Knicks’ Resurrection Project

Charles Mournet - charles.mournet@baruchmail.cuny.edu

Steve Tam - steve.tam@baruchmail.cuny.edu

Greg Martin Teo - gregmartin.teo@baruchmail.cuny.edu

Khang Duong - khang.duong@baruchmail.cuny.edu

CIS 9340

Professor: Dr. Qiang Gao

Thurs 6pm – 9pm

Group 3

**Table of Contents**

1. Executive Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
2. Entity Relationship Model Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . 4
3. Conversion to Relational Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

# Normalization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

# Creating the Database . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

# Application Implementation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

# Navigation Form . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

* 1. Queries . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
  2. Lookup Forms . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
  3. Reports . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

1. Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

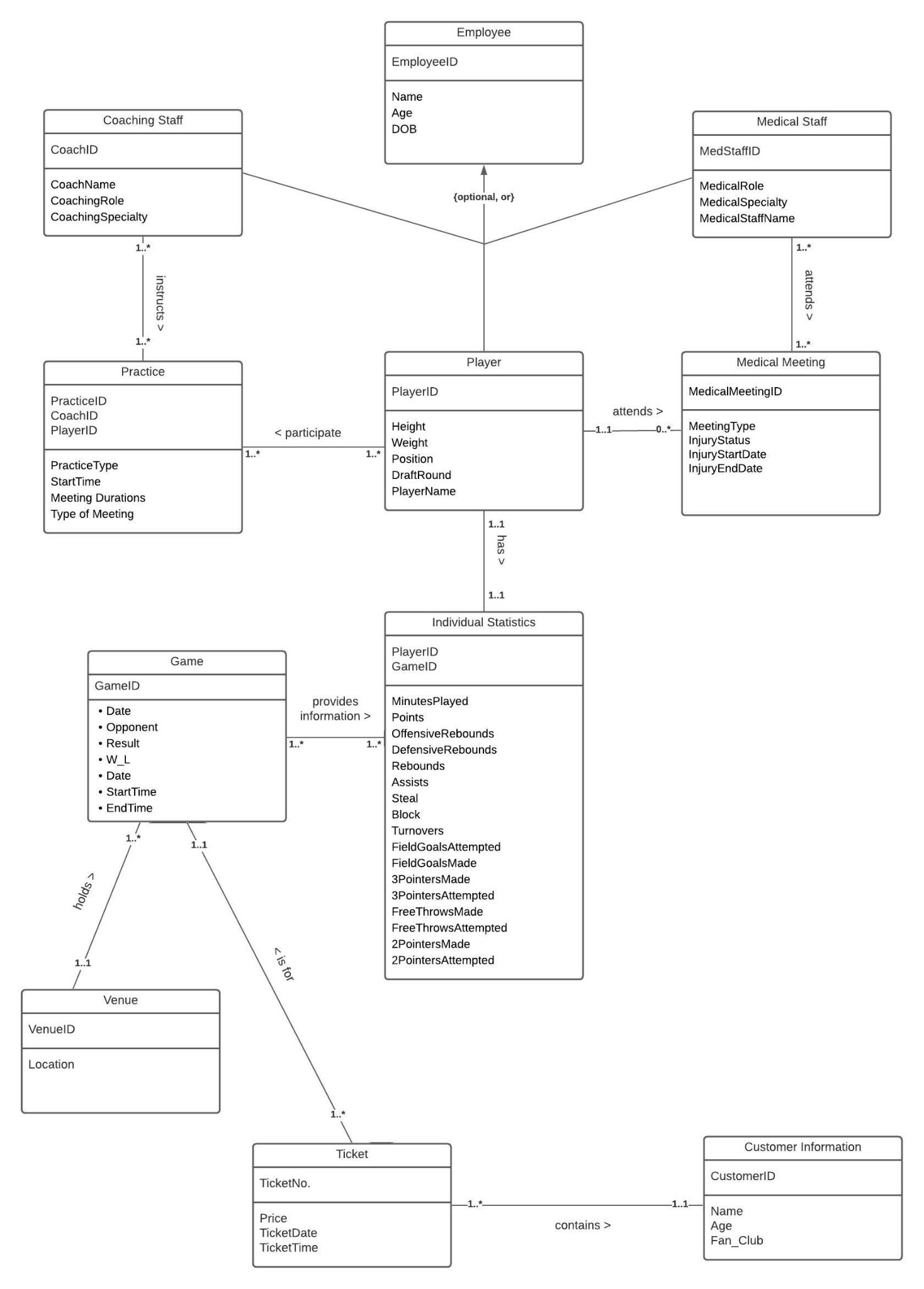
1. **Executive Summary**

The New York Knicks have since fallen behind their competitors in the NBA. For the last 7 consecutive seasons, they have failed to make it to the NBA Playoffs, and it seems that 2020-2021 will not be an exception either. The board has met with the head coach and his staff to discuss a solution to this ongoing problem. The team realized that not keeping proper data storage has created massive inefficiencies for the team in training players, analyzing competitors. We have been brought on to help build a database that keeps track of the player’s performance, staff utilization, and other metrics that can result in useful findings down the road. The Knicks hope that by modernizing their data they will be able to extract key insights and help them get back to their winning ways.

We will start by collecting detailed information on all the players including both physical characteristics and seasonal statistics. We will have data on the coaching staff including relevant characteristics and season-long appointments. The data for the players will be recorded in 2 main focuses, player attributes (name, height, weight, age, draft round, etc.) and game statistics (points per game, assists per game, rebounds per game, steal per game, minutes per game, etc.). These data points will be used to analyze player’s performance along with making predictions and strategies, formations, and player selection for upcoming games. The data for coaching and medical staff will focus on their role and specialties. Furthermore, to ensure that all the staff is being utilized to the best of their abilities, we will record the data for the coaching and medical sessions, with information regarding the type of sessions, which team staff and players attended, and the purpose of each meeting.

Main business scenarios will include examining game statistics to see how players are performing and tracking player injury and medical meetings to know who is currently healthy or dealing with injuries. These medical meetings and doctor patient meetings will be available to the team for internal reference.

1. **Entity Relationship Model Diagram**



Based on the needs of the Knicks described, we developed the Entity Relationship model above. The model not only provides a transparent view of all attributes but also facilitates the creation of relationship sentences to finalize the conceptual ER model.

**Relationship Sentences:**

One **Employee** *must be* a **Coaching Staff, Medical Staff,** or **Player**.

One **Coaching Staff** *must* instruct one or more **Practice**.

One **Practice** *must be*instructed by one or more **Coaching Staff.**

One **Medical Staff** *must* attend one or more **Medical Meeting.**

One **Medical Meeting** *must be* attendedby one or more **Medical Staff.**

One **Player** *must* participate in one or more **Practice.**

One **Player** *may* attend one or more **Medical Meeting.**

One **Medical Meeting** *must be* attended by one and only one **Player.**

One **Player** *must* have one and only one **Individual Statistics.**

One **Individual Statistics** *must* belong to one and only one **Player.**

One **Game** *must* provide one or more **Individual Statistics.**

One **Individual Statistics** *must be* provided by one or more **Game.**

One **Venue** *must* hold one or more **Game.**

One **Game** *must be* held in one and only one **Venue**.

One **Ticket** *must* *be* for one and only one **Game.**

One **Game** *must* have one or more **Ticket.**

One **Customer Information** *must* *be* contained in one or more **Ticket.**

One **Ticket** *must* contain one or more Customer Information

1. **Conversion to Relational Model**

Following the ER model and the relationship sentences, our next step was to convert our conceptual ER model to a logical Relational model. Below is our initial set of relations with each Entity’s identifying attribute dictated by the primary key and the entity’s relationships mapped as foreign keys.

