College Football Database

IST 659 Project

Jonah Witt

Section 72410

**Summary**

The goal of this project was to create a database to store data on NCAA Football teams. This is a personal database that will not be used by any member of the teams involved. Due to this, I am the only stakeholder in the project. This database will eventually serve as a basis to run predictive analysis. A large variety of data and statistics were collected on each team. Initially, data was collected from the 2017 Tennessee Volunteers. Next, the scope was expanded to every team in the Southeastern Conference for the 2017 Season. After this project is complete, the database will expand to include other conferences and seasons. However, this project only included teams from the SEC during the 2017 season. The data included categorical data on the teams, conferences, and players as well as quantitative statistics, by game, for each player.

In order for the database to be relevant, it had to be able to answer the following questions:

1. Who had the most rushing yards for the Tennessee Volunteers in 2017?
2. What team in the SEC had the kicker with the highest Field Goal Percentage in 2017?
3. What teams in the SEC won 6 or more games in 2017?
4. Who is the coach of the team that won the National Championship in 2017?
5. Did the Florida Gators beat the Georgia Bulldogs in 2017?

The database was created using SQL Server and then was connected to Microsoft Access for the user interface. I chose Microsoft Access because there was not a ton of time to create a GUI from scratch, and I was already familiar with the software. I input some of the data using SQL commands for demonstration purposes, but the remainder of the data was imported into access using the external data functionality in access to import the data from excel. I coded views in SQL to answer all of my data questions and they were imported as queries into Microsoft Access. These queries were then turned into reports for clean presentation and easy access.

**Conceptual Model**

Since this is a personal database for recreational analysis, I am the only stakeholder in the project. The business rules collected for the database are as follows:

* A Team can belong to only one Conference per Season.
* A Team can only be a National Champion if that team has a Win result in the National Championship Game type.
* A Team can only be a Conference Champion if that team has a Win result in the Conference Championship Game type.
* A Team can have multiple Coaches in a Season, but may only have one Coach per Game.
* A Player is assigned one Position and one Class per Season.
* A Player can play for only one Team per Season.
* A Coach can coach for only one Team per Season.
* A Game can only have one winner and one loser.
* The Team with the most points is the winner of the Game.
* There is only one National Champion per Division per Season.
* There is only one Conference Champion per Conference per Season.
* Only two teams may play in the same Bowl Game per Season.
* A team can only play a game against another Team once per Game Type per Season.

Again, none of the data collected is intended to be used by any of the teams included in the database. Data will be collected from multiple sources and input into the database. Below are some examples of unrefined data that will be input into the database.

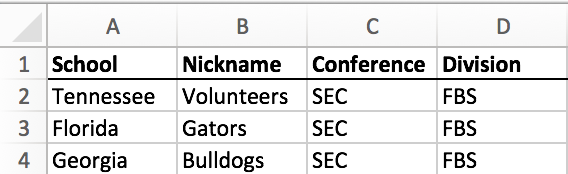


Figure 1: Sample Categorical Data on Teams

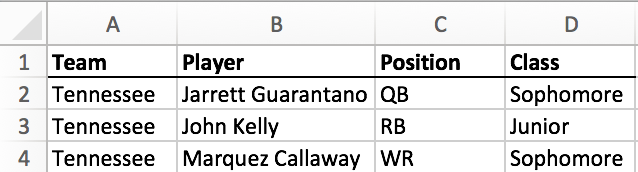


Figure 2: Sample Categorical Data on Players

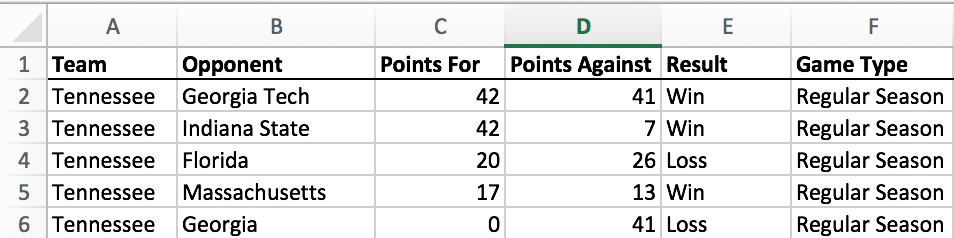


Figure 3: Sample Data from Games

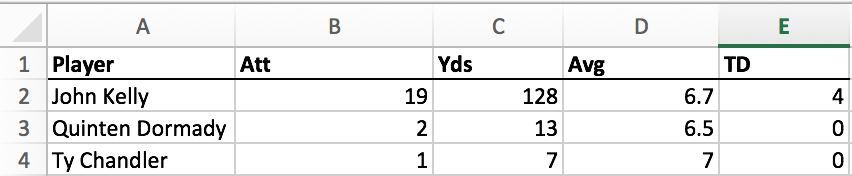


Figure 4: Sample Example of Rushing Statistics for a Game

Here is a glossary of the entities, attributes, and relationships for the database:

|  |  |
| --- | --- |
| **Entity** | **Attribute** |
| Division | Name [ru] |
| Conference | Name [ru] |
| Team | School [ru]  Nickname [r]  Wins [d]  Losses [d]  Conference Championships [d]  National Championships [d] |
| Season | Year [r]  Wins [d]  Losses [d]  Conference Championship [d]  National Championship [d]  Bowl Game [d] |
| Game | Opponent [r]  Points For [r]  Points Against [r]  Result [r]  Game Type [r] |
| Coach | Name [r]  Wins [d]  Losses [d]  First Season [r]  Final Season  Status [r]  National Championships [d]  Conference Championships [d] |
| Player | Name [r]  Position [r]  Class [r] |
| Stats | Pass Completions Pass Attempts Pass Completion Percentage Passing Yards Passing Yards Per Attempt Adjusted Passing Yards Per Attempt Passing Touchdowns Passing Interceptions Passing Efficiency Rating Rushing Attempts Rushing Yards Rushing Yards Per Attempt  Rushing Touchdowns Receptions Receiving Yards Receiving Yards Per Reception Receiving Touchdowns Plays From Scrimmage Yards From Scrimmage  Yards From Scrimmage Per Play Touchdowns From Scrimmage Solo Tackles Assisted Tackles Total Tackles Tackles For Loss Interceptions Interception Return Yards Interception Return Yards Per Interception Interception Return Touchdowns Passes Defended Fumbles Recovered Fumble Recovery Return Yards Fumble Recovery Return Touchdowns Fumbles Forced Kickoff Returns Kickoff Return Yards Kickoff Return Yards Per Return Kickoff Return Touchdowns Punt Returns  Punt Return Yards Punt Return Yards Per Return Punt Return Touchdowns Extra Points Made Extra Point Attempts Extra Point Percentage Field Goals Made Field Goal Attempts Field Goal Percentage Kicking Points Punts Punting Yards Punting Yards Per Punt |
| **Relationships** | |
| Each Division contains 1 or more Conference, each Conference belongs to 1 and only 1 Division  Each Conference has 1 or more Team, each Team belongs to zero or 1 Conference  Each Team participates in 1 or more Season, each Season has 1 or more Team  Each Season contains 1 or more Game, each Game belongs to 1 and only 1 Season  Each Game is coached by 1 and only 1 Coach, each Coach coaches 1 or more Game  Each Game contains 1 or more Stats, each Stats belongs to 1 and only 1 Game  Each Player has zero or more Stats, each Stats belongs to 1 and only 1 Player | |

