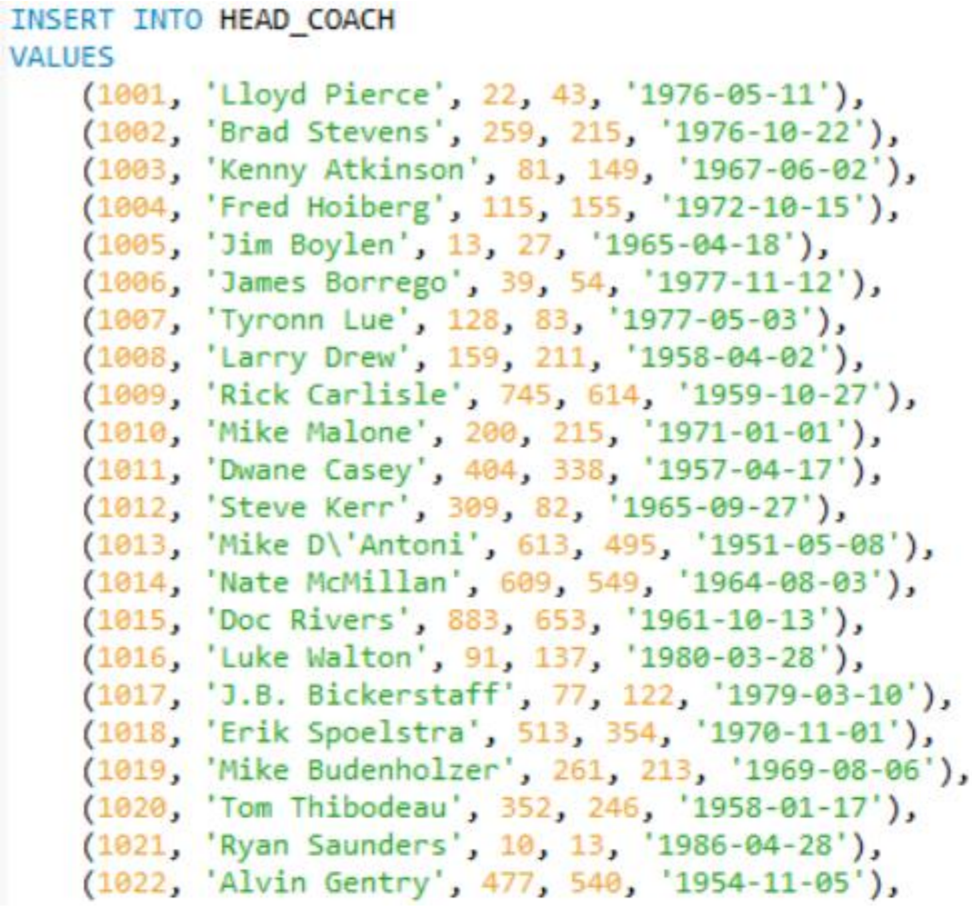
```
DROP TABLE IF EXISTS AWARD_WINNER;
CREATE TABLE AWARD_WINNER
(
    PlayerId        INT unsigned,
    AwardId         INT unsigned,
    YearWon         YEAR,
    CONSTRAINT
        PRIMARY KEY (AwardId, YearWon)
);
```

