# Maksims Kazoha

# R00188979

# SD2-B

# Project Documentation

# File name: SD2-BKazohaMReport

# Marketing Unit - Logos & Crests

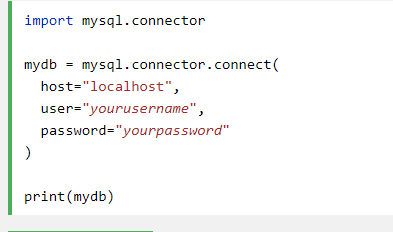
## Table of Contents

1. Project Description.
2. Technical Issues encountered.
3. Code explanation.
4. Conclusion.
5. Reference.

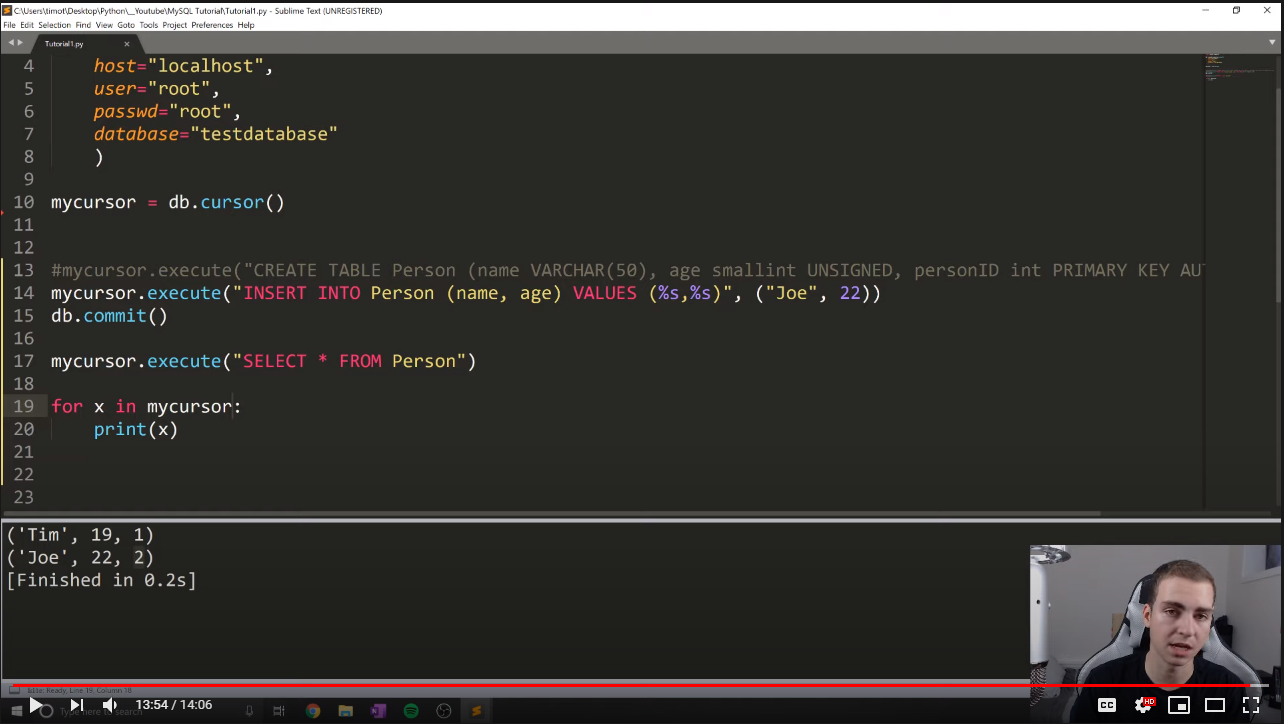
## Project Description:

For this project we needed to use a high-level and general-purpose programming language that uses an external DBMS to store the data. The programming language I’ve selected was python as it was something I already did and was very familiar with, and the DBMS I used to be MySQL to store data as it was the DBMS that I also was familiar with. We need python to interact with MySQL, so we use a connector to establish a connection between them. In MySQL all you need is to make a server i.e. localhost to run the server where the information will be stored in, and for python you can use an IDE called Visual Studio Code which allows to install external packages such as the mysql.connector and python. The program needs to be menu driven meaning you will give the user of the program options to select. Options should be as follows. Upload the information (Use Insert), Change the information (Use Update), Remove the information (Use Delete) and display the information (Use fetch). Sense the program is menu driven and the database is embedded you can use MySQL to view the information to make sure that everything is working properly. The program should be able to be execute using a command prompt not the terminal with in the VSCode, you use that to check if the program runs according to the request and fix any bugs.