* Employee (EmployeeID (pk), PracticeID (fk), MedicalMeetingID (fk), CoachID, MedStaffID, PlayerID, Name, Age, DOB, CoachingRole, CoachingSpecialty, MedicalRole, MedicalSpecialty, Height, Weight, Position, DraftRound, PlayerName, CoachName, MedStaffName)
* MedicalMeeting (MedicalMeetingID (pk), MedStaffID (fk), PlayerID (fk), MeetingType, InjurySatus, InjuryStartDate, InjuryEndDate, MedStaffName (fk))
* Practice (PracticeID (pk), CoachID (fk), PlayerID (fk), PracticeType, CoachName (fk), PlayerName (fk), StartTime, Meeting Durations, Type of Meeting)
* IndividualStatistics (PlayerID (pk) (fk), GameID (fk), MinutesPlayed, Points, OffensiveRebounds, DefensiveRebounds, Rebounds Assists, Steal Block, Turnovers, FieldGoalsAttempted, FieldGoalsMade, FieldGoalPercentage, 3PointersMade, 3PointersAttempted, FreeThrowsMade, FreeThrowsAttempted, 2PointersMade, 2PointersAttempted,)
* Game (GameID (pk), PlayerID (fk), VenueID (fk), Date, Opponent, Result, W-L, StartTime, EndTime)
* Ticket( TicketNo(pk), VenueID(fk), Price, Date, Time)
* Venue (VenueID(pk), GameID(fk), TicketNo(fk),CustomerID (fk), Location, Date)
* CustomerInformation (CustomerID (pk), TicketNo (fk), Name, Age, Fan Club)

# **Normalization**

Before we can use the set of relations to create the database, we had to normalize each relation from 1NF up to BCNF. By normalizing the relations up to BCNF, we ensure that each relation is the following:

1NF: Meets the definition of a relation

2NF: Will have no partial dependencies

3NF: Will have no transitive dependencies

BCNF: Will have all determinants as candidate keys.

**Employee**

Employee (EmployeeID(pk), Name, Age, DOB, MedStaffID(fk), MedStaffName, MedicalRole, MedicalSpecialty, CoachID(fk), CoachName, CoachingRole, CoachingSpecialty, Players(fk), PlayerName, Height, DraftRound

PK EmployeeID

FD1: EmployeeID -> MedStaffID, CoachID, PlayerID, Name, Age, DOB, MedStaffName, MedicalRole, MedicalSpecialty, CoachName, CoachingRole, CoachingSpecialty, PlayerName, Height, Draftround

FD2: EmployeeID -> Name, Age, DOB

FD3: MedStaffID -> MedStaffName, MedicalRole, MedicalSpecialty

FD4: CoachID -> CoachName, CoachingRole, CoachingSpecialty

FD5: PlayerID -> PlayerName, Height, Weight, DraftRound,

1NF: Meets the definition of a relation

There is partial dependency so we are not at 2NF yet

R1(EmployeeID, Name, Age, DOB)

R2(EmployeeID, MedStaffID, CoachID, PlayerID, MedStaffName, MedicalRole, MedicalSpecialty, CoachName, CoachingRole, CoachingSpecialty, PlayerName, Height, Draftround)

FD3: MedStaffID -> MedStaffName, MedicalRole, MedicalSpecialty

FD4: CoachID -> CoachName, CoachingRole, CoachingSpecialty

FD5: PlayerID -> PlayerName, Height, Weight, DraftRound

2NF: There is no partial dependency so we are at 2NF

There are transitive dependencies so we are not at 3NF yet

R3(MedStaffID, MedStaffName, MedicalRole, MedicalSpecialty,)

R4(CoachID, CoachName, CoachingRole, CoachingSpecialty)

R5(PlayerID, PlayerName, Height, Weight, DraftRound,)

R6(EmployeeID, CoachID, PlayerID, MedStaffID)

3NF: No transitive dependency

BCNF: all determinants are candidate key

**Medical Meeting**

MedicalMeeting (MedicalMeetingID(pk), MedStaffID(fk), PlayerID (fk), MeetingType, InjurySatus, InjuryStartDate, InjuryEndDate)

PK: MedicalMeetingID

FD1: MedicalMeetingID -> MedStaffID, PlayerID, MeetingType, InjuryStatus, InjuryStartDate, InjuryEndDate,

FD2: MedStaffID -> MedStaffName

1NF: Meets the definition of a relation

2NF: No partial dependency

There are transitive dependencies so we are not at 3NF

R1 (MedStaffID, MedStaffName)

FD1: MedStaffID -> MedStaffName

R2 (MedicalMeetingID(pk), MedStaffID(fk), PlayerID(fk), MeetingType, InjuryStatus, InjuryStartDate, InjuryEndDate)

FD1: MedicalMeetingID -> MedStaffID, PlayerID, MeetingType, InjuryStatus, InjuryStartDate, InjuryEndDate)

3NF: No transitive dependency

BCNF: all determinants are candidate key

**Practice**

Practice (PracticeID (pk), CoachID (pk)(fk), PlayerID (pk)(fk), PracticeType, CoachName, PlayerName, StartTime, PracticeDurations, PracticeDate)

PK: PracticeID

FD1: PracticeID -> CoachID, PlayerID, PracticeType, CoachName, PlayerName, StartTime, PracticeDurations, PracticeDate

FD2: CoachID -> CoachName

FD3: PlayerID -> PlayerName

1NF: Meets the definition of a relation

2NF: No partial dependency

There are transitive dependencies so we are not at 3NF

R1(CoachID, CoachName)

FD1: CoachID -> CoachName

R2(PlayerID, PlayerName)

FD1: PlayerID -> PlayerName

R3( PracticeID(pk), CoachID(fk), PlayerID(fk), PracticeType, StartTime, Meeting Durations, Type of Meeting)

FD1: PracticeID -> CoachID, PlayerID, PracticeType, StartTime, Meeting Durations, Type of Meeting

3NF: No transitive dependency

BCNF: all determinants are candidate key

**Individual Statistics**

IndividualStatistics (PlayerID (pk) (fk), GameID (fk), MinutesPlayed, Points, OffensiveRebounds, DefensiveRebounds, Rebounds, Assists, Steals, Blocks, Turnovers, FieldGoalsAttempted, FieldGoalsMade, 3PointersMade, 3PointersAttempted, FreeThrowsMade, FreeThrowsAttempted, 2PointersMade, 2PointersAttempted)

PK: PlayerID

FD1: PlayerID -> GameID, MinutesPlayed, Points, OffensiveRebounds, DefensiveRebounds, Rebounds Assists, Steal Block, Turnovers, FieldGoalsAttempted, FieldGoalsMade, 3PointersMade, 3PointersAttempted, FreeThrowsMade, FreeThrowsAttempted, 2PointersMade, 2PointersAttempted)

1NF: Meets the definition of a relation

2NF: No partial dependency

3NF: No transitive dependency

BCNF: all determinants are candidate key

**Game**

Game (GameID (pk), VenueID (fk), GameDate, Opponent, Result, W-L, StartTime, EndTime)

PK: GameID

FD: GameID -> PlayerID, VenueID, GameDate, Opponent, Result, W-L, StartTime, EndTime

1NF: Meets the definition of a relation

2NF: No partial dependency

3NF: No transitive dependency

BCNF: all determinants are candidate key

**Venue**

Venue (VenueID(pk), Location)

PK: VenueID

FD: VenueID -> Location

1NF: Meets the definition of a relation

2NF: No partial dependency

3NF: No transitive dependency

BCNF: all determinants are candidate key

**Ticket**

Ticket(TicketNo.(pk), CustomerID(fk), GameID (fk), Price, TicketDate, TicketTime)

PK: TicketNo.