8.3. TABLE POPULATION

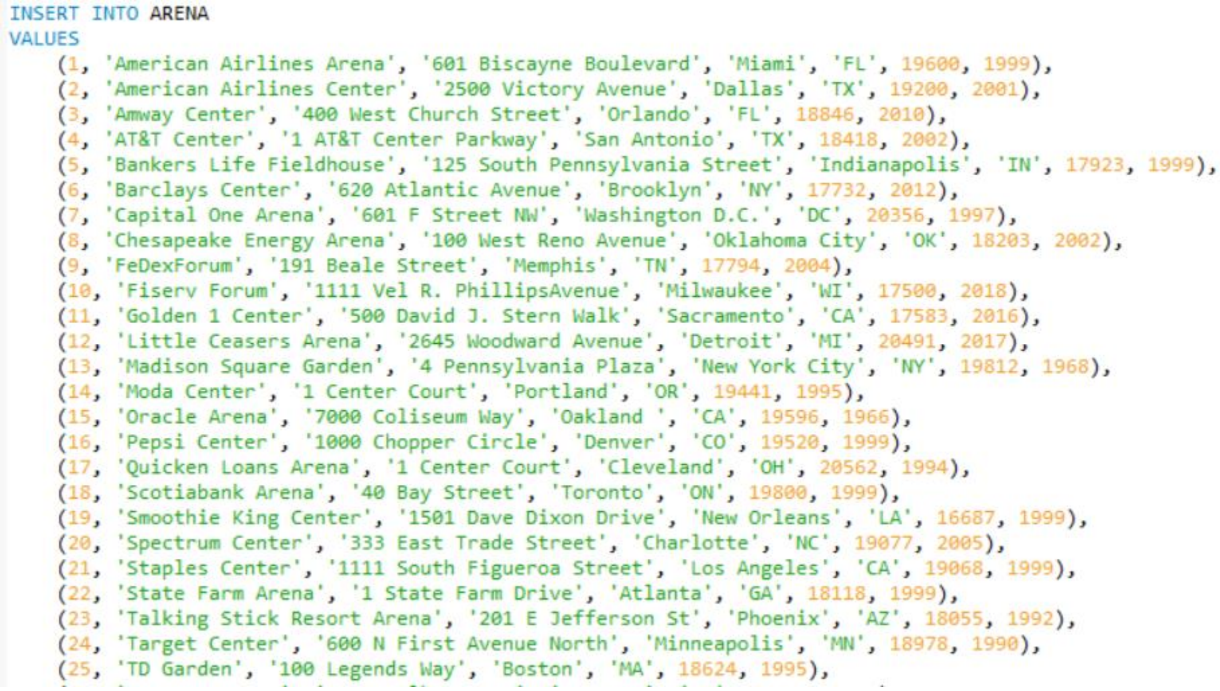
Below are the SQL script and the data inputs for TEAM, HEAD_COACH, and ARENA. The data shown here are not the complete list as the data was too much to fit in this document. In total, there were 1105 tuples inserted to the database.



```
INSERT INTO TEAM
VALUES
    (1, 'Philadelphia 76ers', 1026, 29, 'East', 3, 34, 20),
    (2, 'Portland Trail Blazers', 1028, 14, 'West', 1, 33, 21),
    (3, 'Milwaukee Bucks', 1019, 10, 'East', 1, 40, 13),
    (4, 'Chicago Bulls', 1005, 27, 'East', 6, 12, 42),
    (5, 'Cleveland Cavaliers', 1008, 17, 'East', 1, 11, 43),
    (6, 'Boston Celtics', 1002, 25, 'East', 17, 35, 20),
    (7, 'Los Angeles Clippers', 1015, 21, 'West', NULL, 30, 26),
    (8, 'Memphis Grizzlies', 1017, 9, 'West', NULL, 22, 34),
    (9, 'Atlanta Hawks', 1001, 22, 'East', 1, 18, 36),
    (10, 'Miami Heat', 1018, 1, 'East', 3, 25, 27),
```



