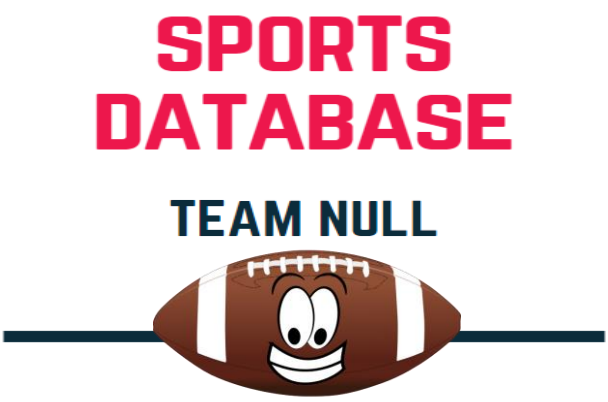
# Sports Management Database




*Project Design Document*

Team name: TEAM NULL

Meghana Pothugunta

Evelio Sosa

Sacheth Swaminathan

Siyuan Yang

Department of Computer Science and Engineering

Texas A&M University

February 17th, 2020

# Table of Contents

# Executive summary

A common way of storing data is through Excel. While this may be a sufficient idea for a single person, when it comes to a group of people needing data simultaneously then a more robust and sustainable option is needed. The way to efficiently solve this problem is by utilizing a database as a means of storing tables of data. In a database, key attributes can be stored, processed, and given relationships which offer convenient insight and capabilities to a user who needs the data. In this specific situation, the user is the TAMU Statistics department who will need data about Football games in order to create statistical inferences and questions for their students taking a statistics course. The objective of this project is to allow the user, whomever it may be (student, teacher, department, etc) to have easy access to college football data. They should also have the option of retrieving certain types of data or relationships in the data.

To keep things as optimal as possible, the database was designed such that almost each table has some sort of relationship to another table in the database. That way, if a specific query is generated (ie. which player has the yards in all of the Big Ten Conference) it will be easy to write an SQL command or stored procedure that accounts for the querying of such data. We will implement most of these stored procedures alongside our API's to ensure that the full stack of the program is taken care of. We will have multiple stored procedures that have basic functions and that will also work with the requests of other statistics and some that are very specific to ensure that no question is left unanswered. These different queries will be made evident in the GUI portion of the program, where most of the querying will be done.

On the completion of this program there are several things that are to be expected. A very robust GUI is to be created to ensure that the user can get the most out of the data. More importantly, proper data tables will be created also to ensure that no fact gets left out and that all of the data will relate to one another in some way. The final outcome of the database is to have the simplest design possible while being versatile and unconstrained under any workload.

# 1 Introduction

## 1.1 Purpose of the Project

Through the creation of this graphical user interface and database interaction, we were able to address the users' need for an efficient way of managing and querying data from the Texas A&M Football team. This is important because an organized location and easy method for accessing the data of interest needs to be used by A&M professors teaching the new sports statistics course which will then be accessed by the students taking the course.

## 1.2 Needs Statement

The department currently manages its data through an Excel sheet that is not sustainable or time-sensitive. In addition to the usability problem, professors teaching the class would not have the ability to spend the extra time needed trying to format, update, and add more. The professors also would not be able to effectively look at the data as a whole and analyze it (ie. looking at unused data, trends, etc).

The use of a user interface and database allows professors and students to obtain names of players, teams, conferences, individual games, hometowns, statistics, records or other items of interest more efficiently.