## Technical issues encountered:

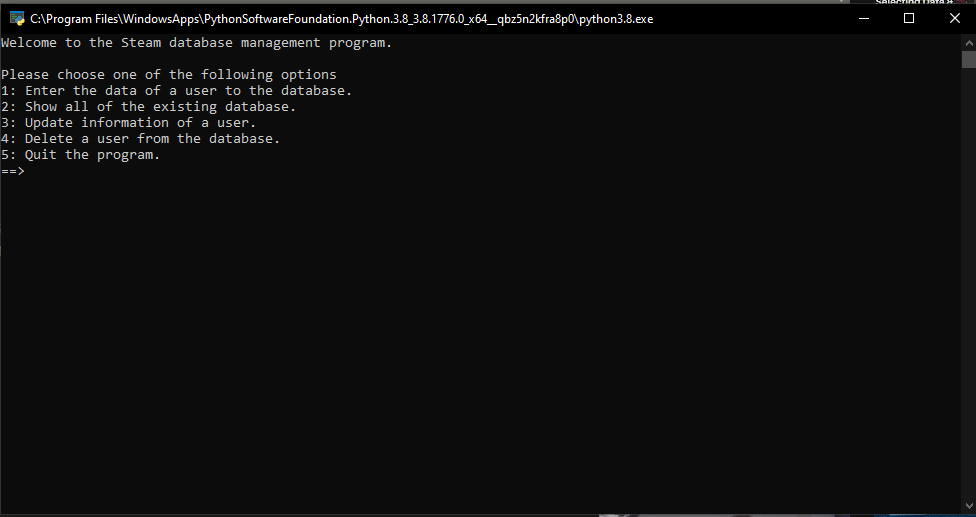
The First issue I encountered was getting mysql.connector to work as it was saying to me that I was trying to import something that doesn’t exist. So due to python being recently updated mysql.connector wouldn’t work as it wasn’t up to date with the new version of python. So, what I had to do to fix the issue was to down grade to python 3.8.6 which worked with mysql.connector. I also tried importing mysql.connector.python but due to the new version it didn’t work. In w3schools I was able to create a connection with the database 

The second issue was to figure out how to use python and mysql code in VSCode. I used a tutorial video by Tech With Tim to learn how the code works with python and MySQL.



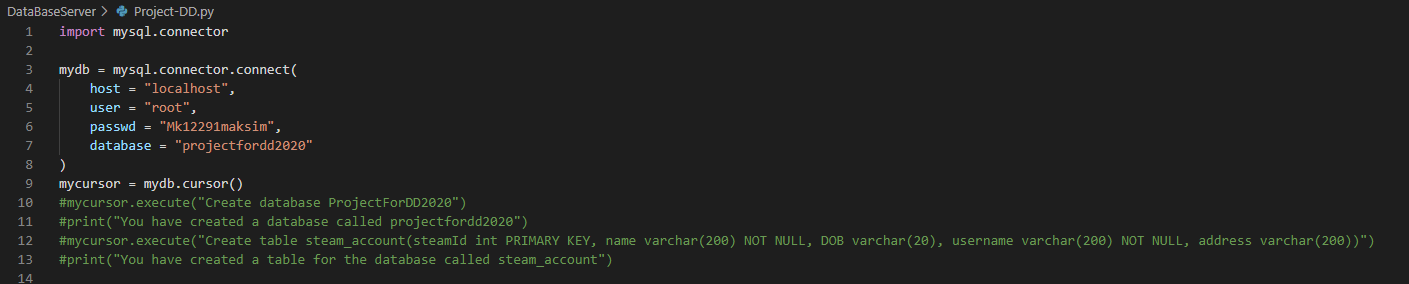
I had trouble finding out how to use my own variables like id name and address in the code and from this video I learned about string formatting. String formatting allowed me to not to hard code the program and use user input.

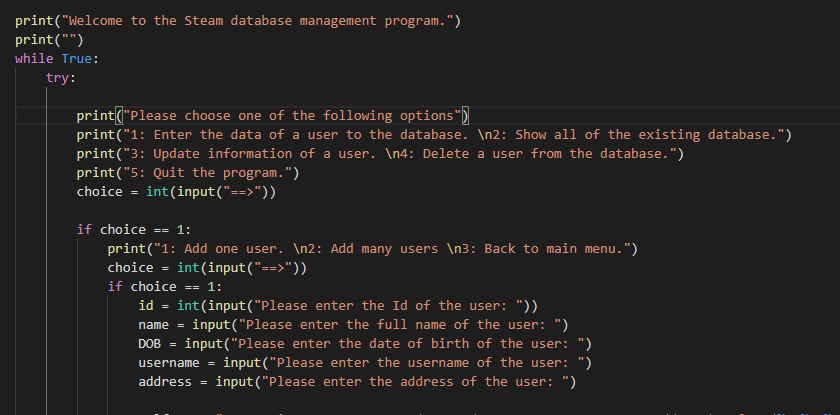
My third issue that I’m still having is that I can’t execute the code in the command prompt and I have to reside to the command prompt that is used by python. I’ve tried figuring out the issue but it seems its on my end where I’m not allowed to access the file and execute the code in the command prompt.