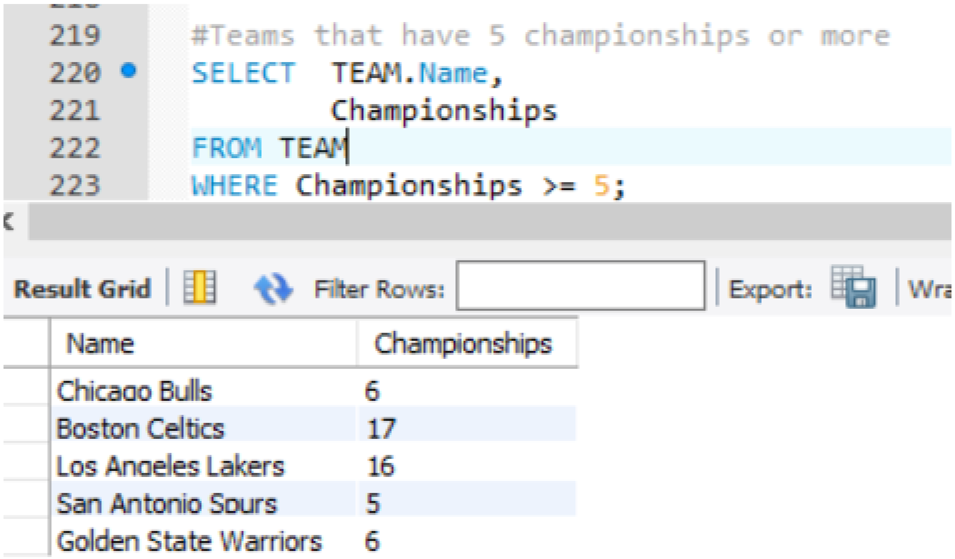
```
INSERT INTO HEAD_COACH
VALUES
    (1001, 'Lloyd Pierce', 22, 43, '1976-05-11'),
    (1002, 'Brad Stevens', 259, 215, '1976-10-22'),
    (1003, 'Kenny Atkinson', 81, 149, '1967-06-02'),
    (1004, 'Fred Hoiberg', 115, 155, '1972-10-15'),
    (1005, 'Jim Boylen', 13, 27, '1965-04-18'),
    (1006, 'James Borrego', 39, 54, '1977-11-12'),
    (1007, 'Tyronn Lue', 128, 83, '1977-05-03'),
    (1008, 'Larry Drew', 159, 211, '1958-04-02'),
    (1009, 'Rick Carlisle', 745, 614, '1959-10-27'),
    (1010, 'Mike Malone', 200, 215, '1971-01-01'),
    (1011, 'Dwane Casey', 404, 338, '1957-04-17'),
    (1012, 'Steve Kerr', 309, 82, '1965-09-27'),
    (1013, 'Mike D\'Antoni', 613, 495, '1951-05-08'),
    (1014, 'Nate McMillan', 609, 549, '1964-08-03'),
    (1015, 'Doc Rivers', 883, 653, '1961-10-13'),
    (1016, 'Luke Walton', 91, 137, '1980-03-28'),
    (1017, 'J.B. Bickerstaff', 77, 122, '1979-03-10'),
    (1018, 'Erik Spoelstra', 513, 354, '1970-11-01'),
    (1019, 'Mike Budenholzer', 261, 213, '1969-08-06'),
    (1020, 'Tom Thibodeau', 352, 246, '1958-01-17'),
    (1021, 'Ryan Saunders', 10, 13, '1986-04-28'),
    (1022, 'Alvin Gentry', 477, 540, '1954-11-05'),
```



```
INSERT INTO ARENA
VALUES
    (1, 'American Airlines Arena', '601 Biscayne Boulevard', 'Miami', 'FL', 19600, 1999),
    (2, 'American Airlines Center', '2500 Victory Avenue', 'Dallas', 'TX', 19200, 2001),
    (3, 'Amway Center', '400 West Church Street', 'Orlando', 'FL', 18846, 2010),
    (4, 'AT&T Center', '1 AT&T Center Parkway', 'San Antonio', 'TX', 18418, 2002),
    (5, 'Bankers Life Fieldhouse', '125 South Pennsylvania Street', 'Indianapolis', 'IN', 17923, 1999),