FD: TicketNo. -> CustomerID, GameID, Price, TicketDate, TicketTime

1NF: Meets the definition of a relation

2NF: No partial dependency

3NF: No transitive dependency

BCNF: all determinants are candidate key

**Customer Information**

CustomerInformation (CustomerID (pk), Name, Age, Fan Club)

PK: CustomerID

FD: CustomerID -> Name, Age, Fan Club

1NF: Meets the definition of a relation

2NF: No partial dependency

3NF: No transitive dependency

BCNF: all determinants are candidate key

**Final Set of Relations:**

**Employee** (EmployeeID(pk), Name, Age, DOB, MedStaffID(fk), MedStaffName, MedicalRole, MedicalSpecialty, CoachID(fk), CoachName, CoachingRole, CoachingSpecialty, Players(fk), PlayerName, Height, DraftRound)

**MedicalMeeting** (MedicalMeetingID(pk), MedStaffID(fk), PlayerID (fk), MeetingType, InjurySatus, InjuryStartDate, InjuryEndDate)

**Practice** (PracticeID (pk), CoachID (pk)(fk), PlayerID (pk)(fk), PracticeType, CoachName, PlayerName, StartTime, PracticeDurations, PracticeDate)

**IndividualStatistics** (PlayerID (pk) (fk), GameID (fk), MinutesPlayed, Points, OffensiveRebounds, DefensiveRebounds, Rebounds, Assists, Steals, Blocks, Turnovers, FieldGoalsAttempted, FieldGoalsMade, 3PointersMade, 3PointersAttempted, FreeThrowsMade, FreeThrowsAttempted, 2PointersMade, 2PointersAttempted)

**Game** (GameID (pk), VenueID (fk), GameDate, Opponent, Result, W-L, StartTime, EndTime)

**Venue** (VenueID(pk), Location)

**Ticket** (TicketNo.(pk), CustomerID(fk), GameID (fk), Price, TicketDate, TicketTime)

**CustomerInformation** (CustomerID (pk), Name, Age, Fan Club)

# **Creating Tables**

Now that we have our normalized set of relations, we can move on to the creation of the database using the following SQL code:

CREATE TABLE Employee (

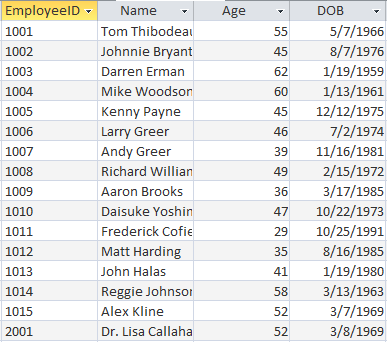
EmployeeID VARCHAR(10) NOT NULL,

Name VARCHAR(100) NOT NULL,

Age NUMBER NOT NULL,

DOB DATE NOT NULL

)



CREATE TABLE Coaching\_Staff (

CoachID VARCHAR(10) NOT NULL,

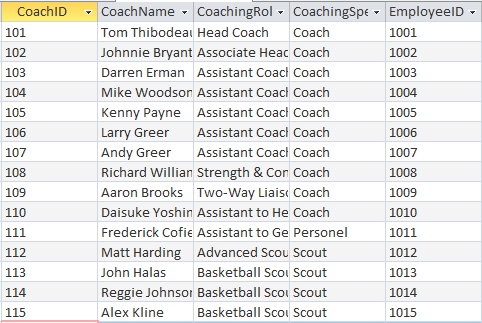
CoachName VARCHAR(100) NOT NULL,

CoachingRole VARCHAR(40) NOT NULL,

CoachingSpecialty VARCHAR(50) NOT NULL,

EmployeeID VARCHAR(10) NOT NULL

)



CREATE TABLE Medical\_Staff (

MedStaffID VARCHAR(10) NOT NULL,

MedStaffName VARCHAR(100) NOT NULL,

MedicalRole VARCHAR(40) NOT NULL,

MedicalSpecialty VARCHAR(50) NOT NULL,

EmployeeID VARCHAR(10) NOT NULL

)



CREATE TABLE Players (

PlayerID VARCHAR(10) NOT NULL,

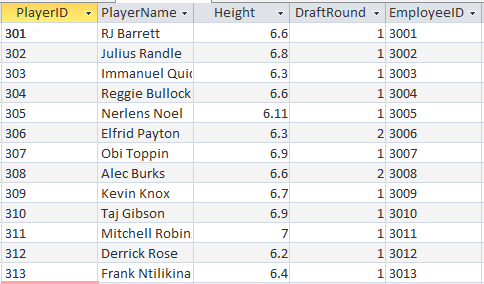
PlayerName VARCHAR(40) NOT NULL,

Height NUMBER NOT NULL,

DraftRound NUMBER NOT NULL,

EmployeeID VARCHAR(10) NOT NULL

)



CREATE TABLE Medical\_Meetings (

MedicalMeetingID VARCHAR(10) NOT NULL,

MedStaffID VARCHAR(10) NOT NULL,

PlayerID VARCHAR(10) NOT NULL,

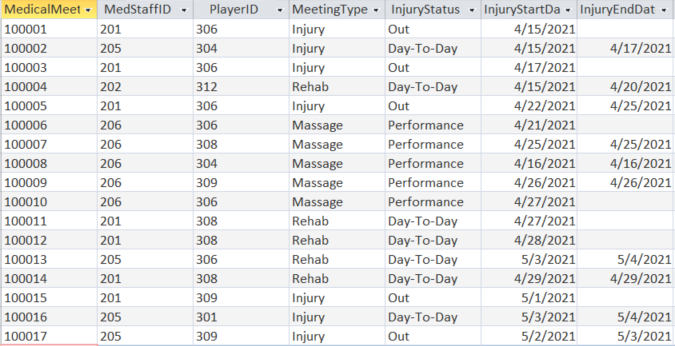
MeetingType VARCHAR(80) NOT NULL,

InjuryStatus VARCHAR(80) NOT NULL,

InjuryStartDate DATE NOT NULL,

InjuryEndDate DATE

)



CREATE TABLE Practice (

PracticeID VARCHAR(10) NOT NULL,

CoachID VARCHAR(10) NOT NULL,

PlayerID VARCHAR(10) NOT NULL,

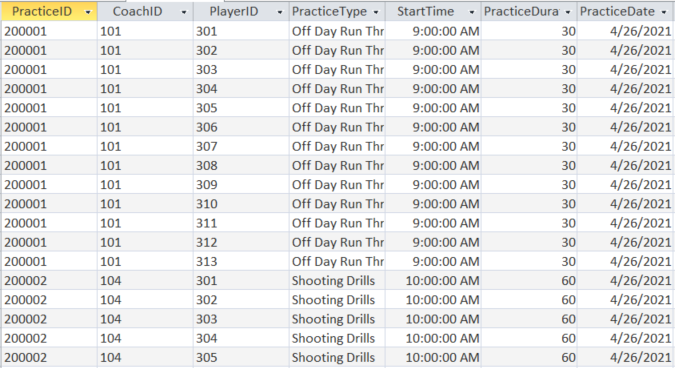
PracticeType VARCHAR(50) NOT NULL,

StartTime TIME NOT NULL,

PracticeDurations NUMBER NOT NULL,

PracticeDate DATE NOT NULL

)