Figure 5: Entity-Relationship Diagram Glossary

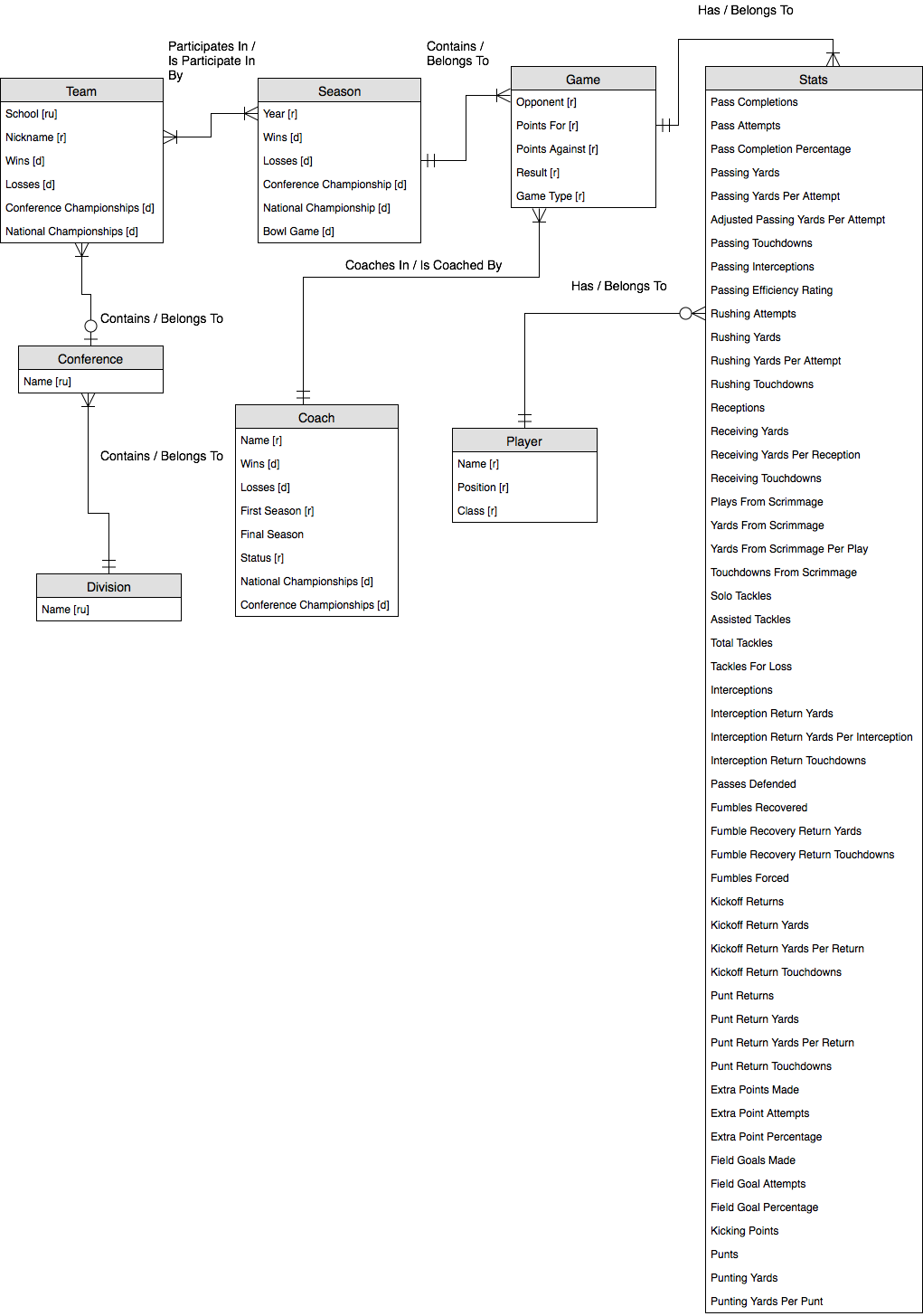


Figure 6: Entity-Relationship Diagram

**Normalized Logical Model**

The conceptual model was decomposed into relations in 3NF as follows:

* division (division id, name)
* conference (conference id, name, *division id*)
* team (team id, school, nickname, wins, losses, conference championships, national championships, *conference id*)
* season (season id, year, wins, losses, conference championship, national championship, bowl game)
* game (game id, opponent, points for, points against, result, game type, *coach id*, *season id*)
* coach (coach id, first name, last name, wins, losses, first season, final season, status, national championships, conference championships)
* player (player id, first name, last name, position, class)
* game\_stats (game\_stats id, pass completions, pass attempts, pass completion percentage, passing yards, passing yards per attempt, adjusted passing yards per attempt, passing touchdowns, passing interceptions, passing efficiency rating, rushing attempts, rushing yards, rushing yards per attempt , rushing touchdowns, receptions, receiving yards, receiving yards per reception, receiving touchdowns, plays from scrimmage, yards from scrimmage, yards from scrimmage per play, touchdowns from scrimmage, solo tackles, assisted tackles, total tackles, tackles for loss, interceptions, interception return yards, interception return yards per interception, interception return touchdowns, passes defended, fumbles recovered, fumble recovery return yards, fumble recovery return touchdowns, fumbles forced, kickoff returns, kickoff return yards, kickoff return yards per return, kickoff return touchdowns, punt returns, punt return yards, punt return yards per return, punt return touchdowns, extra points made, extra point attempts, extra point percentage, field goals made, field goal attempts, field goal percentage, kicking points, punts, punting yards, punting yards per punt, *game id*, *player id*)

Once the relations were normalized they were put into the logical model. In this step, several attributes were added and data types were selected. For each entity, a surrogate key was added as the primary key. Surrogate keys were chosen because some entities (players and coaches) could have the same name. Since surrogate keys were going to need to be used, I chose to streamline that across each entity to keep the database consistent. In addition, the name attributes for Player and Coach were decomposed into first names and last names in case they needed to be accessed or sorted individually.

The data type for each attribute was chosen on the following criteria:

* Name attributes were chosen as varchar to accommodate names of different character sizes. Size for each attribute was chosen based on the estimated amount of characters needed for each type of attribute.
* Year attributes were chosen as char with a character size 4 because no arithmetic will need to be done on the year attributes.
* All of the statistics that are counts were chosen as int type so that arithmetic can be done on them.
* All of the statistics that were averages or percentages were chosen as decimals to show the precise value of each entity.
* Wins and losses were chosen as int type so that they can be kept as counts.
* Championships for every entity except for Season were chosen as int so that they can be kept as counts.
* Championships for Season were chosen as varchar with size 3 to contain “Yes” or “No”
* Coach Status was chosen as varchar with size 7 because that is the largest size value it will hold (“Current”).
* Player position and class were chosen as char with size 2 because each value is an abbreviation with two letters.
* Points for and against were chosen as int so that arithmetic can be done on them.
* Result was chosen as varchar with size 4 since that is the largest size value it will hold (“Loss”).
* Game type was chosen as varchar with size 25 as an estimate of the largest value that it will hold.