My last issue that I have encounter while coding was when in the delete option when I was stating what steam\_account was I had it as steam\_account = (id) but then from the video by Tech With Tim that you need to add steam\_account = (id,) for it to use the information in %s which I couldn’t figure out why.

## Code Explanation:

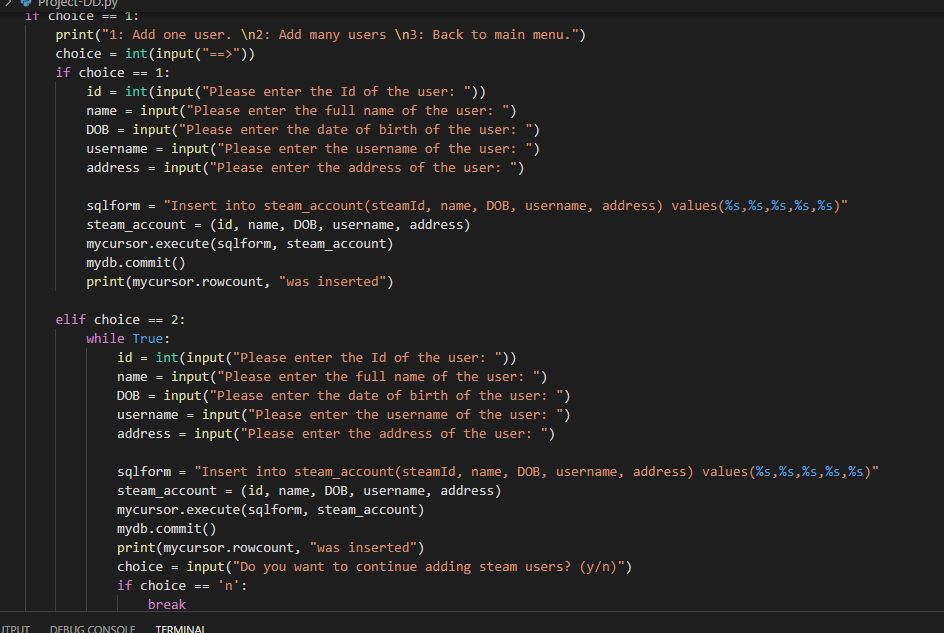
we establish the connection with the database. Localhost is the default that’s what everyone uses, user root same thing. Password doesn’t matter just protects from anyone else entering the database, and the database is what I named the database projectfordd2020 and mycousor = mydb.cursot() is what I use to execute/commit code with. The commented-out lines are for creating the database and making a database table. steamId is the primary key so it’s the unique factor of the row and no user can have the same ids.



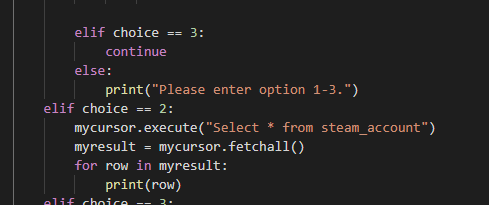
The print statement is before the while true so that it doesn’t print every time you return to the main menu. The while True is a while loop that runs each time you enter an option and return from the option. To stop the while loop you will need to use break as it trying to run the code constantly. Print statements for a menu 5 options

1: insert data 2: show data 3: update data 4: remove data and 5 close the program(break is here).