## 1.3 High-Level Entity Design

1. Player
    a. “Player” is necessary for the database because they are the participants of the game. The students will be able to gather more information about the players involved.
    b. **Elements:** ID, Age, Name, Position, College, GameID, InjuryID
2. Game
    a. “Game” is important because students may want to keep track of game details. This table will be more for the organization and have keys to other tables in the Database. It will be easy to track and pull up statistics about relationships between game elements using this table.
    b. **Elements:** GameID, Date, Winning Team, Losing Team, ScoreID, Stadium
3. Score
    a. “Score” will keep track of the results of a game between two colleges. This information will provide the user with a way to compare teams and predict future scores. The scoreboard will be dependent on what colleges are playing and the date they played.
    b. **Elements:** WinningScore, LosingScore, TouchDowns, GA
4. Injury Report
    a. “Injury Report” provides details regarding the players’ injuries, during which game, and how they got injured. The injury report will help sports leagues

and regional organizations identify key areas of risk in play, and develop new regulations to prevent them.

b. **Elements:** PlayerName, InjuryType, GameID

5. Location
    a. “Location” will provide information about the geographical position of the football stadium. It’s necessary because football games are held in the locations provided, and both players and audiences need this information to prepare for games.
    b. **Elements:** Stadium, City, State, ZipCode
6. Play
    a. “Play” is for if information about certain plays that were made in a game are points of interest for some descriptive statistics.
    b. **Elements:** GameID, CollegeID, CoachID, Play
7. College
    a. “College” will provide the name of a college that has a football team. The college name is important because it will be used to represent the team which includes many of the entries mentioned in this section. This data entry will be one of the most important in the database.
    b. **Elements:** Name, RankID, Logo, PlayersID, CoachID, Location, GamesWon, GamesLost, TotalScore, DivisionID
8. Rank
    a. “Rank” will indicate the position of a team compared to the others based on the match history for a given season. The ranking of a team provides an

easier way to see what teams are performing the best and worst throughout the season.

b. **Elements:** Rank, College, Point, ConferenceRank

9. Conference
    a. “Conference” is a manner of splitting teams from a particular league into two groups that primarily play against each other. Teams play against teams from their conference more often than teams from the other conference. The two conference champions play against each other in the final game.
    b. **Elements:** Division, DivisionID, College, RankinConference
10. Coach
    a. “Coach” is important to keep track of who coached what team during what games. It can reveal the impact the coach has on the condition of the team.
    b. **Elements:** CoachID, Age, Name, Team

## 1.4 Low-Level Entity Design

1. Player
    - **Elements:** ID, Age, Name, Position, College, GameID, InjuryID
    - “Player” gets information from “Injury Report” and gives information to “Injury Report”, “College”, and “Game”. The information provided by “Player” will allow a more in-depth analysis of how a college team is structured and could be part of the prediction of what colleges will perform the best throughout a given season. The “Player” data has been constantly updated since age, position, injuries, and college can vary throughout the career of the football player. If information like college is updated in “Player” then the entry “College” must also update its array of players. The risk of these changes not occurring can result in inconsistent information provided to the user using the database.
2. Game
    - **Elements:** GameID, Date, Winning Team, Losing Team, ScoreID, Stadium
    - “Game” gets information from “Score”, “Location”, “Players”, and “Injury Report” and gives information to “Play”. “Game” will provide important statistics about a given game that allows for the ranking of colleges. The information given by “Game” will not be updated after being put into the database, thus the only risk would occur in the input of the data.
3. Score
    - **Elements:** WinningScore, LosingScore, TouchDowns, GameID

- “Score” does not get information from any of the entities and gives information to “Game”. The information provided by “Score” will allow for comparisons between colleges based on the performance of each team in a game and the results. The data within “Score” is based on which colleges were playing for a given game. The colleges and games have unique IDs associated with them, therefore, any possible risk will come from the possibility of different games and colleges having the same ID which is very unlikely to happen.

4. Injury Report
    - **Elements:** PlayerName, InjuryType, GameID
    - “Injury Report” gets information from “Player” and “Game”, and gives information back to “Player”. The information provided by the “Injury Report” will be used to provide a better understanding of the performance of a player throughout a season. The injury report has its own InjuryType, that’s to say, as long as the input Game ID and player names are correct, there will be no risks.
5. Location
    - **Elements:** Stadium, City, State, ZipCode
    - “Location” does not get information from any of the entities and gives information to “Game” and “College”. The information provided by “Location” could be used to show the impact of the location of where a game is played to the score results. The “location” will mostly stay the same unless college teams change the place where they play their home games. As a result, the

only risk would come from the input of the data such as city, state, and ZipCode.