    (6, 'Barclays Center', '620 Atlantic Avenue', 'Brooklyn', 'NY', 17732, 2012),
    (7, 'Capital One Arena', '601 F Street NW', 'Washington D.C.', 'DC', 20356, 1997),
    (8, 'Chesapeake Energy Arena', '100 West Reno Avenue', 'Oklahoma City', 'OK', 18203, 2002),
    (9, 'FeDexForum', '191 Beale Street', 'Memphis', 'TN', 17794, 2004),
    (10, 'Fiserv Forum', '1111 Vel R. PhillipsAvenue', 'Milwaukee', 'WI', 17500, 2018),
    (11, 'Golden 1 Center', '500 David J. Stern Walk', 'Sacramento', 'CA', 17583, 2016),
    (12, 'Little Ceasers Arena', '2645 Woodward Avenue', 'Detroit', 'MI', 20491, 2017),
    (13, 'Madison Square Garden', '4 Pennsylvania Plaza', 'New York City', 'NY', 19812, 1968),
    (14, 'Moda Center', '1 Center Court', 'Portland', 'OR', 19441, 1995),
    (15, 'Oracle Arena', '7000 Coliseum Way', 'Oakland ', 'CA', 19596, 1966),
    (16, 'Pepsi Center', '1000 Chopper Circle', 'Denver', 'CO', 19520, 1999),
    (17, 'Quicken Loans Arena', '1 Center Court', 'Cleveland', 'OH', 20562, 1994),
    (18, 'Scotiabank Arena', '40 Bay Street', 'Toronto', 'ON', 19800, 1999),
    (19, 'Smoothie King Center', '1501 Dave Dixon Drive', 'New Orleans', 'LA', 16687, 1999),
    (20, 'Spectrum Center', '333 East Trade Street', 'Charlotte', 'NC', 19077, 2005),
    (21, 'Staples Center', '1111 South Figueroa Street', 'Los Angeles', 'CA', 19068, 1999),
    (22, 'State Farm Arena', '1 State Farm Drive', 'Atlanta', 'GA', 18118, 1999),
    (23, 'Talking Stick Resort Arena', '201 E Jefferson St', 'Phoenix', 'AZ', 18055, 1992),
    (24, 'Target Center', '600 N First Avenue North', 'Minneapolis', 'MN', 18978, 1990),
    (25, 'TD Garden', '100 Legends Way', 'Boston', 'MA', 18624, 1995),
```