CREATE TABLE IndividualStatistics (

PlayerID VARCHAR(10) NOT NULL,

GameID VARCHAR(10) NOT NULL,

MinutesPlayed NUMBER,

Points NUMBER,

OffensiveRebounds NUMBER,

DefensiveRebounds NUMBER,

Rebounds NUMBER,

Assists NUMBER,

Steals NUMBER,

Blocks NUMBER,

Turnovers NUMBER,

FieldGoalsMade NUMBER,

FieldGoalsAttempted NUMBER,

3PointersMade NUMBER,

3PointersAttempted NUMBER,

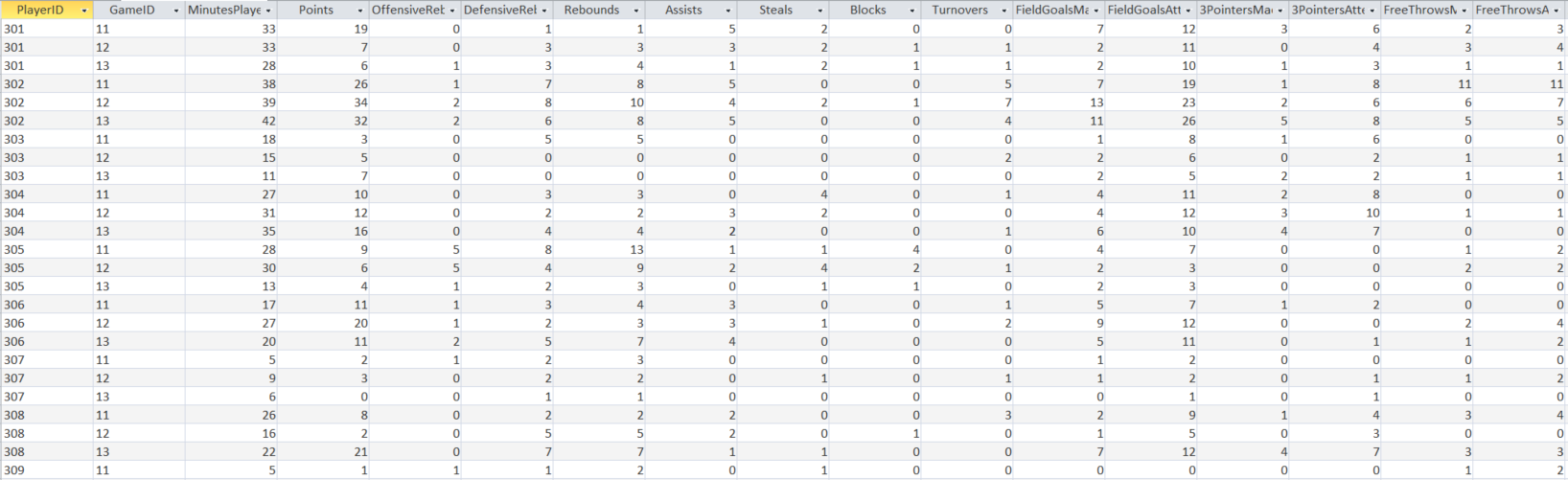
FreeThrowsMade NUMBER,

FreeThrowsAttempted NUMBER,

2PointersMade NUMBER,

2PointersAttempted NUMBER

)



CREATE TABLE Game (

GameID VARCHAR(10) NOT NULL,

VenueID VARCHAR(10) NOT NULL,

GameDate DATE,

Opponent VARCHAR(30),

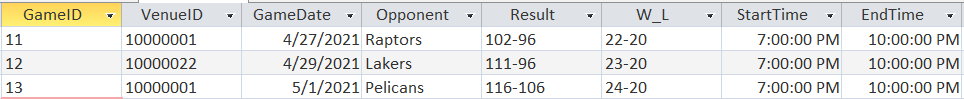
Result VARCHAR(10),

W\_L VARCHAR(10),

StartTime TIME,

EndTime TIME

)



CREATE TABLE Ticket (

TicketNo VARCHAR(10) NOT NULL,

CustomerID VARCHAR(10) NOT NULL,

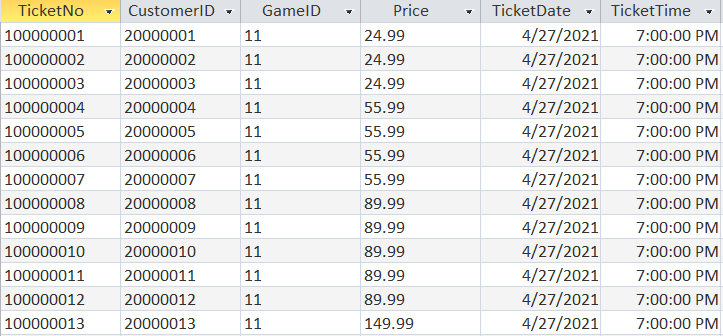
GameID VARCHAR(10) NOT NULL,

Price VARCHAR(10) NOT NULL,

TicketDate DATE,

TicketTime TIME NOT NULL

)

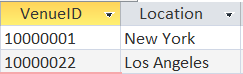


CREATE TABLE Venue (

VenueID VARCHAR(10) NOT NULL,

Location VARCHAR(40) NOT NULL,

)



CREATE TABLE Customer\_Information (

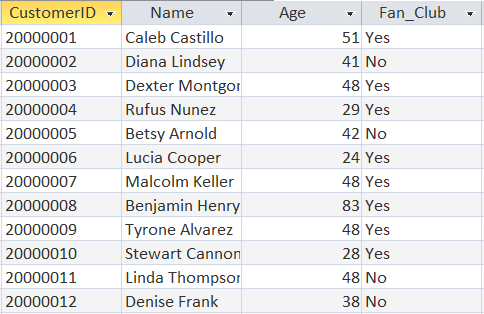
CustomerID VARCHAR(10) NOT NULL,

Name VARCHAR(40) NOT NULL,

Age NUMBER NOT NULL,

Fan\_Club VARCHAR(40) NOT NULL

)



**Adding Primary and Foreign Keys**

ALTER TABLE Employee

ADD CONSTRAINT pk\_employees PRIMARY KEY (EmployeeID)

ALTER TABLE Coaching\_Staff

ADD CONSTRAINT pk\_coaching\_staff PRIMARY KEY (CoachID),

CONSTRAINT fk\_coaching\_staff FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID)

ALTER TABLE Medical\_Staff

ADD CONSTRAINT pk\_medical\_staff PRIMARY KEY (MedStaffID),

CONSTRAINT fk\_medical\_staff FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID)

ALTER TABLE Players

ADD CONSTRAINT pk\_players PRIMARY KEY (PlayerID),

CONSTRAINT fk\_player FOREIGN KEY (EmployeeID) REFERENCES Employee (EmployeeID)

ALTER TABLE Practice

ADD CONSTRAINT pk\_practice PRIMARY KEY (PracticeID, PlayerID, CoachID),

CONSTRAINT fk1\_practice FOREIGN KEY (PlayerID) REFERENCES Players (PlayerID),

CONSTRAINT fk2\_practice FOREIGN KEY (CoachID) REFERENCES Coaching\_Staff(CoachID)

ALTER TABLE Medical\_Meetings

ADD CONSTRAINT pk\_medical\_meeting PRIMARY KEY (MedicalMeetingID),

CONSTRAINT fk1\_medical\_meeting FOREIGN KEY (PlayerID) REFERENCES Players (PlayerID),