Below is the completed database diagram:

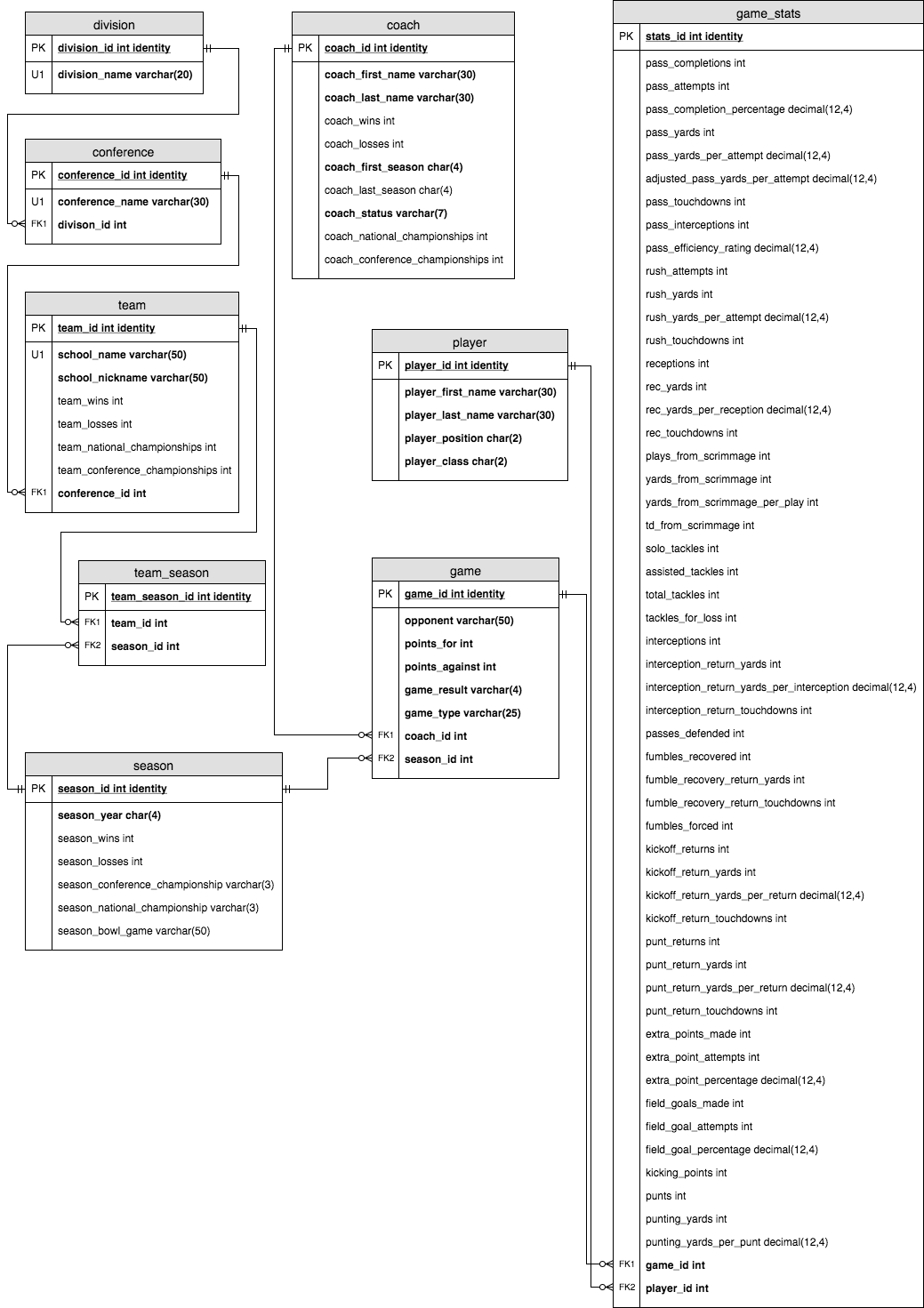


Figure 7: Database Diagram

**Physical Database Design**

Below is the Data Definition Language (DDL) used to in creating the database:

------RESET DATABASE------

--Select Database

USE CollegeFootball

--Remove all Tables

DROP TABLE IF EXISTS game\_stats, game, coach, player, team\_season, season, team, conference, division;

--Remove all Procedures

DROP PROCEDURE IF EXISTS addDivision, addConference, addTeam, addSeason, addCoach, fireCoach, addGame, addPlayer, addStats, updateResults, updateChampionships;

--Drop all Views

DROP VIEW IF EXISTS field\_goal\_totals, sec\_bowl\_eligible, field\_goal\_percentage, game\_results, national\_championship\_coaches, rushing\_yard\_totals

--Drop all Functions

DROP FUNCTION IF EXISTS getFieldGoalPercentage

------CREATE TABLES------

--Create Division Table

CREATE TABLE division(

division\_id INT IDENTITY NOT NULL,

division\_name VARCHAR(20) NOT NULL,

CONSTRAINT division\_PK PRIMARY KEY (division\_id));

--Create Conference Table

CREATE TABLE conference(

conference\_id INT IDENTITY NOT NULL,

conference\_name VARCHAR(30) NOT NULL,

division\_id INT NOT NULL,

CONSTRAINT conference\_PK PRIMARY KEY (conference\_id),

CONSTRAINT conference\_FK1 FOREIGN KEY (division\_id) REFERENCES division(division\_id));

--Create Team Table

CREATE TABLE team(

team\_id INT IDENTITY NOT NULL,

school\_name VARCHAR(50) NOT NULL,

school\_nickname VARCHAR(50) NOT NULL,

team\_wins INT,

team\_losses INT,

team\_national\_championships INT,

team\_conference\_championships INT,

conference\_id INT NOT NULL,

CONSTRAINT team\_PK PRIMARY KEY (team\_id),

CONSTRAINT team\_FK1 FOREIGN KEY (conference\_id) REFERENCES conference(conference\_id));

--Create Season Table

CREATE TABLE season(

season\_id INT IDENTITY NOT NULL,

season\_year CHAR(4) NOT NULL,

season\_wins INT,

season\_losses INT,

season\_conference\_championship VARCHAR(3),

season\_national\_championship VARCHAR(3),

season\_bowl\_game VARCHAR(50),

CONSTRAINT season\_PK PRIMARY KEY (season\_id));

--Create Team\_Season Table

CREATE TABLE team\_season(

team\_season\_id INT IDENTITY NOT NULL,

team\_id INT NOT NULL,

season\_id INT NOT NULL,

CONSTRAINT team\_season\_PK PRIMARY KEY (team\_season\_id),

CONSTRAINT team\_season\_FK1 FOREIGN KEY (team\_id) REFERENCES team(team\_id),

CONSTRAINT team\_season\_FK2 FOREIGN KEY (season\_id) REFERENCES season(season\_id));

--Create Coach Table

CREATE TABLE coach(

coach\_id INT IDENTITY NOT NULL,

coach\_first\_name VARCHAR(30) NOT NULL,

coach\_last\_name VARCHAR(30) NOT NULL,

coach\_wins INT,

coach\_losses INT,

coach\_first\_season CHAR(4) NOT NULL,

coach\_last\_season CHAR(4),

coach\_status VARCHAR(7) NOT NULL,

coach\_national\_championships INT,

coach\_conference\_championships INT,

CONSTRAINT coach\_PK PRIMARY KEY (coach\_id));

--Create Game Table

CREATE TABLE game(

game\_id INT IDENTITY NOT NULL,

opponent VARCHAR(50) NOT NULL,

points\_For INT NOT NULL,

points\_against INT NOT NULL,

game\_result VARCHAR(4) NOT NULL,

game\_type VARCHAR(25) NOT NULL,

coach\_id INT NOT NULL,

season\_id INT NOT NULL,

CONSTRAINT game\_PK PRIMARY KEY (game\_id),

CONSTRAINT game\_FK1 FOREIGN KEY (coach\_id) REFERENCES coach(coach\_id),

CONSTRAINT game\_FK2 FOREIGN KEY (season\_id) REFERENCES season(season\_id));

--Create Player Table

CREATE TABLE player(

player\_id INT IDENTITY NOT NULL,

player\_first\_name VARCHAR(30) NOT NULL,

player\_last\_name VARCHAR(30) NOT NULL,

player\_position CHAR(2) NOT NULL,

player\_class CHAR(2) NOT NULL,

CONSTRAINT player\_PK PRIMARY KEY (player\_id));