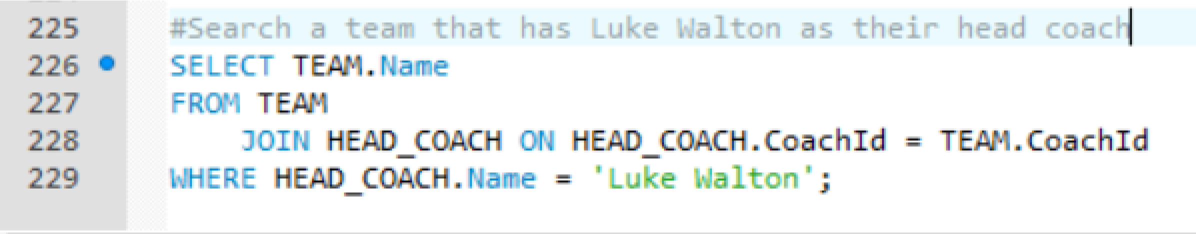
8.4. SAMPLE QUERIES

Below are some sample queries that can be generated from our database.

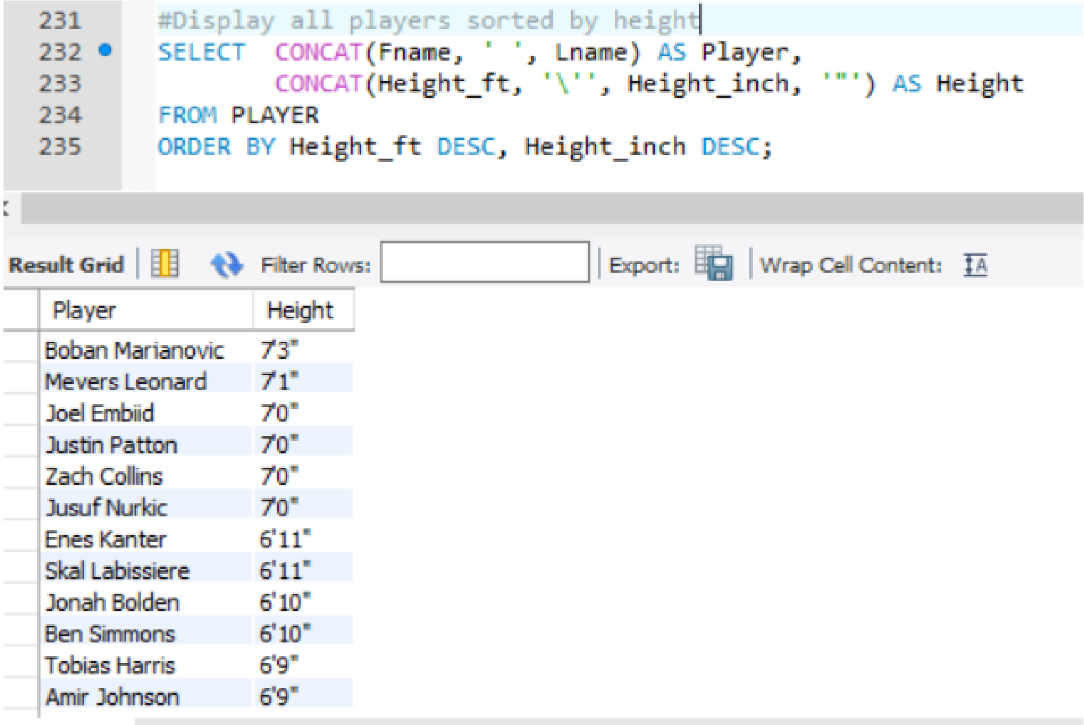


```
#Teams that have 5 championships or more
SELECT  TEAM.Name,
        Championships
FROM TEAM
WHERE Championships >= 5;
```

Name	Championships
Chicago Bulls	6
Boston Celtics	17
Los Angeles Lakers	16
San Antonio Spurs	5
Golden State Warriors	6



```
#Search a team that has Luke Walton as their head coach
SELECT TEAM.Name
FROM TEAM
    JOIN HEAD_COACH ON HEAD_COACH.CoachId = TEAM.CoachId
WHERE HEAD_COACH.Name = 'Luke Walton';
```

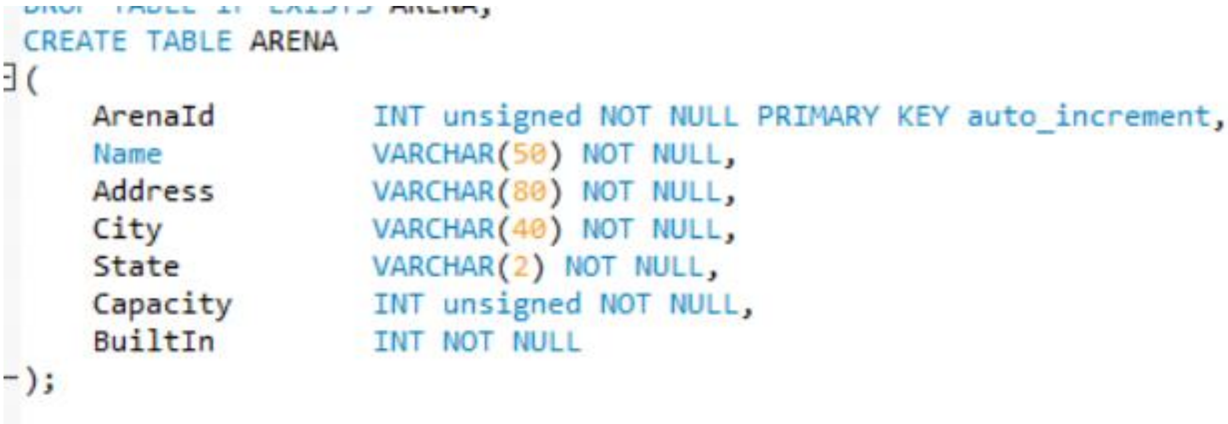


```
#Display all players sorted by height
SELECT  CONCAT(Fname, ' ', Lname) AS Player,
        CONCAT(Height_ft, '\'', Height_inch, '"') AS Height
FROM PLAYER
ORDER BY Height_ft DESC, Height_inch DESC;
```