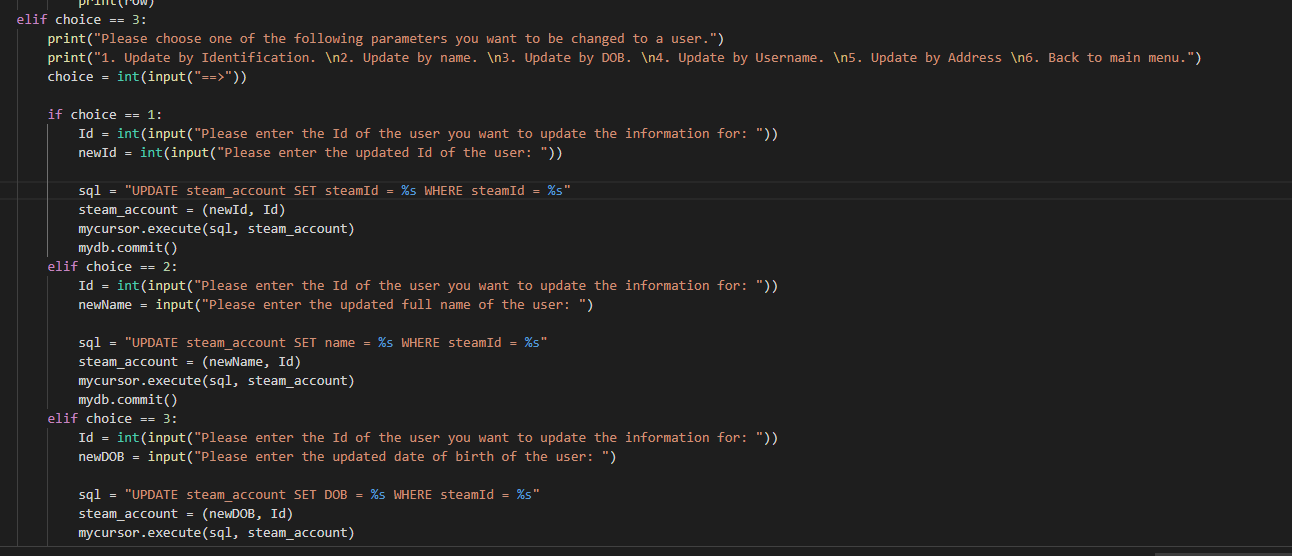
Option 1- 1: add one user(adds one user) 2: Add many users(while loop) 3: return to main menu(continue which allows the code to return to the main menu)



Id, name, DOB, username, address are variables that store the user’s input. Sqlform is the form of that I want the data to be used in. in this case its to insert data. %s are place holders and steam account determine what position the variables are going to be used in. mycursor.execute(sqlform, steam\_account) executes the code for MySQL so the data can be sent to the database. Mydb.commit() sends it away to the database. Mycoursor.rowcount counts how much rows where added. For me its always one. But it tells the user that it was added to the database. It’s the same for the choice 2 of option 1 but instead it has a while true and at the end it asks the user if they want to continue or not is the answer is n then it break the loop and returns to the main menu.

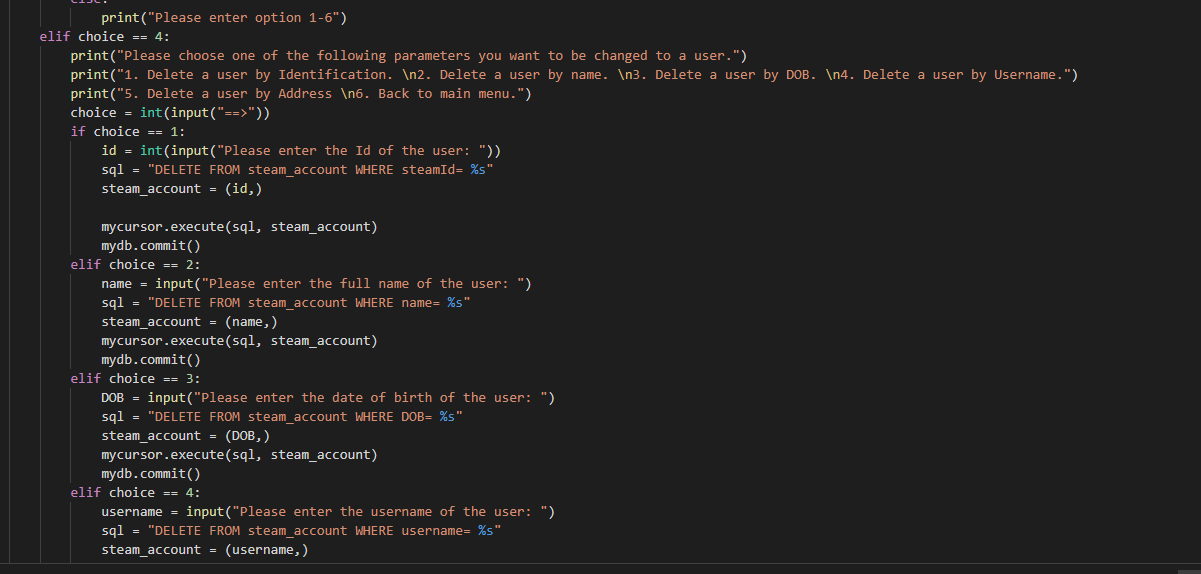


Choice 3 for option 1 is just continue that returns the user to the main menu. Option 2 selects all from the steam\_account table and fetches all the data that is in there. The for loop prints all the rows that it fetched.