--Create Game\_Stats Table

CREATE TABLE game\_stats(

game\_stats\_id INT IDENTITY NOT NULL,

pass\_completions INT,

pass\_attempts INT,

pass\_completion\_percentage DECIMAL(12,4),

pass\_yards INT,

pass\_yards\_per\_attempt DECIMAL(12,4),

adjusted\_pass\_yards\_per\_attempt DECIMAL(12,4),

pass\_touchdowns INT,

pass\_interceptions INT,

pass\_efficiency\_rating DECIMAL(12,4),

rush\_attempts INT,

rush\_yards INT,

rush\_yards\_per\_attempt DECIMAL(12,4),

rush\_touchdowns INT,

receptions INT,

rec\_yards INT,

rec\_yards\_per\_reception DECIMAL(12,4),

rec\_touchdowns INT,

plays\_from\_scrimmage INT,

yards\_from\_scrimmage INT,

yards\_from\_scrimmage\_per\_play DECIMAL(12,4),

td\_from\_scrimmage INT,

solo\_tackles INT,

assisted\_tackles INT,

total\_tackles INT,

tackles\_for\_loss INT,

interceptions INT,

interception\_return\_yards INT,

interception\_return\_yards\_per\_interception DECIMAL(12,4),

interception\_return\_touchdowns INT,

passes\_defended INT,

fumbles\_recovered INT,

fumble\_recovery\_return\_yards INT,

fumble\_recovery\_return\_touchdowns INT,

fumbles\_forced INT,

kickoff\_returns INT,

kickoff\_return\_yards INT,

kickoff\_return\_yards\_per\_return DECIMAL(12,4),

kickoff\_return\_touchdowns INT,

punt\_returns INT,

punt\_return\_yards INT,

punt\_return\_yards\_per\_return DECIMAL(12,4),

punt\_return\_touchdowns INT,

extra\_points\_made INT,

extra\_point\_attempts INT,

extra\_point\_percentage DECIMAL(12,4),

field\_goals\_made INT,

field\_goal\_attempts INT,

field\_goal\_percentage DECIMAL(12,4),

kicking\_points INT,

punts INT,

punting\_yards INT,

punting\_yards\_per\_punt DECIMAL(12,4),

game\_id INT NOT NULL,

player\_id INT NOT NULL,

CONSTRAINT game\_stats\_PK PRIMARY KEY (game\_stats\_id),

CONSTRAINT game\_stats\_FK1 FOREIGN KEY (game\_id) REFERENCES game(game\_id),

CONSTRAINT game\_stats\_FK2 FOREIGN KEY (player\_id) REFERENCES player(player\_id));

------CREATE PROCEDURES------

--Add Division

GO

CREATE PROCEDURE addDivision(@division\_name varchar(20))

AS

BEGIN

--Create Division

INSERT INTO division(division\_name)

VALUES (@division\_name)

END

GO

--Add Conference

CREATE PROCEDURE addConference(@conference\_name varchar(30), @division\_name varchar(20))

AS

BEGIN

--Get Division ID

DECLARE @division\_id int

SELECT @division\_id = division\_id FROM division

WHERE @division\_name = division\_name

--Create Conference

INSERT INTO conference(conference\_name, division\_id)

VALUES (@conference\_name, @division\_id)

END

GO

--Add Team

CREATE PROCEDURE addTeam(@school\_name varchar(50), @school\_nickname varchar(50), @conference\_name varchar(30))

AS

BEGIN

--Get Conference ID

DECLARE @conference\_id int

SELECT @conference\_id = conference\_id FROM conference

WHERE @conference\_name =conference\_name

--Create Team

INSERT INTO team(school\_name, school\_nickname, conference\_id)

VALUES (@school\_name, @school\_nickname, @conference\_id)

END

GO

--Add Season

CREATE PROCEDURE addSeason(@school\_name varchar(50), @season\_year char(4))

AS

BEGIN

--Create Season

INSERT INTO season(season\_year)

VALUES (@season\_year)

--Get Season ID

DECLARE @season\_id int

SET @season\_id = @@IDENTITY

--Get Team ID

DECLARE @team\_id int

SELECT @team\_id = team\_id FROM team

WHERE school\_name = @school\_name

--Create Team\_Season

INSERT INTO team\_season(team\_id, season\_id)

VALUES (@team\_id, @season\_id)

END

GO

--Add Coach

CREATE PROCEDURE addCoach(@coach\_first\_name varchar(30), @coach\_last\_name varchar(30), @coach\_first\_season char(4))

AS

BEGIN

INSERT INTO coach(coach\_first\_name, coach\_last\_name, coach\_first\_season, coach\_status)

VALUES (@coach\_first\_name, @coach\_last\_name, @coach\_first\_season, 'Current')

END

GO

--Add Game

CREATE PROCEDURE addGame(@team varchar(50), @opponent varchar(50), @year char(4), @game\_type varchar(25), @coach\_first\_name varchar(30), @coach\_last\_name varchar(30), @points\_for int, @points\_against int)

AS

BEGIN

--Get Coach ID

DECLARE @coach\_id int

SELECT @coach\_id = coach\_id FROM coach WHERE coach\_first\_name = @coach\_first\_name AND coach\_last\_name = @coach\_last\_name

--Get Team ID

DECLARE @team\_id INT

SELECT @team\_id = team\_id FROM team WHERE school\_name = @team

--Get Season

DECLARE @season\_id int

--Get All Team\_Season for Team

--Create temporary table

CREATE TABLE one\_team(

one\_team\_id INT IDENTITY,

season\_id INT,

season\_year char(4))

--Add Season ID's to temporary table

INSERT INTO one\_team(season\_id)

SELECT season\_id FROM team\_season WHERE team\_id = (SELECT team\_id FROM team WHERE school\_name like '%' + @team + '%')

--Add Season Years to temporary table

UPDATE one\_team

SET season\_year = season.season\_year

FROM one\_team

JOIN season

ON season.season\_id = one\_team.season\_id

--Get Correct Year

SELECT @season\_id = season\_id FROM one\_team

WHERE season\_year = @year

--Delete temporary Table

DROP TABLE one\_team

--Get Result

DECLARE @game\_result varchar(4)

IF (@points\_for > @points\_against)

BEGIN

SET @game\_result = 'Win'

--Update Season

Update Season

SET season\_wins = season\_wins + 1

WHERE season\_id = @season\_id

--Update Team

Update Team

SET team\_wins = team\_wins + 1

WHERE team\_id = @team\_id

--Update Update Coach

UPDATE Coach

SET coach\_wins = coach\_wins + 1

WHERE coach\_id = @coach\_id

--Update National Championship

IF @game\_type = 'National Championship'

BEGIN

Update Season

SET season\_national\_championship = ‘Yes’

WHERE season\_id = @season\_id

--Update Team

Update Team

SET team\_national\_championships = team\_national\_championships + 1

WHERE team\_id = @team\_id

--Update Update Coach

UPDATE Coach

SET coach\_national\_championships = coach\_national\_championships + 1

WHERE coach\_id = @coach\_id

END

ELSE IF @game\_type = 'Conference Championship'

BEGIN

--Update Conference Championship

Update Season

SET season\_conference\_championship = ‘Yes’

WHERE season\_id = @season\_id

--Update Team

Update Team

SET team\_conference\_championships = team\_conference\_championships + 1

WHERE team\_id = @team\_id

--Update Update Coach

UPDATE Coach

SET coach\_conference\_championships = coach\_conference\_championships + 1

WHERE coach\_id = @coach\_id

END

END

ELSE

BEGIN

SET @game\_result = 'Loss'

--Update Season

Update Season

SET season\_losses = season\_losses + 1

WHERE season\_id = @season\_id

--Update Team

Update Team

SET team\_losses = team\_losses + 1

WHERE team\_id = @team\_id

--Update Update Coach

UPDATE Coach

SET coach\_losses = coach\_losses + 1

WHERE coach\_id = @coach\_id

END

--CREATE Game

INSERT INTO game(season\_id, opponent, points\_For, points\_against, game\_result, game\_type, coach\_id)

VALUES (@season\_id, @opponent, @points\_for, @points\_against, @game\_result, @game\_type, @coach\_id)

END

GO

--Add Player

CREATE PROCEDURE addPlayer(@player\_first\_name varchar(30), @player\_last\_name varchar(30), @player\_class char(2), @player\_position char(2))