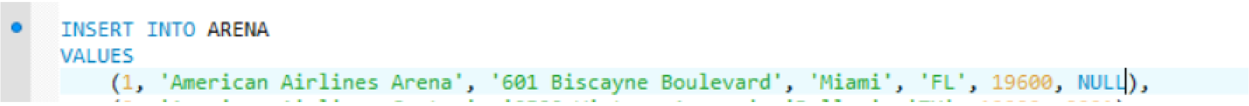
Player	Height
Boban Marianovic	7'3"
Mevers Leonard	7'1"
Joel Embiid	7'0"
Justin Patton	7'0"
Zach Collins	7'0"
Jusuf Nurkic	7'0"
Enes Kanter	6'11"
Skal Labissiere	6'11"
Jonah Bolden	6'10"
Ben Simmons	6'10"
Tobias Harris	6'9"
Amir Johnson	6'9"

8.5. CONSTRAINT VIOLATION TEST CASE

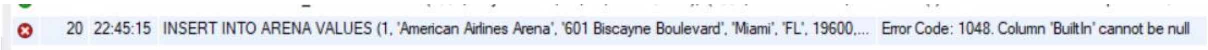
Below is a test case where we try to break the constraint where we put a NULL value in a column that has a NOT NULL constraint. The expected outcome would be an error message, and the test succeeded in giving a warning message.



```
DROP TABLE IF EXISTS ARENA;
CREATE TABLE ARENA
(
    ArenaId         INT unsigned NOT NULL PRIMARY KEY auto_increment,
    Name            VARCHAR(50) NOT NULL,
    Address         VARCHAR(80) NOT NULL,
    City            VARCHAR(40) NOT NULL,
    State           VARCHAR(2) NOT NULL,
    Capacity        INT unsigned NOT NULL,
    BuiltIn         INT NOT NULL
);
```



```
INSERT INTO ARENA
VALUES
    (1, 'American Airlines Arena', '601 Biscayne Boulevard', 'Miami', 'FL', 19600, NULL),
```



```
20  22:45:15  INSERT INTO ARENA VALUES (1, 'American Airlines Arena', '601 Biscayne Boulevard', 'Miami', 'FL', 19600,...  Error Code: 1048. Column 'BuiltIn' cannot be null
```

9. Normalization

In this section we will discuss the functional dependencies of each table and its normalization from. All tables in this database satisfies **BCNF** as all independent attributes are the super key of each table, and all dependent attributes are non-primary attributes.

9.1. TEAM

{TeamId} → {TeamName, CoachId, ArenaId, Conference, Division, Championships, Wins, Losses}

9.2. PLAYER

{PlayerId} → {TeamId, Fname, Lname, DOB, Height_ft, Height_inch, Weight, DraftYear}

9.3. ARENA

{ArenaId} → {ArenaName, Address, City, State, Capacity, BuiltIn}

9.4. HEAD_COACH

{CoachId} → {Name, Wins, Losses, DOB}

9.5. PLAYED_FOR

{PlayerId, TeamId} → {PPG, RPG, APG, BPG, SPG, FG, FG3, FT}

9.6. AWARD

{AwardId} → {Name, Description}

9.7. AWARD_WINNER

{AwardId, YearWon} → {PlayerId}

10. SCHEDULE

Milestones for this project largely mirror the project iteration for each of the following assignments. Dates shown are the date it was finished on or the expected deadline.

