Michelle Yoon

CPSC 408

Final Project

May 23, 2021

**Panther Professors**

**Introduction**

Being that I was a transfer student, it was hard to know which professors to take for my classes my first semester at Chapman University. Rate My Professor was a tool I used to help with that. Of course, it was not something I heavily relied on because Rate My Professor is not fact based, it is opinion based. However, generally you kind of get the consensus of what the professor is like in that class from the student reviews. A problem that I had with Rate My Professor was that there was not one specifically tailored towards the Fowler School of Engineering at Chapman University. For a solution, I wanted to create an application that would accommodate the Fowler School of Engineering and help any students who would like to know more about the professor and the student experiences in those classes. This would be a tool that new incoming students or transfer students can use. It would be updated ever year to make sure students are getting updated information from the application. The application was built in PyCharm and the languages I use were Python and MySQL. I used command line visuals for the entire application.

**Normalized Tables**

For the normalize tables for this application, I had a Professor table, Subject table, Course table, User table, Rate table.

***Professor Table***

The professor table that consists of a primary key for the professor id, the professor’s first name, and last name.

***Subject Table***

The subject table consists of a primary key for the subject id and the subject name.

***Course Table***

The course table consist of a primary key for the course id and the course name. I had two foreign keys, one was for the professor id and the second was the subject id.

***User Table***

The user table consists of a primary key for user id, user email, and deleted user. If the user wants to delete the account, we will use deleted user to indicate that.

***Rate Table***

The rate table consists of the primary key for the rate id, rate number, and user comment. I had four foreign keys, one was for the professor id, the second was the subject id, the third was the course id, and the fourth was the user id.

**Indexes**

I created indexes for professor’s first name, professor’s last name, subject name, course name, rate number, and user comment.

**Database View**

I created view for professor information that includes professor’s first name, and last name, subject name, user email, rate number, user comment, and deleted user. This view was very helpful in this application because there were many tables I needed to join in order to get the correct information for the main menu and subset menus.

**Schema Diagram**

A screenshot of a computer

Description automatically generated with medium confidence

**Menu**

For the main menu, I included display records, insert, update, delete, search, generate reports, and quit. In addition, I included subset menu for display records, update, search and generate report. All the subset menus had to option to go back to the main menu or to quit the at any time during the application.

**Print/display records from database/tables**

In the display records menu, the user had the option to display the entire record which joined all the normalized tables together to display ever attribute. In addition, the user had the option to display the user information which joined the professor, course, and user tables to view the attributes of the user id, user email, the professor’s first and last name, course, rate number, and user comment that the user left for that professor.

**Create a new record**

In the insert menu, this is where the user would create a review for the course. The information the user inputted were user email, professor’s first and last name, the subject, the course name that the professor was teaching, the rate of the course, and the review or comment.

**Update records**

In the update menu, the user had the option to update the user email which included the user to type their user id and their updated email. In addition, the user had the option to update the user comment which included the user to type their user id and their updated comment.

**Delete a record**

In the delete menu, the user is allowed of delete their account. This would include the user to enter their user id.

**Query for data/results with various parameters/filters**

In the search menu, the user had the option to search the records by subject which included all the courses within that subject. In addition, the user had the option to search the record by rates. The range of rates were 1-5. It would display all the course that the user chooses for the input of the rate. Here I used a subquery, I joined the professor, course, user, and rate tables and in my where case I selected my attribute from the rate table to find the rate that the user was looking for.

**Generate reports that can be exported (excel or csv file)**

In the generate report menu, the user had the option to generate reports for deleted users and average rate of courses or import all reports in csv files. Here I used an aggregation and a group by clause on find the average rate on course and group them by course name.

**Results**

**Display all records**

**Text

Description automatically generated**

**Display user information**

**A picture containing text, outdoor, plaque

Description automatically generated**

**Search by subject: ex) Computer Science**

**A picture containing text, plaque

Description automatically generated**

**Search by rate: ex) rate: 3**

**A screen shot of a computer

Description automatically generated with low confidence**

**Generate report: ex) Average rate of courses**

**Graphical user interface

Description automatically generated**

**Generated report: ex: Deleted users**

**Text

Description automatically generated**