AS

BEGIN

--CREATE Player

INSERT INTO player(player\_first\_name, player\_last\_name, player\_position, player\_class)

VALUES(@player\_first\_name, @player\_last\_name, @player\_class, @player\_position)

END

GO

--Add Stats

CREATE PROCEDURE addStats(@player\_first\_name varchar(30), @player\_last\_name varchar(30), @team varchar(50), @year CHAR(4), @opponent varchar(50), @pass\_completions INT,

@pass\_attempts INT,

@pass\_completion\_percentage DECIMAL(12,4),

@pass\_yards INT,

@pass\_yards\_per\_attempt DECIMAL(12,4),

@adjusted\_pass\_yards\_per\_attempt DECIMAL(12,4),

@pass\_touchdowns INT,

@pass\_interceptions INT,

@pass\_efficiency\_rating DECIMAL(12,4),

@rush\_attempts INT,

@rush\_yards INT,

@rush\_yards\_per\_attempt DECIMAL(12,4),

@rush\_touchdowns INT,

@receptions INT,

@rec\_yards INT,

@rec\_yards\_per\_reception DECIMAL(12,4),

@rec\_touchdowns INT,

@plays\_from\_scrimmage INT,

@yards\_from\_scrimmage INT,

@yards\_from\_scrimmage\_per\_play DECIMAL(12,4),

@td\_from\_scrimmage INT,

@solo\_tackles INT,

@assisted\_tackles INT,

@total\_tackles INT,

@tackles\_for\_loss INT,

@interceptions INT,

@interception\_return\_yards INT,

@interception\_return\_yards\_per\_interception DECIMAL(12,4),

@interception\_return\_touchdowns INT,

@passes\_defended INT,

@fumbles\_recovered INT,

@fumble\_recovery\_return\_yards INT,

@fumble\_recovery\_return\_touchdowns INT,

@fumbles\_forced INT,

@kickoff\_returns INT,

@kickoff\_return\_yards INT,

@kickoff\_return\_yards\_per\_return DECIMAL(12,4),

@kickoff\_return\_touchdowns INT,

@punt\_returns INT,

@punt\_return\_yards INT,

@punt\_return\_yards\_per\_return DECIMAL(12,4),

@punt\_return\_touchdowns INT,

@extra\_points\_made INT,

@extra\_point\_attempts INT,

@extra\_point\_percentage DECIMAL(12,4),

@field\_goals\_made INT,

@field\_goal\_attempts INT,

@field\_goal\_percentage DECIMAL(12,4),

@kicking\_points INT,

@punts INT,

@punting\_yards INT,

@punting\_yards\_per\_punt DECIMAL(12,4))

AS

BEGIN

--Get Player ID

Declare @player\_id int

SELECT @player\_id = player\_id FROM player WHERE player\_first\_name = @player\_first\_name AND player\_last\_name = @player\_last\_name

--Get Season

DECLARE @season\_id int

--Get All Team\_Season for Team

--Create temporary table

CREATE TABLE one\_team(

one\_team\_id INT IDENTITY,

season\_id INT,

season\_year char(4))

--Add Season ID's to temporary table

INSERT INTO one\_team(season\_id)

SELECT season\_id FROM team\_season WHERE team\_id = (SELECT team\_id FROM team WHERE school\_name like '%' + @team + '%')

--Add Season Years to temporary table

UPDATE one\_team

SET season\_year = season.season\_year

FROM one\_team

JOIN season

ON season.season\_id = one\_team.season\_id

--Get Correct Year

SELECT @season\_id = season\_id FROM one\_team

WHERE season\_year = @year

--Delete temporary Table

DROP TABLE one\_team

--Get Game ID

Declare @game\_id int

SELECT @game\_id = game\_id FROM game WHERE season\_id = @season\_id AND opponent = @opponent

--Create Stats

INSERT INTO game\_stats (pass\_completions,

pass\_attempts,

pass\_completion\_percentage,

pass\_yards,

pass\_yards\_per\_attempt,

adjusted\_pass\_yards\_per\_attempt,

pass\_touchdowns,

pass\_interceptions,

pass\_efficiency\_rating,

rush\_attempts,

rush\_yards,

rush\_yards\_per\_attempt,

rush\_touchdowns,

receptions,

rec\_yards,

rec\_yards\_per\_reception,

rec\_touchdowns,

plays\_from\_scrimmage,

yards\_from\_scrimmage,

yards\_from\_scrimmage\_per\_play,

td\_from\_scrimmage,

solo\_tackles,

assisted\_tackles,

total\_tackles,

tackles\_for\_loss,

interceptions,

interception\_return\_yards,

interception\_return\_yards\_per\_interception,

interception\_return\_touchdowns,

passes\_defended,

fumbles\_recovered,

fumble\_recovery\_return\_yards,

fumble\_recovery\_return\_touchdowns,

fumbles\_forced,

kickoff\_returns,

kickoff\_return\_yards,

kickoff\_return\_yards\_per\_return,

kickoff\_return\_touchdowns,

punt\_returns,

punt\_return\_yards,

punt\_return\_yards\_per\_return,

punt\_return\_touchdowns,

extra\_points\_made,

extra\_point\_attempts,

extra\_point\_percentage,

field\_goals\_made,

field\_goal\_attempts,

field\_goal\_percentage,

kicking\_points,

punts,

punting\_yards,

punting\_yards\_per\_punt,

game\_id,

player\_id)

VALUES (@pass\_completions,

@pass\_attempts,

@pass\_completion\_percentage,

@pass\_yards,

@pass\_yards\_per\_attempt,

@adjusted\_pass\_yards\_per\_attempt,

@pass\_touchdowns,

@pass\_interceptions,

@pass\_efficiency\_rating,

@rush\_attempts,

@rush\_yards,

@rush\_yards\_per\_attempt,

@rush\_touchdowns,

@receptions,

@rec\_yards,

@rec\_yards\_per\_reception,

@rec\_touchdowns,

@plays\_from\_scrimmage,

@yards\_from\_scrimmage,

@yards\_from\_scrimmage\_per\_play,

@td\_from\_scrimmage,

@solo\_tackles,

@assisted\_tackles,

@total\_tackles,

@tackles\_for\_loss,

@interceptions,

@interception\_return\_yards,

@interception\_return\_yards\_per\_interception,

@interception\_return\_touchdowns,

@passes\_defended,

@fumbles\_recovered,

@fumble\_recovery\_return\_yards,

@fumble\_recovery\_return\_touchdowns,

@fumbles\_forced,

@kickoff\_returns,

@kickoff\_return\_yards,

@kickoff\_return\_yards\_per\_return,

@kickoff\_return\_touchdowns,

@punt\_returns,

@punt\_return\_yards,

@punt\_return\_yards\_per\_return,

@punt\_return\_touchdowns,

@extra\_points\_made,

@extra\_point\_attempts,

@extra\_point\_percentage,

@field\_goals\_made,

@field\_goal\_attempts,

@field\_goal\_percentage,

@kicking\_points,

@punts,

@punting\_yards,

@punting\_yards\_per\_punt,

@game\_id,

@player\_id)

END

GO

--Fire Coach

CREATE PROCEDURE fireCoach(@coach\_first\_name varchar(30), @coach\_last\_name varchar(30), @coach\_last\_season char(4))

AS

BEGIN

--Get Coach ID

DECLARE @coach\_id int

SELECT @coach\_id = coach\_id FROM coach

WHERE coach\_first\_name = @coach\_first\_name AND coach\_last\_name = @coach\_last\_name

--Set Status to Former

UPDATE coach SET coach\_status = 'Former'

