

Final Project Part B

Submitted to : Submitted by:

Ahmad Alhamed Tutku Ramazanoglu C0768385

While the project had been developing SqlServer was used for database and python was used as a development environment. Database is designed according to figure 1 using draw.io.

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 1 Database design

After database design, pypyodbc library is imported for connecting python and SqlServer each other. After that, database and tables create in python according to figure 2. First off all, database connection is defined. Using create table command and values is added to table. Also using try except block, table is checked which is exist or not. If it does not exist table is created and added values in try block, otherwise just values are added in except blocks.

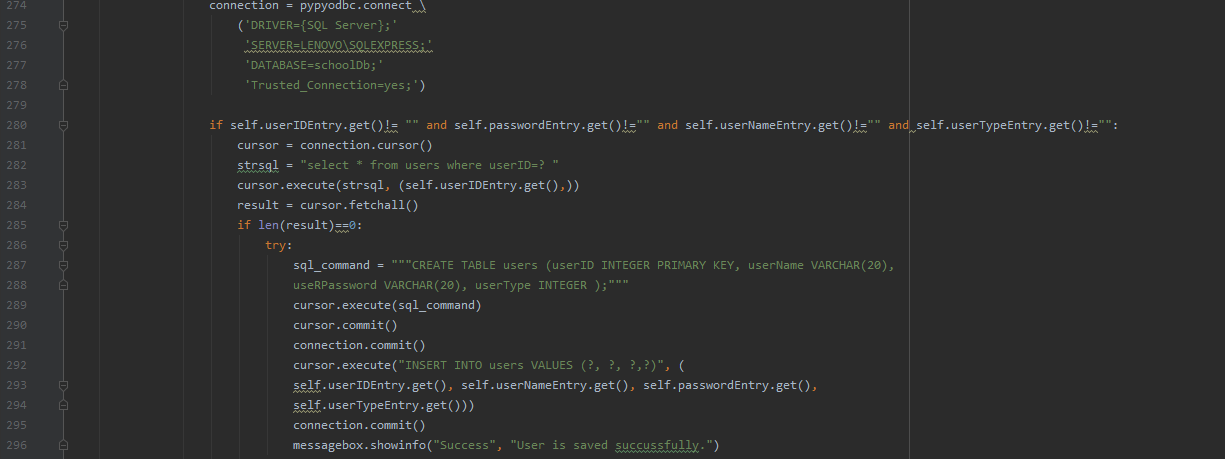


Figure 2 Create table in python

According to requirements, admin can add course and user who is admin, professor or student. Also, admin has a authority who update and delete professor, course and student. For add user, database connection is provided according to figure 2. Figure 3 will explain how admin add user.

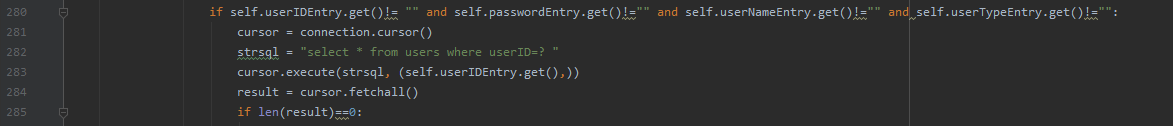


Figure 3 Add admin to database

If admin wants to add user, username, user id, user password and user type entry cannot be empty so this condition has checked in line 280. Line 282 has controlled the new user according to user id who is exists or nor in database. If user does not exist in database, result has returned 0 in line 284. When these two conditions provide new user add database. At the same, new user should add student, admin or professor table according to user type. This proviso checks figure 4.

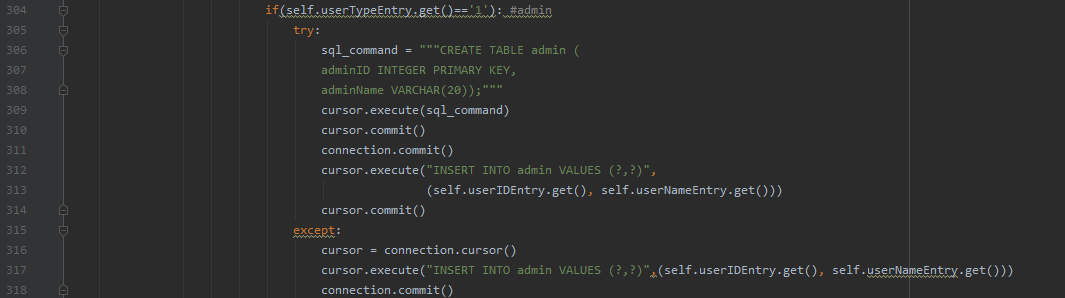


Figure 4 Add to related table according to user type

This program has a three different type user which are student, admin or professor. These types hold in database as a number. 1 represent admin, 2 represent professor and 3 represent student. In figure four example for add new admin to admin table. Line 304 checks user type which gets user type entry. After that table create, if it does not exist and new admin add database. Through connection.comit() method all differences are saved in database. As a same logic, new user is added related table.