6. Play
    - **Elements:** GameID, CollegeID, CoachID, Play
    - “Play” gets information from “Game”, “College”, and “Coach” and does not give any information. The information provided by “Play” can be used to better understand the performance of the overall team, since there are times where the end result does not reflect who was the better team in the game. The possible risk will come from different games, colleges and coaches having the same ID.
7. College
    - **Elements:** Name, RankID, Logo, PlayersID, CoachID, Location, GamesWon, GamesLost, TotalScore, DivisionID
    - “College” gets information from “Location”, “Coach”, “Conference”, “Player” and “Rank”, and gives information back to “Play”. The information provided by "College" will provide the overall overview of a college team and their performance throughout a given season. It will be the main component in the statistical analysis of a football league. The entry “College” will interact with many of the other entries in the database. Thus, the possible risks will come from requesting information with the wrong id. Also, if any information is updated in other entries, it would also have to be updated in "College".
8. Rank
    - **Elements:** Rank, College, Point, ConferenceRank

- “Rank” gets information from “Conference”, and gives information to “College”. The entry “Rank” will keep the ranking of the teams based on their points throughout the season. The information provided by “Rank” is important because it allows the user to access the best and worst teams without the analysis of other information. It is an easier way to see how a given team is performing in a given time in the season. The information in “Rank” is based on the points of all teams, thus any risks would come from the input of data into the database.

9. Conference
    - **Elements:** Division, DivisionID, College, RankinConference
    - “Conference” does not get information from any of the entities and gives information to “College” and “Rank”.
    - The information provided by “Conference” is important because it lets the person using the database know that even though a set of college teams belong to the same league, they will only play with teams in their conference for the entire season until playoffs. The information can be to categorize teams based on these different conference groups. The possible risk would come from the input of the data from division, college, and rankin conference.
10. Coach
    - **Elements:** CoachID, Age, Name, Team
    - “Coach” gets information from “College”, and gives information to “Play” and “College”. The information provided by “Coach” can be used to analyze the

impact of a coach on the performance of a college team. Also, with the use analysis of plays, the user can conclude what the style of play is for a given coach. The possible risk associated with “Coach” is the input of the data such as age, name, and team. In addition, coaches could change teams at any given time so there could be inconsistency if the information is not updated correctly.

