Matthew Olivo

CPSC 408 Final Project Report

**Philadelphia Eagles Database**

**Introduction to the problem**

The goal of my project was to assemble a complete database of every single player, general manager, owner, head coach, and team season in the history of the Philadelphia Eagles; a professional football team that plays in the National Football League. While I did not completely succeed in accomplishing this task, I came up with a solution that is either better than or equivalent to any solution that can be found online. This gives my project validity in possibly posting it as an online resource, however that would require additional work/time as I have not integrated an online version of this database yet. The motivation behind this project was rabid fandom of not only the Philadelphia Eagles, but also the historic NFL as a whole. It’s been my favorite sports league since I was a kid, and one of my favorite things to discuss with friends and family as a whole. As I can attest to firsthand, if a perfect version of my database existed out there and had a touched up user interface, I would look through it for hours on end. The querying and seeing which players played for which management and coaches is addicting and can teach the average fan so much about the history of their favorite team.

**Solution’s elements**

Files and aspects

My solution is comprised of three total files. The first of which is a mySQL file that sets up my entire database, the tables within it, the variables in each of those tables, and the restrictions for each field. Next, I created a simple Python script that simply reads in the data in my .csv files to populate my mySQL database with all of the player, coach, general manager, owner, and team season data. This file is only run when the database’s tables have been created but are completely empty. If this script is run while the database’s tables aren’t created or already are populated, the script will return an error. The mySQL database I created is named EaglesDB and has all of the tables mentioned above. The schema and relationships concerning these tables will be discussed later. Final my last Python script contains all of my database’s operations. This script contains all of my CRUD operations with advance views, whole database views, create new records, update current records, and deleting records. This file is the most complex out of all the other project files and it features 7 different functions each providing a pivotal function to the project. The first 2 main menu options direct you to functions which perform various simple and complex database views. The third and fourth options are the create new and delete operations. The delete is a hard delete and requires the primary key of the desired entity to be deleted. Finally, the last 2 operations are updating a desired record, and exporting a report to a .csv file. For updating an entities’ records, one must specify the desired record’s primary key just as if they were to delete the record.

Unique or novel database aspects

My database features 6 different tables each of which involve a different type of personnel that works for an NFL football team. My database allows for views of not only the entirety of each table, but also allows you to select a coach, general manager, owner, or team season and view all other entities that served during that select record’s time frame. This is done by using the YearFrom and YearTo attributes in each of the personnel tables, and joining with them each other, and when trying to view the team season table, those two fields join with the primary key Year. I also scripted the ability to view, sort or limit certain table’s views in order to view only the desired outcomes and reduce clutter. The tabulate library was recommended to me by Professor Ali after my presentation day, and I was able to effortlessly implement it in under an hour. Tabulate instantly made my user interface much more readable and organized.

Design/Technology Decisions

Initially I wanted to create a more visually appealing database than the options that were already out there and make it feel more interactive. I initially attempted to create this database with a Unity user interface and use C# scripting to provide the database operations. Unfortunately, I wasn’t able to easily implement mySQL in unity and found out it would require a lot of plumbing. The amount of time I spent trying to get Unity to operate with mySQl could have been better spent getting my database able to be accessed on the web. As far as the database design, I felt it would be good to separate each of the core functions into different methods. The basic view function and the more complex view function design could have been configured differently, but it was a tough decision breaking all of the different options and parameters into different loops.

Similar work and projects

As far as competition for my project, when looking at alternatives that can be found already online, there are really 2 that are nearly as intensive as mine. The first of which is FootballDatabase, which has every player cataloged since 1936 with stats and years active. The main thing that separates my database from this online resource is I am covering all different types of people and positions involved with the entire organization, FootballDatabase only covers players. The other comparable database that I found online is an open-source platform named profootballreference. I actually used the tables found on this website to populate my database. This is a great website to look at the stats for each player, coach, and team season, while still being able to view the owners and general managers for each season. My database distinguishes itself from profootballreference because my viewing capabilities is more comprehensive and informative than this website. Apart from these two websites, there’s nothing that comes even close to the comprehensiveness nor usefulness of my database.

Final Results and Schema

What my finished product entails is an extremely comprehensive list of nearly all players, head coaches, owners, general managers, and team seasons. Like professor Ali mentioned, had I worked more diligently/had more time a major improvement to the project would have been implementing it in an online capacity. Provided I have more time to work on this project this summer, this may be a direction I will take the project on my own time. This would be the last step needed in order to make this a database that someone would use to look through a team’s history and settle debates as to who was part of which teams in a franchise’s history. With the advanced views allowing you to limit and sort certain records, it makes it a much more seamless solution than others already online. Below I have my final project schema. I feel as though I should reiterate that the YearFrom and YearTo variables are foreign keys pointing to the SeasonResult’s primary key Year. When I attempted to set the foreign key constraint on those variables, I was met with an array of errors which I choose to simply avoid.

`A screenshot of a computer

Description automatically generated with low confidence