Admin update course, professor and student. Figure 5 is an example how admin update user.

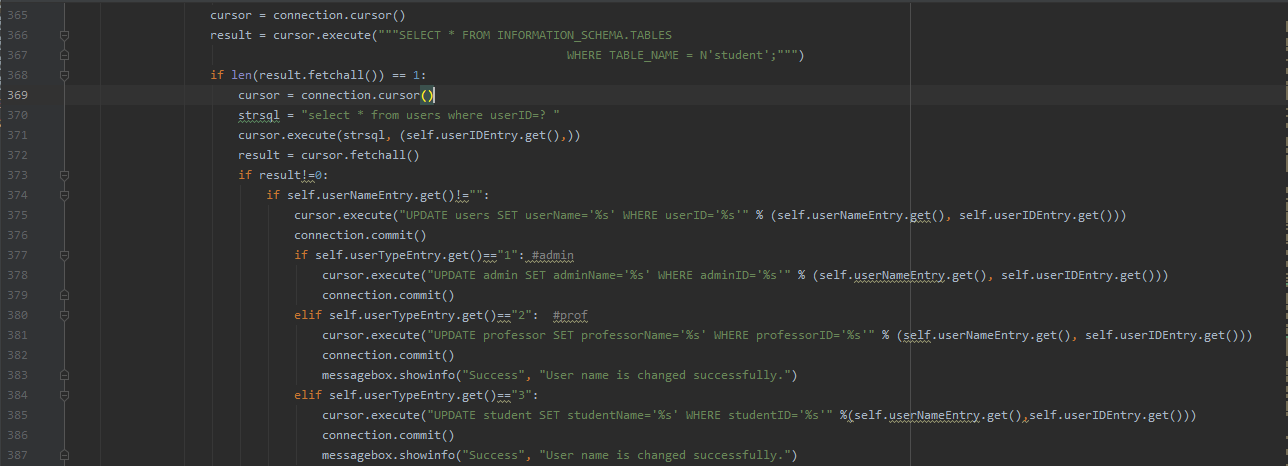


Figure 5 How admin update data

First, database connection provides based on figure 1. In line 366, table has checked which exists or not in database. If result returns 1, it means the table exits in database otherwise, admin will get an error message. In line 370, user search related table who exists or not according to user id. If result does not return 0, it means user exists in database and admin can update user. In line 370 admin can update username, if username entry is not empty. Line 374 update username in user table. At the same time username should update other tables according to user type. Line 377 updates admin table, line 376 updates professor table and line 380 updates student table. Line 384 gives error when admin does not enter user type.

When admin wants to update admin, student or professor also, changes are saved other tables which are related with the user type.

Finally, admin can delete data in database. Figure 6 is an example how admin delete course in database. Database connection made by virtue of figure 1. In line 151, table has checked which exists or not in database. If result returns 1, it means the table exits in database otherwise, admin will get an error message. In line 155, the query gets the course based on course id. In line 157, if result will return different from 0, it means the course exists in database otherwise admin will get an error message. Course has a relation between two tables which are courseStudent and professorCourse. Line 159,160 and 161 are delete course in tables.

Other delete process logic is same with course. Also, if admin wants delete professor, professor has relation two more tables which are users and professorCourse so professor’s information will delete these two table. If admin wants delete student, student has relation two more tables which are users and courseStudent so student’s information will delete these two table. If admin wants delete user, user has relation more than one table according to user type so based on this relationship, user will delete in database.

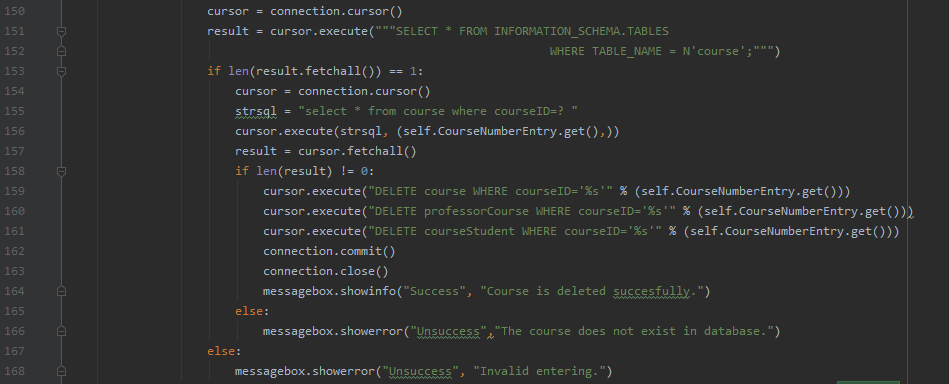


Figure 6 How admin delete course

According to requirements, professor just can add grade for student. Before professor add grade for student, database connection is provided based on figure 1. All student and grade information’s are kept in database courseStudent table so figure 7 exemplify how does table check exist or not in database. In line 63, if result is not equal to zero, table exists in database.3



