CSCE 4523 Database Management Systems

Project

Cassie Smith Amber Morton

csce.uark.edu/~cps019/Project/website/home.html

csce.uark.edu/~aemorton/Project/website/home.html

**Objectives**

The objective of this homework is to make a website that links to a database. The website functions as the front end that is seen by the users and takes in the inputs from the user. The website links to the database using a python file which allows for the commands that run the database to be called without the user needing to call them.

**Approach**

For the project we used python, php, and html to complete the assignment. We implemented this using turing and mySQL on turing where it is also hosted. The website was designed like a form, which keeps the information centered on the page and makes it easier for the user to enter information and understand the instructions of the website. The home page is an html file, that links to the subpages which are php files that call the python code.

**Database Design**

We began by using the table outlines on the instructions page. That gave us three tables for players, teams, and games that link using the team IDs as foreign keys. The attributes for the player table are the player’s name, team ID, player ID, and position. For the team table the attributes are the team ID, location, nickname of the team, conference, and division. The final table is the game table that consists of the attributes game ID, team IDs of the two teams that played, the date of the game, and the scores of both teams.

The main constraints are when adding players and games to the tables, since they both require existing teams, if the user tries to add a player into a team that does not exist within the database, then the user should be unable to add the information and be prompted to input a valid team. We also added constraints to the primary keys for the player table since there are no attributes that are specifically unique, since names can repeat, and numbers can repeat as well.

**Results**

When opening the website, users are greeted with a home page that contains buttons that lead to the rest of the options that interact with the database and take in inputs.

A screenshot of a sports management system

Description automatically generated

The first function within the website creates the databases and inserts the initial data into the tables. The button leads to a website that creates the table and calls the insert functions; however, every function calls the open database function. There is not information for the user to input since it is located within the python files so that the user has access to the updated database.

A screen shot of a computer

Description automatically generated

Once the tables are created, the functions that are used by the user can be called and implemented. The first function adds a game to the game table, taking in information for all the columns of the game table: both team IDs, the game ID, the date, and the scores for both teams.

The add player function is a similar function to the add game function only it adds player data to the player table. This table takes in the name, player ID, team ID, and position of the player.

The function that displays the games played by a specific team takes in the team ID and searches the games table for that team ID in both team ID columns to get every game played by the input team. It displays the location of the team, the team nickname, the opponents location and nickname, the date, and the score which specifies who won or lost.

A screenshot of a login form

Description automatically generated

To get the players that play a specific position, the function takes in the position that the user is searching for and outputs players names for all players in the players table with that position.

A screenshot of a computer screen

Description automatically generated

Viewing all players on a team requires the user to input the team ID, then outputs the players with that team ID, which includes their name and position.

A screenshot of a computer login

Description automatically generated

The function that outputs the teams that played in a specific conference takes in the conference as a parameter and outputs the teams alphabetically based on nickname along with the number of wins overall and the number of wins within the conference.

A screenshot of a computer login form

Description automatically generated

The last required option is to output the results from games on a given date. This takes in a date and outputs the teams that played by their nickname and team name, the location, and the scores.

A screenshot of a computer

Description automatically generated

**Discussion**

For this assignment, we split the assignment into creating the database and creating the webpage. We began by creating the database and inserting initial data. From there we created the webpage for the home screen that contains buttons that take the user to the different webpages to enter the data for each of the options. Once the main page opened and displayed the different functions the user uses to interact with the database, the functionality of the options was added. First, we added the functions within the python file that include the select and update statements to read from the database and add new players and games. Next, we added the php files into the website to give the user to ability to input information into the commands as arguments. Once the php files are displaying properly, we then linked the php file to the python file and worked to get the php file to call the python file correctly and output the results of the functions.

The main issues that we had with the website is when the user was asked to enter multiple values to insert into the tables. This issue made it so that the pages to add games and add players did not display and would give user’s an error. Another issue that arose was with displaying the results after an insert or select function. This issue made it challenging to test and make sure that the python functions were running properly and made sure that the results being printed were correct.

For this project, Cassie worked on the website, creating, and editing the html and php files to link together and to the python code. The website consisted mostly of php files, with the home screen being an html file that did not need to connect to a python file to run commands with the database. Amber worked on the python files. She made sure that each of the select files worked individually within the terminal before connecting it with the website.