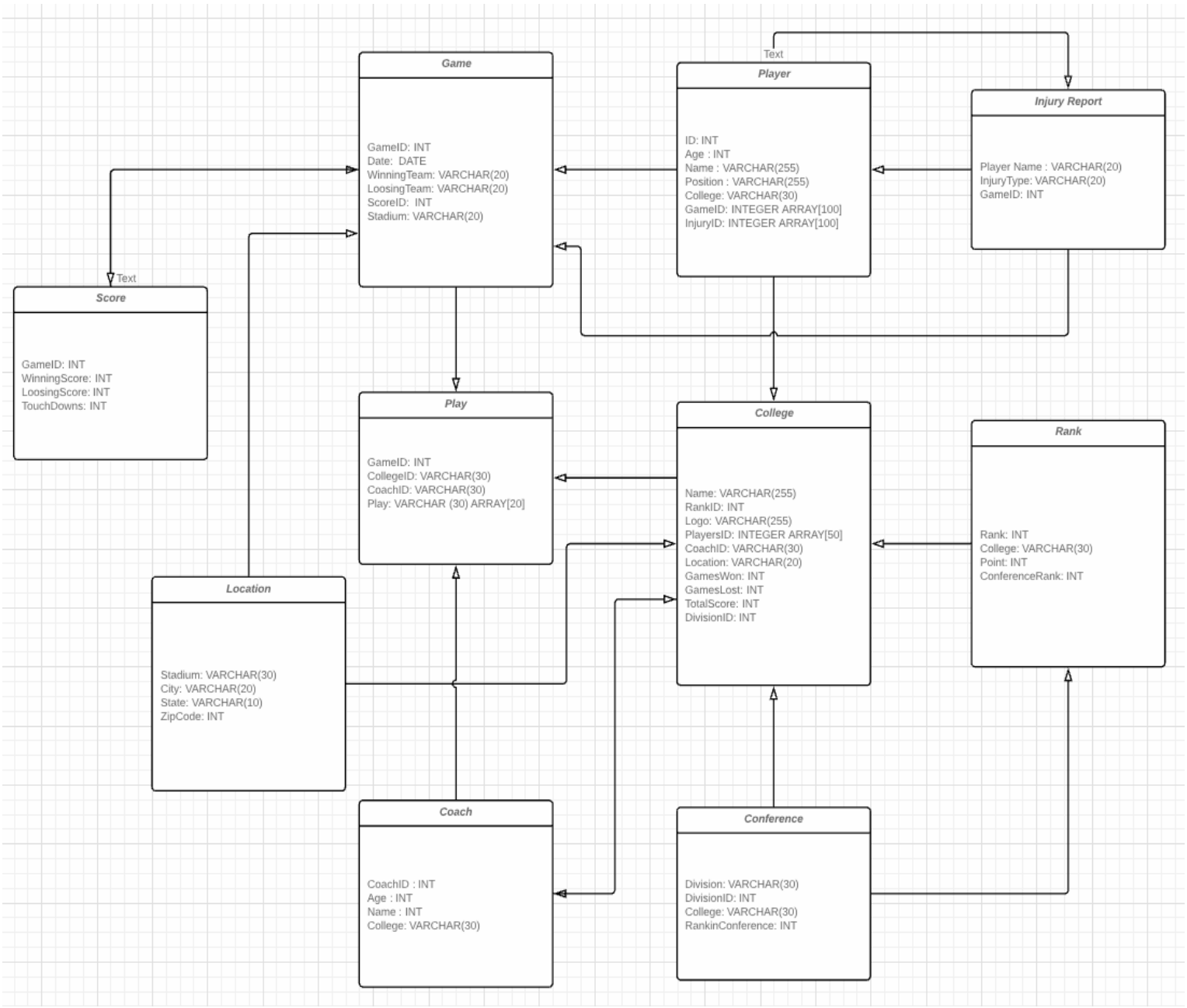



Diagram 1. UML of tables in the database

Description: Tables are represented in rectangular boxes. Each attribute is a column in the table, and each interaction is represented by a directional arrow.

## 1.5 Expectations

Benefits

As opposed to managing data on an Excel spreadsheet, data redundancy would not be much of a problem since the tables in the database are organized in such a way that data does not get repeated. Such redundancy is controlled by application programming and kept to a minimum by introducing as little as possible when designing the database.

Another advantage of a database management system is how it allows for data independence. Each data item is stored in only one place in the database, and the system data descriptions or data describing are separated from the application programs. Thus, changes to the data structure are handled by the database management system instead of embedded in the program itself.

Lastly, the database system provides backup and recovery methods that allow us to protect the data from loss. If a hard drive fails and the database stored on the hard drive is not accessible, the database system has a separate process from that of a network backup for backing up and recovering data. The only way to recover the database is from a backup.

Assumptions

One assumption that we are making is that the data is going to be present to us and that it is acceptable to give out to other parties.

Another assumption is that we are getting accurate and unbiased data from a credible source.

Risks

A potential risk is that the data could be leaked or corrupted if not managed via some secure interface.

In addition to this, the risk of data not being accurate or up to date may create inconsistencies in the backend of the project. If the program is multi-user, then the data has to be read-only or read/write depending on the level of access each user has.

Issues

Connecting to TAMU network could be an issue, if a member of the development team was not connected or if access to the database from an IP that was outside TAMU's network was made then that data could be blocked.

Another issue could be that relationships among tables could be redundant leading to the possibility of a slower runtime when accessing data from the database or doing complex queries.