WHERE coach\_id = @coach\_id

--Set Last Season

UPDATE coach SET coach\_last\_season = @coach\_last\_season

WHERE coach\_id = @coach\_id

END

GO

------CREATE VIEWS------

--Get Field Goal Totals

CREATE VIEW field\_goal\_totals

AS SELECT season.season\_id, player\_first\_name, player\_last\_name, Sum(field\_goal\_attempts) AS sum\_field\_goal\_attempts, Sum(field\_goals\_made) AS sum\_field\_goals\_made

FROM ((game\_stats INNER JOIN player ON game\_stats.player\_id = player.player\_id) INNER JOIN game ON game\_stats.game\_id = game.game\_id) INNER JOIN ((team\_season INNER JOIN season ON team\_season.season\_id = season.season\_id) INNER JOIN team ON team\_season.team\_id = team.team\_id) ON game.season\_id = season.season\_id

WHERE (((game\_stats.field\_goal\_attempts)>0))

GROUP BY player.player\_first\_name, player.player\_last\_name, season.season\_id;

GO

--Get Bowl Eligible Teams

CREATE VIEW sec\_bowl\_eligible

AS SELECT school\_name, season\_wins, season\_losses

FROM (team\_season INNER JOIN season ON team\_season.season\_id = season.season\_id) INNER JOIN (team INNER JOIN conference ON team.conference\_id = conference.conference\_id) ON team\_season.team\_id = team.team\_id

WHERE season\_wins >= 6

GO

--Get Best Field Goal Percentage

CREATE VIEW field\_goal\_percentage

AS

SELECT player\_first\_name, player\_last\_name, CONVERT(DECIMAL(12,4), sum\_field\_goals\_made)/CONVERT(DECIMAL(12,4),sum\_field\_goal\_attempts) AS FieldGoalPercentage

FROM field\_goal\_totals;

GO

--Get Game Results

CREATE VIEW game\_results

AS

SELECT school\_name, opponent, game\_result, points\_For, points\_against, season\_year

FROM game INNER JOIN ((team\_season INNER JOIN season ON team\_season.season\_id = season.season\_id) INNER JOIN team ON team\_season.team\_id = team.team\_id) ON game.season\_id = season.season\_id

GO

--Get National Championship Coaches

CREATE VIEW national\_championship\_coaches

AS

SELECT season\_year, coach\_first\_name, coach\_last\_name, school\_name

FROM (team\_season INNER JOIN ((game INNER JOIN coach ON game.coach\_id = coach.coach\_id) INNER JOIN season ON game.season\_id = season.season\_id) ON team\_season.season\_id = season.season\_id) INNER JOIN team ON team\_season.team\_id = team.team\_id

WHERE game\_result='Win' AND game\_type='National Championship'

GO

--Get Rushing Yard Totals

CREATE VIEW rushing\_yard\_totals

AS

SELECT Sum(rush\_yards) AS sum\_rush\_yards, player\_first\_name, player\_last\_name, school\_name

FROM ((((game\_stats INNER JOIN player ON game\_stats.player\_id = player.player\_id) INNER JOIN game ON game\_stats.game\_id = game.game\_id) INNER JOIN season ON game.season\_id = season.season\_id) INNER JOIN team\_season ON season.season\_id = team\_season.season\_id) INNER JOIN team ON team\_season.team\_id = team.team\_id

GROUP BY player.player\_first\_name, player.player\_last\_name, team.school\_name

GO

------CREATE FUNCTIONS------

--Get Field Goal Percentage for Season

CREATE FUNCTION getFieldGoalPercentage(@player\_first\_name varchar(30), @player\_last\_name varchar(30), @team varchar(50), @season\_year char(4))

RETURNS DECIMAL(12,4)

AS

BEGIN

--get season

DECLARE @season\_id INT

SELECT @season\_id = season\_id FROM team\_season WHERE season\_id = (SELECT season\_id FROM season WHERE season\_year = @season\_year) AND team\_id = (SELECT team\_id FROM team WHERE school\_name = @team)

--get field goal percentage

DECLARE @field\_goal\_percentage DECIMAL(12,4)

SELECT @field\_goal\_percentage = sum\_field\_goals\_made/sum\_field\_goal\_attempts

FROM field\_goal\_totals

WHERE player\_first\_name = @player\_first\_name AND player\_last\_name = @player\_last\_name AND season\_id = @season\_id

RETURN @field\_goal\_percentage

END

GO

**Data Creation**

Below are ten examples of data being entered into the database using SQL insert statments:

--add a division

INSERT INTO division(division\_name)

VALUES ('FBS');

--add a conference

INSERT INTO conference(conference\_name, division\_id)

VALUES ('SEC', (SELECT division\_id FROM division WHERE division\_name = 'FBS'));

--add a team

INSERT INTO team(school\_name, school\_nickname, conference\_id)

VALUES ('Tennessee', 'Volunteers', (SELECT conference\_id FROM conference WHERE conference\_name = 'SEC'));

--add a season

INSERT INTO season(season\_year)

VALUES ('2017');

--add a season

INSERT INTO team\_season(team\_id, season\_id)

VALUES ((SELECT team\_id FROM team WHERE school\_name = 'Tennessee'), (SELECT season\_id FROM season WHERE season\_year = '2017'));

--add a coach

INSERT INTO coach(coach\_first\_name, coach\_last\_name, coach\_first\_season, coach\_status)

VALUES('Butch', 'Jones', '2013', 'Current')

--add a game

INSERT INTO game(opponent, points\_for, points\_against, game\_result, game\_type, coach\_id, season\_id)

VALUES ('Georgia Tech', 42, 41, 'Win', 'Regular Season', (SELECT coach\_id FROM coach WHERE coach\_first\_name = 'Butch' AND coach\_last\_name = 'Jones'), (SELECT season\_id FROM season WHERE season\_year = '2017'));

--add a player

INSERT INTO player(player\_first\_name, player\_last\_name, player\_position, player\_class)

VALUES ('Quinten', 'Dormady', 'QB', 'JR');

--add stats from a game

INSERT INTO game\_stats(pass\_completions, pass\_attempts, pass\_completion\_percentage, pass\_yards, pass\_yards\_per\_attempt, adjusted\_pass\_yards\_per\_attempt, pass\_touchdowns, pass\_interceptions, pass\_efficiency\_rating, player\_id, game\_id)

VALUES (20, 37, 54.1, 221, 6, 7.1, 2, 0, 122.1, (SELECT player\_id FROM player WHERE player\_first\_name = 'Quinten' AND player\_last\_name = 'Dormady'), (SELECT game\_id FROM game WHERE opponent = 'Georgia Tech'));

Below are several examples of data being entered into the database using stored procedures:

--add a division

EXEC addDivision('FBS')

--add a conference

EXEC addConference('SEC','FBS')

--add a team

EXEC addTeam('Tennessee', 'Volunteers', 'SEC')

--add a season

EXEC addSeason('Tennessee','2017')

--add a coach

EXEC addCoach('Butch', 'Jones', '2013')

--add a game

EXEC addGame('Tennessee', 'Georgia', '2017', 'Regular Season', 'Butch', 'Jones', 0, 41)

--add a player

EXEC addPlayer('John', 'Kelly', 'JR', 'RB')

**Data Manipulation**

Data manipulation in this database is done using stored procedures. The statement below, *fireCoach*, will update a coach’s record when he is fired or quits a team. The procedure changes the coach’s status from ‘Current’ to ‘Former’ and sets the coach’s last season to the season input by the user. This relates to the business rule that a team may only have one coach at a time, but may have multiple coaches in a season.