CONSTRAINT fk2\_medical\_meeting FOREIGN KEY (MedStaffID) REFERENCES Medical\_Staff (MedStaffID)

ALTER TABLE Venue

ADD CONSTRAINT pk\_venue PRIMARY KEY (VenueID)

ALTER TABLE Game

ADD CONSTRAINT pk\_game PRIMARY KEY (GameID)

ALTER TABLE Game

ADD CONSTRAINT fk\_game FOREIGN KEY (VenueID) REFERENCES Venue (VenueID)

ALTER TABLE Customer\_Information

ADD CONSTRAINT pk\_Customer\_Information PRIMARY KEY (CustomerID)

ALTER TABLE Ticket

ADD CONSTRAINT pk\_Ticket PRIMARY KEY (TicketNo),

CONSTRAINT fk1\_Ticket FOREIGN KEY (CustomerID) REFERENCES Customer\_Information (CustomerID)

CONSTRAINT fk2\_Ticket FOREIGN KEY (GameID) REFERENCES Game (GameID)

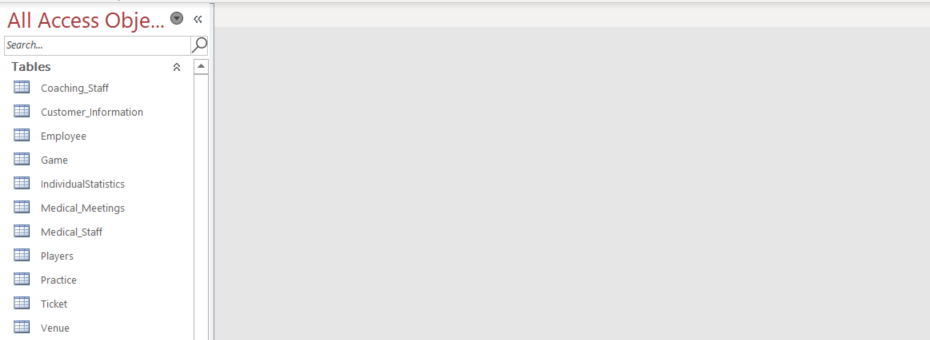
ALTER TABLE IndividualStatistics

ADD CONSTRAINT pk\_individual\_statistics PRIMARY KEY (PlayerID, GameID),

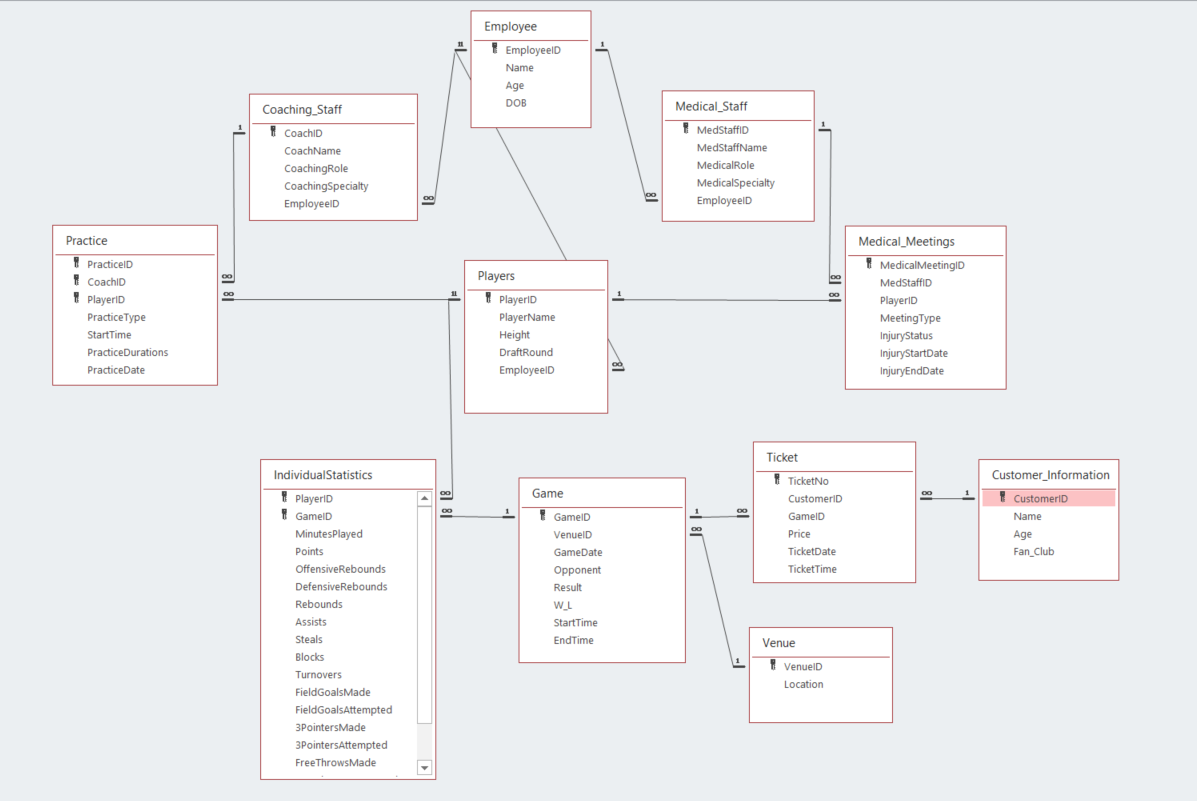
CONSTRAINT fk1\_individual\_statistics FOREIGN KEY (PlayerID) REFERENCES Players (PlayerID),

CONSTRAINT fk2\_individual\_statistics FOREIGN KEY (GameID) REFERENCES Game (GameID)

**Database Schema**



**Relationship View**



**Adding Data to the table using SQL INSERT statements**

INSERT INTO Employee VALUES ("1001", "Tom Thibodeau", 45, "5/7/1966");

INSERT INTO Coaching\_Staff VALUES ("101", "Tom Thibodeau", "Head Coach", "Coach", "1001");

INSERT INTO Players VALUES ("301", "RJ Barrett", 6.6, 1, "3001");

INSERT INTO Medical\_Staff VALUES ("201", "Dr. Lisa Callahan", "Chief Medical Officer, "Injury", "2001");

INSERT INTO Practice VALUES ("200001", "101", "301", "Off Day Run Through", "9:00, 30, "4/26/2021");

INSERT INTO Medical\_Meeting VALUES ("100001", "201", "301", "Injury", "Out", 4/15/2021");

INSERT INTO IndividualStatistics VALUES ("301", "11", 33, 19, 0, 1, 1, 5, 2, 0, 0, 7, 12, 3, 6, 2, 3, 4, 6);

INSERT INTO Game VALUES ("11", "301", "10000001", "4/27/2021", "Raptors", "102-96", "22-20", "19:00", "22:00");

INSERT INTO Customer\_Information VALUES ("20000001", "100000001", "Caleb Castillo", 51, "Yes");

INSERT INTO Ticket VALUES ("100000001", "20000001", "11", 24.99, "4/27/2011", "19:00");

INSERT INTO Venue VALUES ("10000001", "New York");