10.1. TEAM FORMATION (01-09-2019)

1. Set team by laws
2. Brainstorm project ideas
3. Check feasibility
4. Decide project idea

10.2. DIAGRAM (01-21-2019)

1. Create Entity Relationship Diagram
2. Create Relational Model Schema

10.3. TOOLING SECTION (01-21-2019)

1. Choose DBMS
2. Choose front-end framework tool
3. Choose web-hosting service
4. Choose data source

10.4. DEVELOP DATABASE (03-08-2019)

1. Set up web server
2. Write up database generation code
3. Populate database with data

10.5. FINISH DATABASE AND PRESENTATION (03-18-2019)

1. Develop website to access database
2. Make poster for presentation
3. Present final project to class

11. WORK DISTRIBUTION

With regards to selecting a database idea, our group discussed what level of interest we had in several different ideas to provide input on entity types and selection. During the brainstorming phase, we all provided ideas and input regarding what entities should exist inside an NBA database. We discussed at length regarding what relationships and references should exist between the entities that we identified. When it comes to implementing the database, Jeremy used his experience with MySQL to write code for the database and developed the ER Diagram and RM Schema on draw.io. When it comes to creating the documents for the project, Aaron was able to compile team ideas and record them with proper formatting. With regards to ensuring group collaboration, Brandon provided the framework for pushing group documents to Github and helped practice version control. Nathan worked with the group on setting up the web hosting service for the project. All members of Team Avatar will contribute equally to the voter database application. It's anticipated that each team member will produce 25% of the work overall; however, discrepancies within individual deliverables are expected.

12. TOOLING ASSESSMENT

In this section, we will outline the tools we chose to use for this project and its reasons behind the decision.

12.1. DBMS: MYSQL WORKBENCH CE

Team Avatar has chosen this DBMS because of its ease of use, popularity, and the fact that one of our members has experience with MySQL Workbench CE.

12.2. UI: HTML/CSS

For the front-end side, we have chosen a simple html page with buttons and multiple pages for each table would be sufficient enough for this project as the main focus is the database, not the UX.

12.3. VERSION CONTROL: GITHUB

GitHub is a very well-known platform for collaborators to share their code and it's free to use. Besides that, most of our members have prior experience in using GitHub as a version control platform.

12.4. WEB HOSTING SERVICE: USBWEBSERVER V8.5

USBWebsever is service that ties up all necessary service for a database-backed website that includes Apache, PHP (that also includes PHPMyAdmin), and MySQL.

13. DOCUMENT REVISION HISTORY

In this section, we will discuss what changes each proposal iteration made during the process of completing the project proposal document for this project.

13.1. PROJECT ITERATION #0

- Initialization of project proposal document
- Used Google documents as collaboration platform
- Addressed the applications, entities, queries, data generation, schedule, and work distribution

13.2. PROJECT ITERATION #1

- Added ER diagram and RM schema as an attachment for the project proposal
- Revised entities and attributes to match ER diagram and RM schema

13.3. PROJECT ITERATION #2

- Moved document to Microsoft Word, as google docs lacks consistent and reliable styling features
- Integrated ER diagram and RM schema into the document instead of a separate attachment
- Added table of content and tooling assessment

13.4. PROJECT ITERATION #3

- Added sample SQL script that included, database creation, data inputs, and basic queries
- Added concrete dates to schedule

13.5. PROJECT ITERATION #4

- Added methodology section
- Fixed ER diagram and RM schema
- Fixed entity names to match the updated version of ER and RM
- Added explanation for every entity and its attribute

14. GRADING

Graders can access our website at http://nbadatabase.tk

15. EVALUATION

Looking back to our project, we believe that we did a great job in applying the database concepts that we learned throughout the quarter. The team spent 8 hours per week working on the project, whether it's through a video conference meeting or building the database itself.

The one thing that we believe we could've improved is the scope of the database. Even though it already has 1105 tuples of data inputs, the database is confined to the current NBA season. Besides that, we also decided to not take some data due to the time constraint we have in a quarter system. A database with a much bigger scope, for example the whole history of the NBA, and with more types of data will be the next step of building this class project.

Lastly, we would like to thank Professor Erika Parsons for guiding us throughout the quarter and teaching us the concept of database systems.

REFERENCES

1. “National Basketball Association.” Wikipedia, Wikimedia Foundation, 13 Mar. 2019, http://en.wikipedia.org/wiki/National_Basketball_Association.
2. “Basketball Statistics and History.” *Basketball*, www.basketball-reference.com/.
3. “NBA Basketball Teams.” *CBCnews*, CBC/Radio Canada, http://www.stats.cbc.ca/basketball/nba-teams.aspx?page=/data/nba/teams/teams.html.