SQL Code:

--Fire Coach

CREATE PROCEDURE fireCoach(@coach\_first\_name varchar(30), @coach\_last\_name varchar(30), @coach\_last\_season char(4))

AS

BEGIN

--Get Coach ID

DECLARE @coach\_id int

SELECT @coach\_id = coach\_id FROM coach

WHERE coach\_first\_name = @coach\_first\_name AND coach\_last\_name = @coach\_last\_name

--Set Status to Former

UPDATE coach SET coach\_status = 'Former'

WHERE coach\_id = @coach\_id

--Set Last Season

UPDATE coach SET coach\_last\_season = @coach\_last\_season

WHERE coach\_id = @coach\_id

END

GO

The following code is from the *addGame* procedure. This first statement checks to see if the game was a win or a loss and sets the game\_result variable. An if statement is used to update the win and loss totals for each season, team, and coach. In addition, another if statement is used to check to see if the game was either a national championship or a conference championship. If either of these is true, the total championships is updated for each season, team and coach. This affirms our business rule that the team with the most points in a game is the winner and the team with the fewest points in a game is the loser. This also affirms the rule that the team is the conference champion when that team has a win in the conference championship game type, and it affirms the rule that the team is the national champion when that team has a win in the national championship game type.

DECLARE @game\_result varchar(4)

IF (@points\_for > @points\_against)

BEGIN

SET @game\_result = 'Win'

--Update Season

Update Season

SET season\_wins = season\_wins + 1

WHERE season\_id = @season\_id

--Update Team

Update Team

SET team\_wins = team\_wins + 1

WHERE team\_id = @team\_id

--Update Update Coach

UPDATE Coach

SET coach\_wins = coach\_wins + 1

WHERE coach\_id = @coach\_id

--Update National Championship

IF @game\_type = 'National Championship'

BEGIN

Update Season

SET season\_national\_championship = ‘Yes’

WHERE season\_id = @season\_id

--Update Team

Update Team

SET team\_national\_championships = team\_national\_championships + 1

WHERE team\_id = @team\_id

--Update Update Coach

UPDATE Coach

SET coach\_national\_championships = coach\_national\_championships + 1

WHERE coach\_id = @coach\_id

END

ELSE IF @game\_type = 'Conference Championship'

BEGIN

--Update Conference Championship

Update Season

SET season\_conference\_championship = ‘Yes’

WHERE season\_id = @season\_id

--Update Team

Update Team

SET team\_conference\_championships = team\_conference\_championships + 1

WHERE team\_id = @team\_id

--Update Update Coach

UPDATE Coach

SET coach\_conference\_championships = coach\_conference\_championships + 1

WHERE coach\_id = @coach\_id

END

END

ELSE

BEGIN

SET @game\_result = 'Loss'

--Update Season

Update Season

SET season\_losses = season\_losses + 1

WHERE season\_id = @season\_id

--Update Team

Update Team

SET team\_losses = team\_losses + 1

WHERE team\_id = @team\_id

--Update Update Coach

UPDATE Coach

SET coach\_losses = coach\_losses + 1

WHERE coach\_id = @coach\_id

END

**Answering Data Questions**

Below are the select statements that answer the data questions posed in the summary. These queries all reference views, but the views were left vague so that they could be used for multiple queries (different players, teams, etc).

Who had the most rushing yards for the Tennessee Volunteers in 2017?

Business Rule:

*A player can only play for one team per season.* All of John Kelly’s rushing yards were for the Tennessee Volunteers.

SQL Code:

SELECT TOP 1 sum\_rush\_yards, player\_first\_name, player\_last\_name, school\_name

FROM rushing\_yard\_totals

WHERE school\_name = 'Tennessee'

ORDER BY sum\_rush\_yards DESC

Results:



Figure 8: Results from Query for Data Question 1

What team in the SEC had the kicker with the highest Field Goal Percentage in 2017?

Business Rule:

*A player can only play for one team per season.* All of David Marvin’s field goals were for the Georgia Bulldogs.

SQL Code:

SELECT school\_name FROM team WHERE team\_id =

(SELECT team\_id FROM team\_season WHERE season\_id =

(SELECT season\_id FROM game WHERE game\_id =

(SELECT TOP 1 game\_id FROM game\_stats WHERE player\_id =

(SELECT player\_id FROM player WHERE player\_first\_name =

(SELECT TOP 1 player\_first\_name FROM field\_goal\_percentage ORDER BY FieldGoalPercentage

DESC) AND player\_last\_name = (SELECT TOP 1 player\_last\_name FROM

field\_goal\_percentage ORDER BY FieldGoalPercentage DESC )))))

Results:



Figure 9: Results from Query for Data Question 2

What teams in the SEC won 6 or more games in 2017?

Business Rule:

*A team can only belong to one conference per season.* All of these teams were a part of the SEC and only the SEC for the 2017 season.

SQL Code:

SELECT \*

FROM sec\_bowl\_eligible

ORDER BY season\_wins DESC

Results:

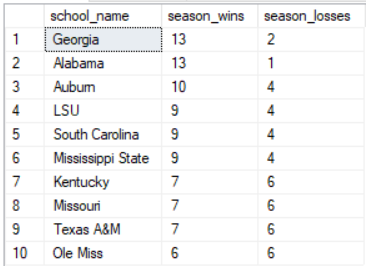


Figure 10: Results from Query for Data Question 3

Who is the coach of the team that won the National Championship in 2017?

Business Rule:

*A team can only have one coach per game.* Nick Saban was the only head coach of the Alabama Crimson Tide during the National Championship Game.

SQL Code:

SELECT \*

FROM national\_championship\_coaches

WHERE season\_year = '2017'

Results:



Figure 11: Results from Query for Data Question 4

Did the Florida Gators beat the Georgia Bulldogs in 2017?

Business Rule:

*A game can only have one winner and one loser.* The Florida Gators lost the game, and the Georgia Bulldogs won the game.

SQL Code:

SELECT \*

FROM game\_results

WHERE school\_name = 'Florida' AND opponent = 'Georgia'

Results:



Figure 12: Results from Query for Data Question 5

**Implementation**

The GUI for the college football database was created using Microsoft Access. I created a form that would serve as a “Main Menu” for the database. This main menu has links to game results, stats, conference standings, stats, season results, championship lookup, and coaching records. The menu also has a link for data entry.

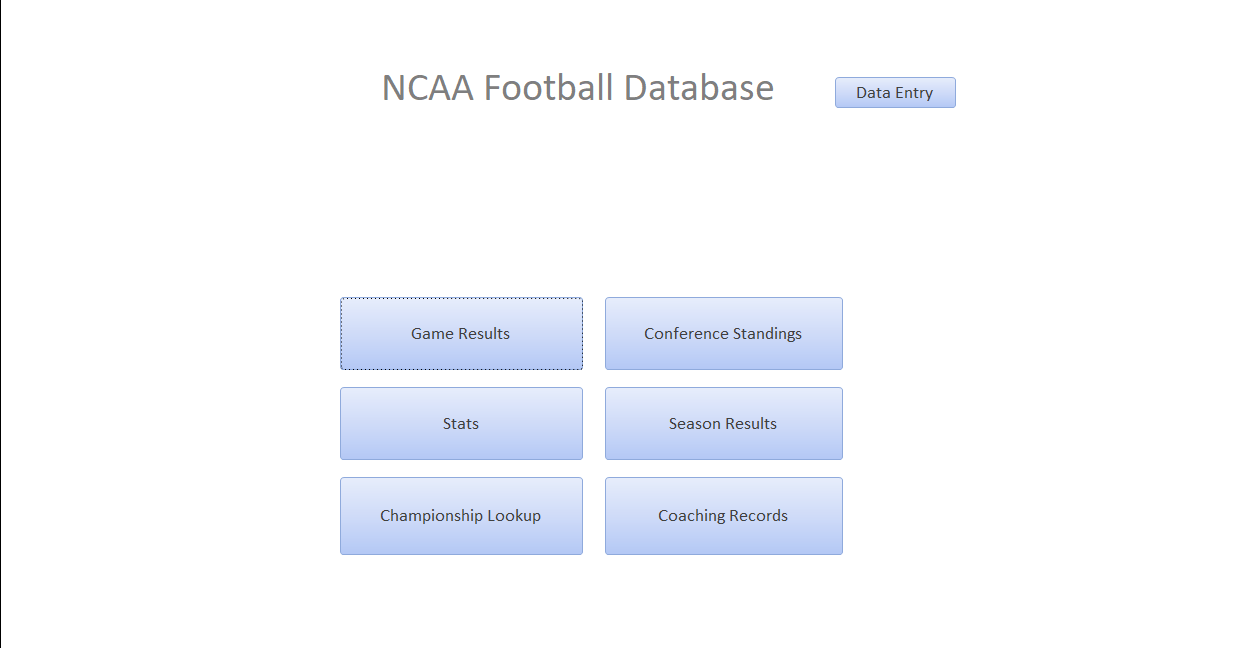


Figure 13: Main Menu of User Interface

The link to data entry takes the user to a data entry menu. From here the user can access a form to input data into each table in the database.