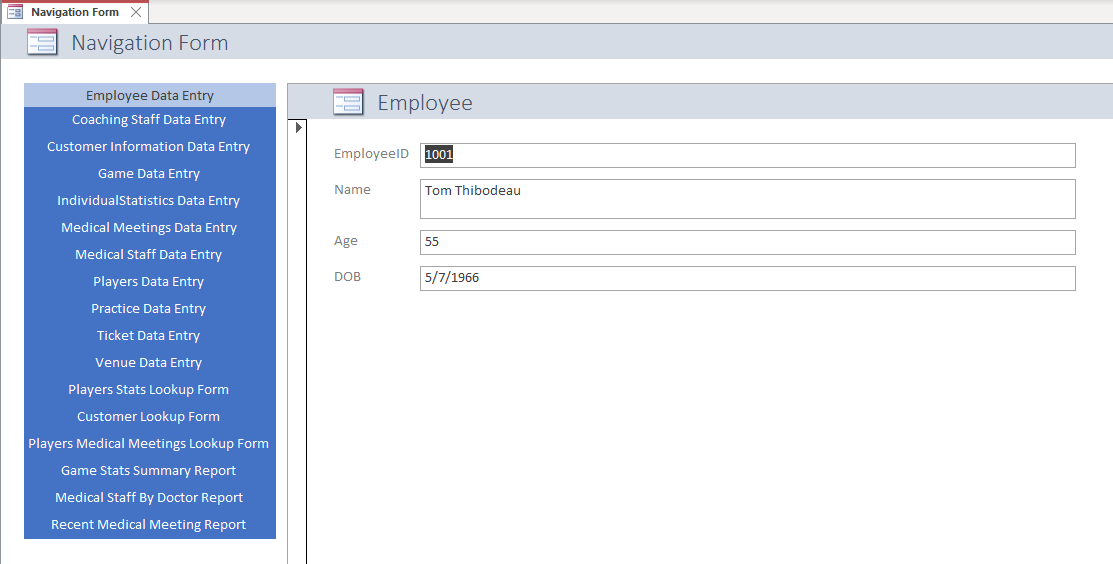
# 

# 

# 

# **Application Implementation**

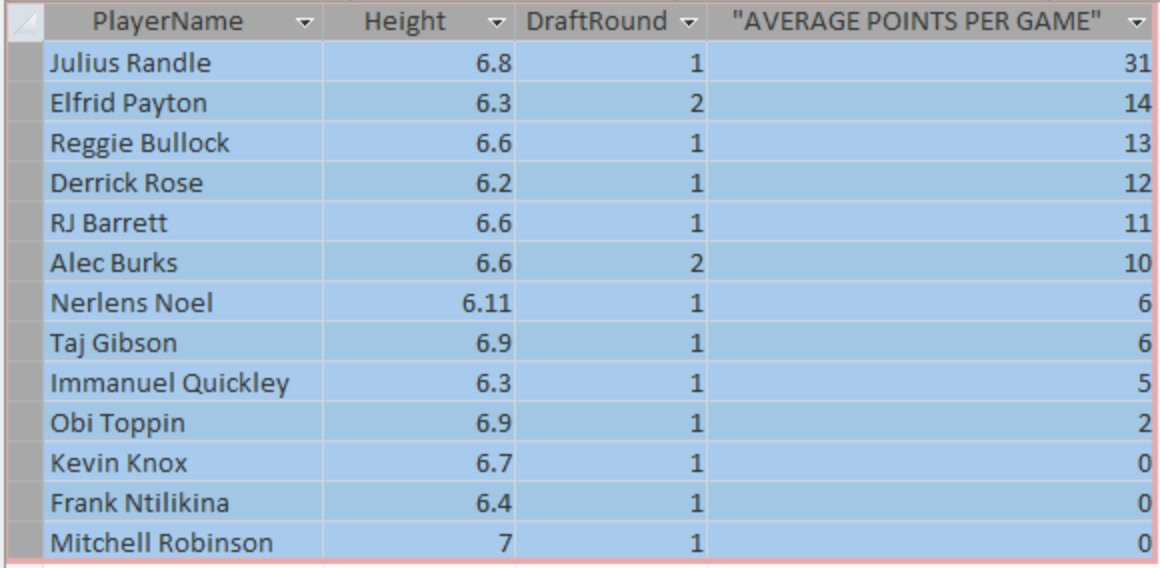
**NAVIGATION FORM**

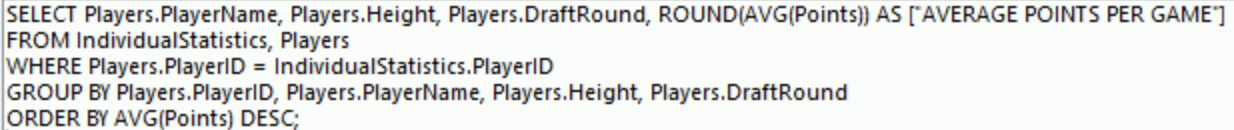
****

**QUERIES**

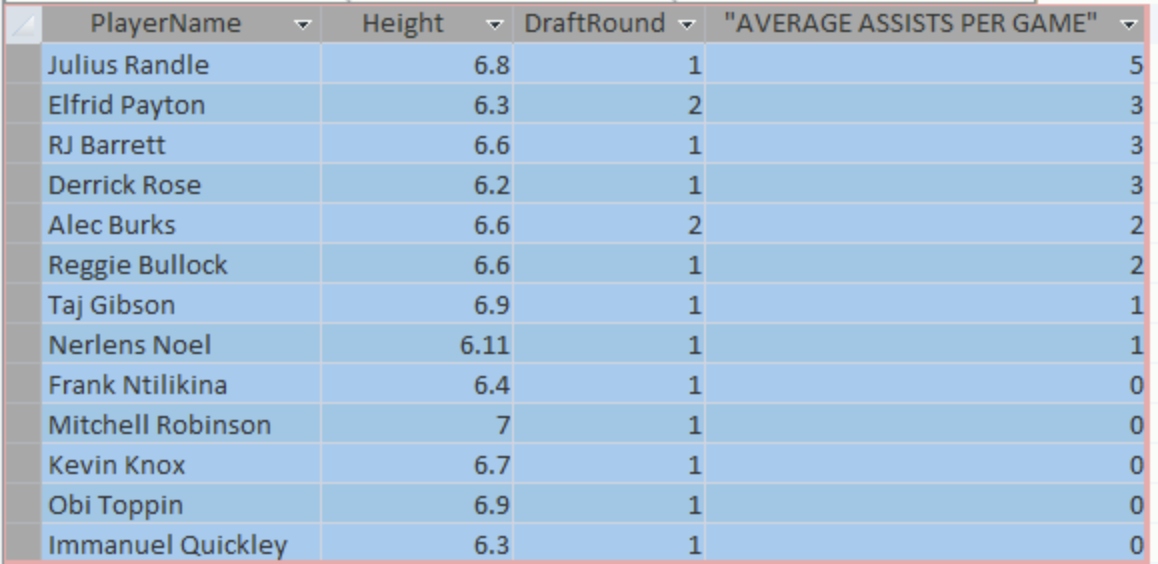
The queries are developed for the data analyst to support the coaches, medical staff, and the players with the latest statistics, information and insights into each player, each opponent. Therefore, we have created a few example of those:

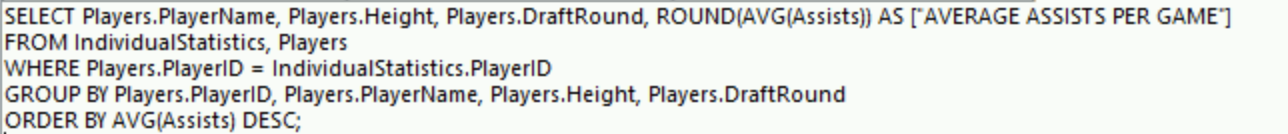
Information for players with highest PPG:



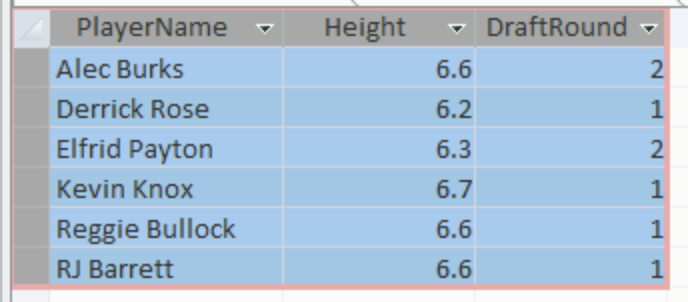


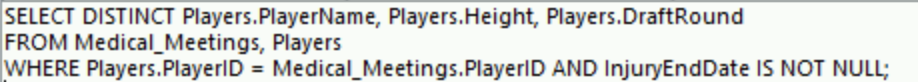
Information for players with highest APG:





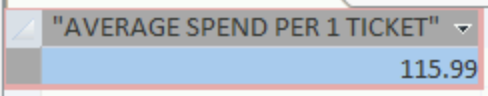
Information for players who are recovered from injury:





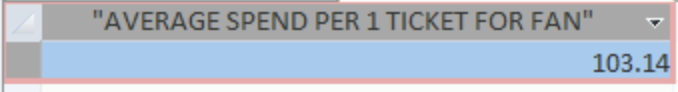
We also developed queries to be used to analyze revenue. In this, you can see that we are calculating average spend on 1 ticket among fans and non-fans and everyone. In order to do this, we are able to see if we can leverage any activities to engage with fan betters to increase the team revenue or to see if we need to focus on non-fans in the near future.

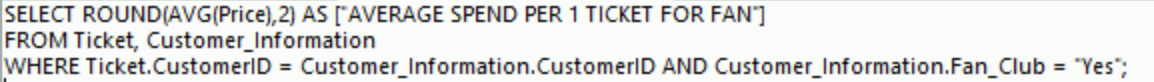
Average Spend per 1 Ticket:



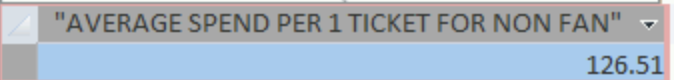


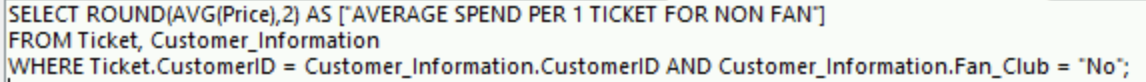
Average Spend per 1 Ticket by Fans:





Average Spend per 1 Ticket by non-Fans:



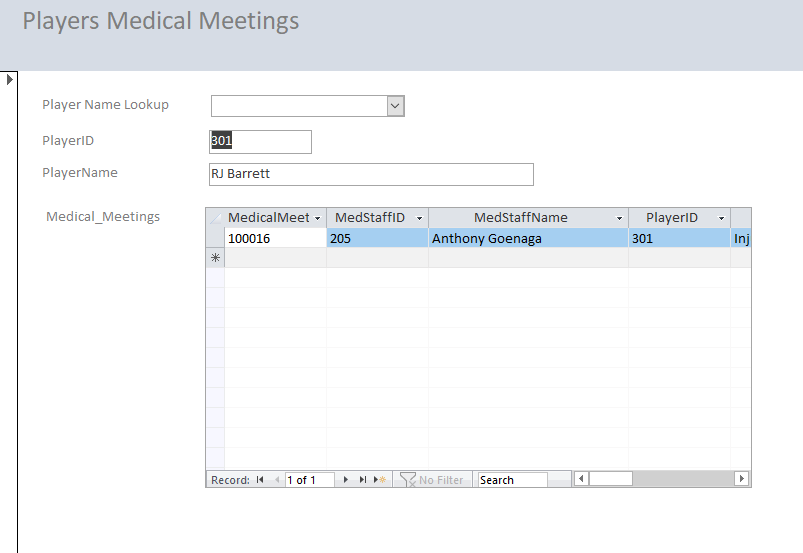


# 

# **Lookup Forms**

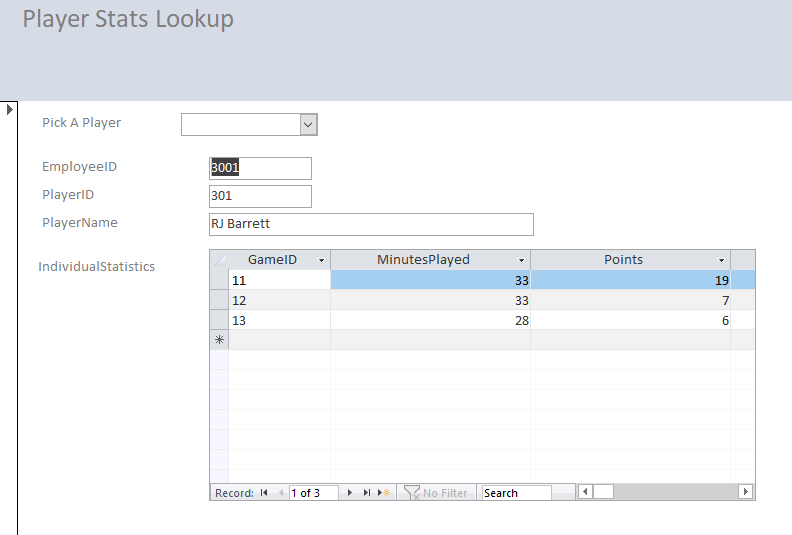
Customer Lookup Form: The Fan lookup form pulls all information from the customer ID, and ticket table. This allows the Knicks team staff to search for information by client name. This has many potential useful cases; if a fan has been selected for an event, any promotional marketing, as well as keeping track of VIP customers, keeping close track

of customer information is very valuable for the organization.