Figure 7 How to check table exist or not in database

Before professor submit grade for student, s/he must enter all entry which is student id, course id and grade. This control is done figure 8, line66. When this condition is ensured, SQL command is executed which control the course is taken by student or not. If figure 8, line 70 returns different from zero, the condition is ensured. Grade can be saved database using SQL command which is showed in figure 8 line 72. For this process, update command is used because while table create, grade assign 0 automatically. All changes are saved database using connection.comit() command. If figure 8, line 70 returns zero, professor will get error which has showed figure 8, line 77.

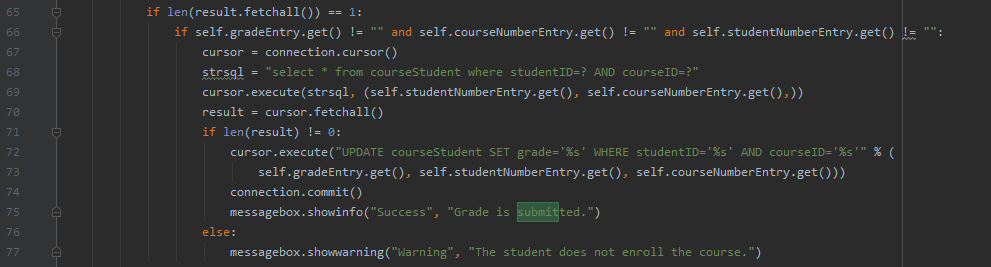


Figure 8 Submit grade

As regards student requirement who can just display their grade based on course name. As a beginning, database connection is assured model on figure 1. When database connection is successful, courseStudent table has checked which exist or not exist in database. figure 9 is indicate this condition. When result returns zero, the table exist in database and program has checked student exists in database.

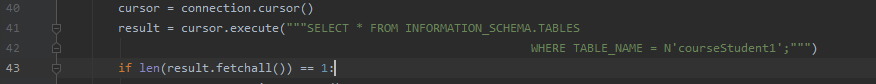


Figure 9 Table exists or not in database

Because, if student exists in database who can see her/his grade in GUI window so figure 10 is demonstrated this condition. If result returns different from zero, the student endures in database. For next step which will control student is enrolled course or not, program needs student id. Result returns as a dimensional list which first index keeps all row and second index keeps column. StudentID is kept by index 0 which is showed figure 10 in line 51.

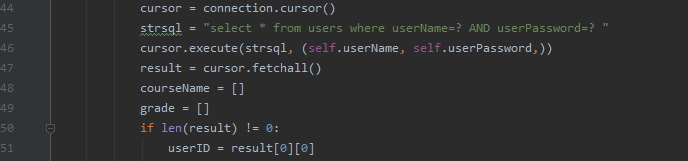


Figure 10 User exists or not in database

Using studentID, student is controlled is enrolled course or not. The related SQL command is figure 11 in line 62. Also, when this result returns as a dimensional list which include courseID in index zero. In figure 11 line 67 returns all courseID which is taken by student. Result[i] holds all records course is taken by student but for next control just needs courseID. In figure 11 line 68 gets course name from course table based on courseID. This result adds a list in figure 11 line 71. Finally, all grade is added grade list in figure 11 line 72. This time result[i][2] holds grade which relates courseID.

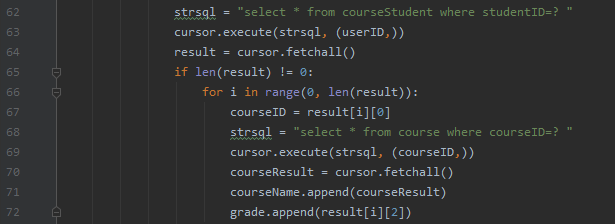


Figure 11 Get to grade

In figure 12, using for loop course name and grade is written GUI window. For loop continue length of courseName. Line 75 holds course name records by order and line 76 holds grade by order based on course name. When label adapt the GUI window, row has incremented one by one using for increment. Line 78 and 80 is an example how label import to GUI window.

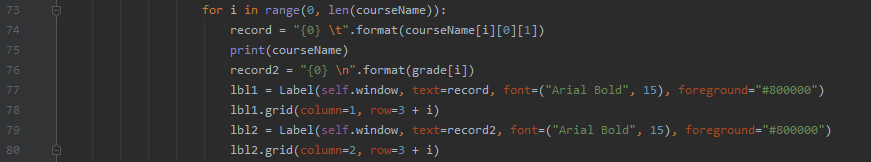


Figure 12 How grade display