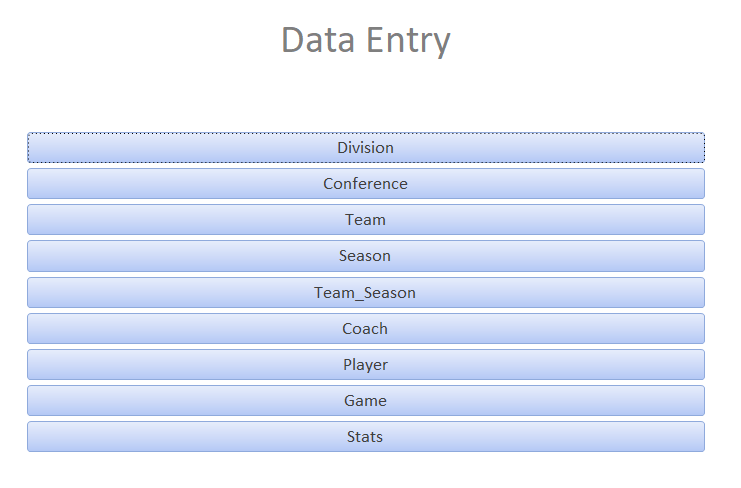


Figure 14: Data Entry Menu

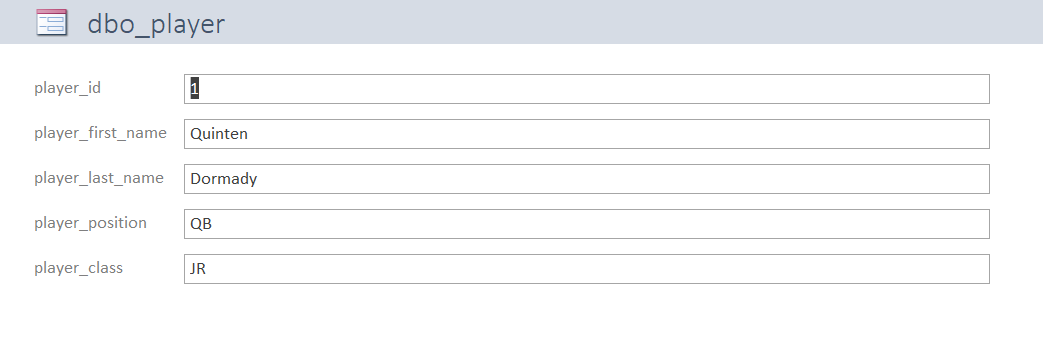


Figure 15: Example Data Entry Form

From the main menu, the user can also access a sub menu for each type of data. For this example, I have provided the sub menu for the championships data. From here the user has the choice to lookup data from conference championships or national championships.

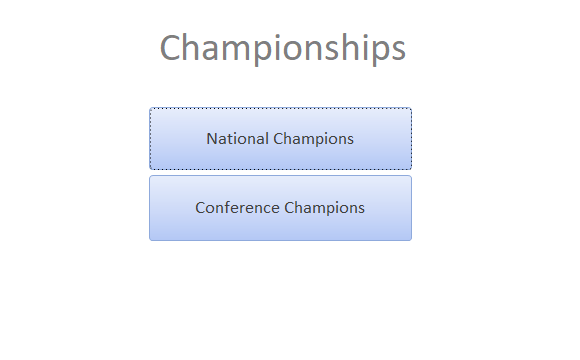


Figure 16: Example Sub Menu

The user will be prompted to enter the year of the data that he or she wishes to access.

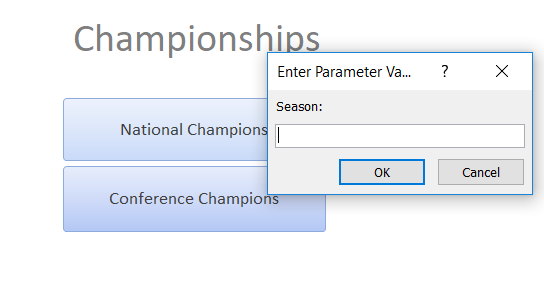


Figure 17: Example Prompt

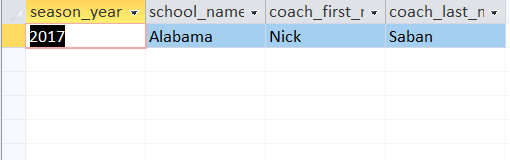


Figure 18: Example Results

Answering Data Questions in Microsoft Access:

From the views I coded in SQL Server, I created a query in access to answer all of my data questions. Below are the results of the access output for each of my data questions:

Who had the most rushing yards for the Tennessee Volunteers in 2017?

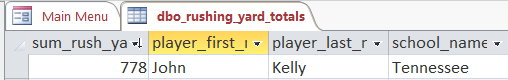


Figure 19: Output for Data Question 1

What team in the SEC had the kicker with the highest Field Goal Percentage in 2017?



Figure 20: Output for Data Question 2

What teams in the SEC won 6 or more games in 2017?

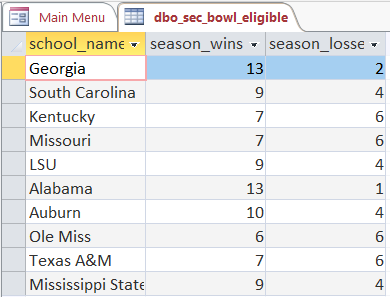


Figure 21: Output for Data Question 3

Who is the coach of the team that won the National Championship in 2017?

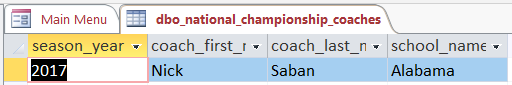


Figure 22: Output For Data Question 4

Did the Florida Gators beat the Georgia Bulldogs in 2017?

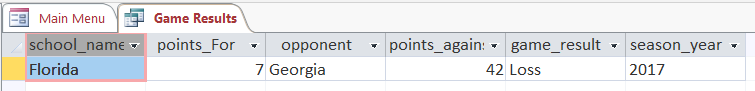


Figure 23: Output for Data Question 5

**Reflection**

At the start of this project, I expected that most of my time would be spent with the coding of the database creation. I found that this was the shortest step in the entire process. I severely underestimated how much time would be spent gathering the data and inserting it into the database. This was by far the longest step in my process. Part of this was because I designed my database without doing enough research on how the data would be gathered. I had to spend a lot of time manipulating my data to get it into the format that I needed to append to my database. Going forward, I will be sure to do thorough research on the data I will be gathering before I begin to create my conceptual model. This should save a lot of time in my next project.

This project has given me a new perspective on how I will use data as an information professional. If sufficient steps are taken in the creation and maintenance of the database, I will have clean data that can be analyzed and used in any capacity that I need. This will save valuable time that can be better spent adding value to my stakeholders. I now have the knowledge and abilities that I will need to accomplish this goal.