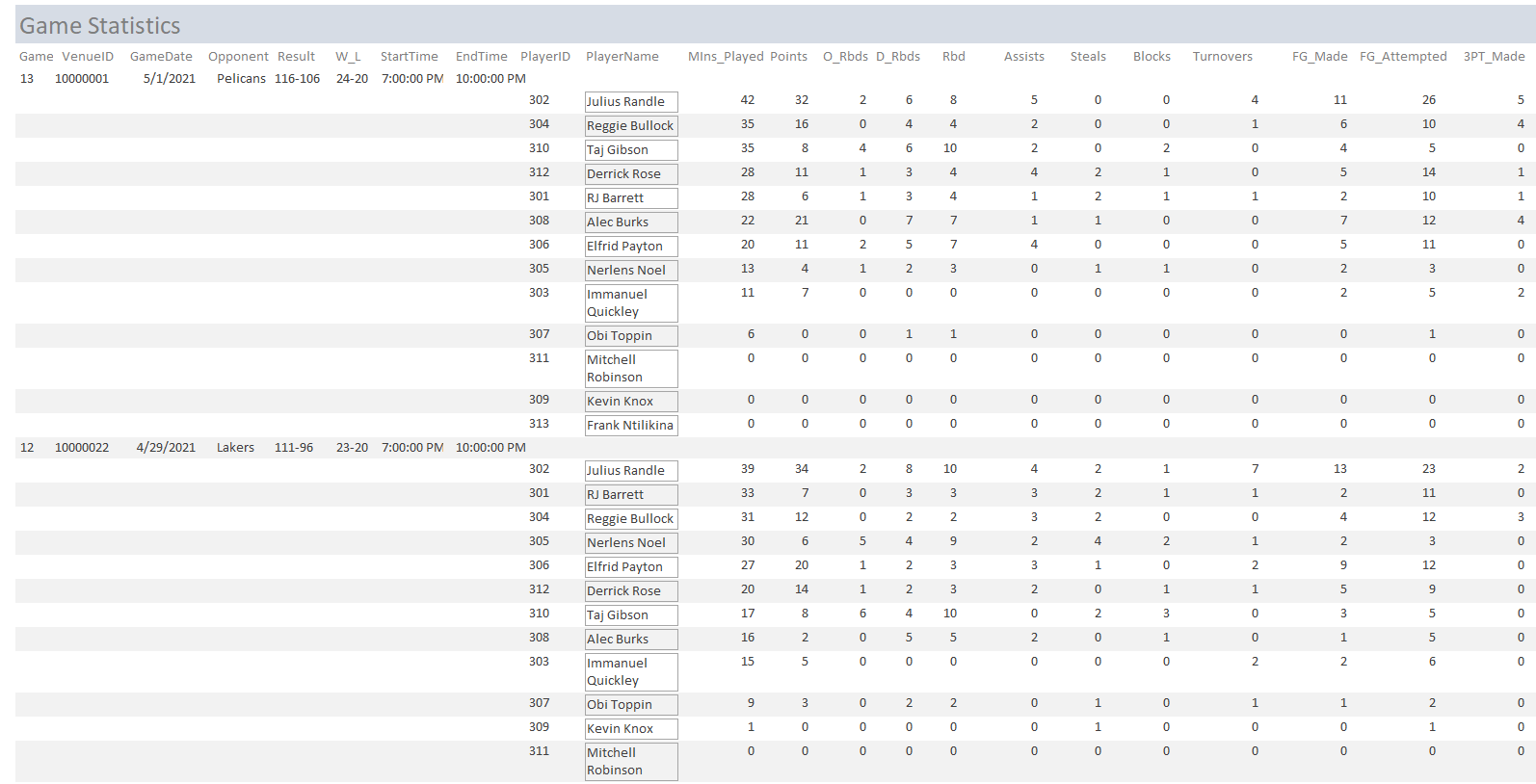
Players Medical Meeting: The player medical meeting form allows for all recent medical meetings to be shown for the player selected. The player name lookup provides a drop down of the names on the team and selecting a player shows their meeting info. This is useful for a medical staff to reference to see a players recent injury history, as well as the coaching staff to reference and monitor what is happening with the player.

Player Stats Lookup Form: This is the most fundamental of forms for the team to have. The player stats is important for the coaches to monitor to understand performance, as well as keeping track of usage time. All stats are monitored and displayed for all of the players' games. The Pick a Player is a drop down menu that allows for a player to be selected. These statistics are also published on the team site and available to the media.



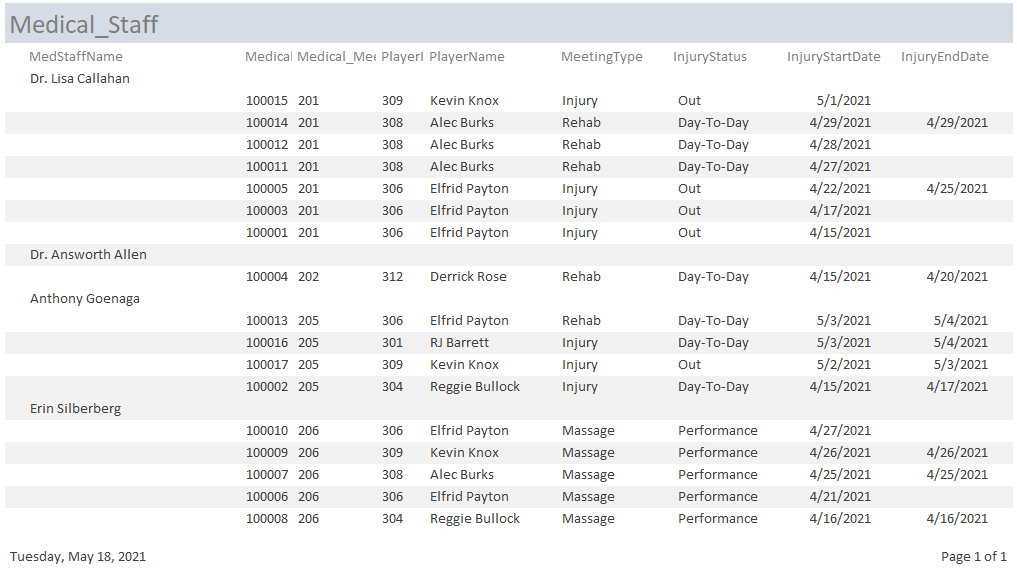
**Reports**

Game Statistics Report: The game statistic report shows the most recent game information with the nested individual statistics. This report is incredibly useful for the media and coaches after the game as well as referencing to review performance. It is common for players to be given the stat sheets when they are interviewed after the game. This report is sorted by most recent games at the top and by players usage number at the top.



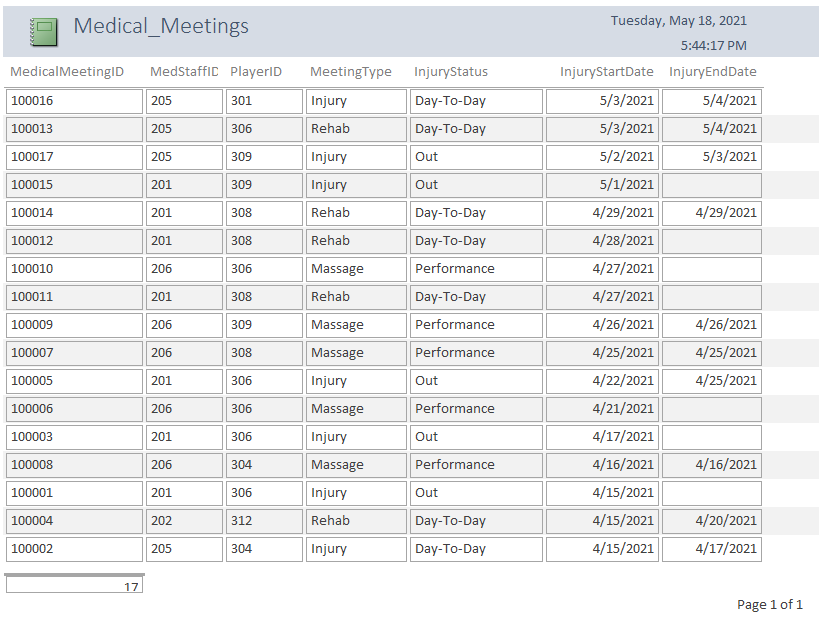
Medical Staff Report: The medical staff report shows the most recent medical meeting by each doctor. This information allows for an organized viewpoint of who each doctor is currently treating and on top of that allows for seeing the progression of players treatment’s by the doctor.

(shown on next page)

****

Medical Meeting Report:

This report shows the Medical Meetings sorted by most recent. This is a more general form for the coaches so they can constantly be updated on the medical meetings going on, and have enough of the recent information to see the previous meetings.



# **Conclusion**

The database created for the Knicks allows for optimal control over the data for the coaches and medical employees. In order to keep track of player performance and health it is important to have the information organized and readily available for usage. The queries, lookups, and reports, serve to help enable the team to be as informed as possible. The queries provide the manipulation available to give more advanced stats and select necessary sections of the data. The lookups allow for ease of reference for the coaches, medical team, and executives to see any necessary information. And the reports provide daily detailed data necessary to the team.