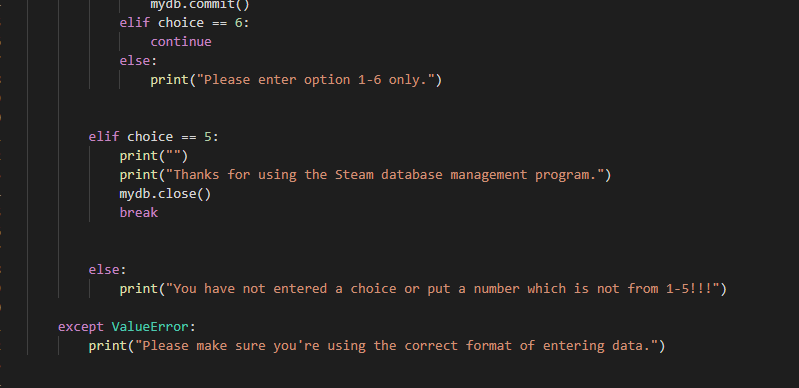


Option 3:

it has 6 choices to be made last choice is for returning to the main menu and the first 5 is for updating information by enter the steamId as a common chain. As the steamId is required to be used it’s the best way to update by the id as its common and will only update for that specific user.

Option 4:

It asks the user how do they want to delete the data by id, name, DOB, username or address if the user select one of the following options the user will need to enter the information of the row he wants to delete. The coding is similar with update but instead I’m using only one variable to remove so i.e. if I remove by address multiple people can be living in that address so all of them will be removed. As that’s the only way I learned to delete.



Option 5:

Thanks, the user for using the program and closes the database.

I have added except ValueError: so, if the user tries to enter instead of number letters, they will greeted with a message saying to use the correct format i.e. numbers instead of letters.

## Conclusion:

In conclusion I found this project very nice to work on. Due to the semester starting off poorly and a lot of deadlines for labs and projects I think I did well. The journal part was interesting as I was able to see where I was in the project as I had a plan made up. And my original plan was to short and was short term. Doing a project of this size can’t be short term as I was connecting two languages to work with each other for the first time and I’ll be honest I didn’t know what we where meant to do at the start. But in the end when I came up with a plan that contained stages, the project kicked off very nicely. As I’ve done some research on how to connect python to MySQL. I personally think that was the most difficult part of the project as finding the information without knowing what to do was difficult. But using languages that we already did was good as there was less trouble and I didn’t need to learn everything from the start. The tutorial was nice to do as it gave me a starting point from which I easily worked on the project. My project management skills could have been better, and I’ve learned my lesson of leaving projects until last minute and I’ll make sure that it won’t happen again. A very valuable lesson was learned during this project. I learned how to connect MySQL to Python how to do project management, how to keep up a journal with plans and due dates.

## Reference:

<https://www.w3schools.com/python/python_mysql_getstarted.asp> (w3schools)

<https://www.youtube.com/watch?v=91iNR0eG8kE> (Python MySQL Tutorial - Creating Tables, Inserting & Selecting By Tech With Tim)

## Appendix:

import mysql.connector

mydb = mysql.connector.connect(

    host = "localhost",

    user = "root",

    passwd = "Mk12291maksim",

    database = "projectfordd2020"

)

mycursor = mydb.cursor()

#mycursor.execute("Create database ProjectForDD2020")

#print("You have created a database called projectfordd2020")

#mycursor.execute("Create table steam\_account(steamId int PRIMARY KEY, name varchar(200) NOT NULL, DOB varchar(20), username varchar(200) NOT NULL, address varchar(200))")

#print("You have created a table for the database called steam\_account")

print("Welcome to the Steam database management program.")

print("")

while True:

    try:

        print("Please choose one of the following options")

        print("1: Enter the data of a user to the database. \n2: Show all of the existing database.")

        print("3: Update information of a user. \n4: Delete a user from the database.")

        print("5: Quit the program.")

        choice = int(input("==>"))

        if choice == 1:

            print("1: Add one user. \n2: Add many users \n3: Back to main menu.")

            choice = int(input("==>"))

            if choice == 1:

                id = int(input("Please enter the Id of the user: "))

                name = input("Please enter the full name of the user: ")

                DOB = input("Please enter the date of birth of the user: ")

                username = input("Please enter the username of the user: ")

                address = input("Please enter the address of the user: ")

                sqlform = "Insert into steam\_account(steamId, name, DOB, username, address) values(%s,%s,%s,%s,%s)"

                steam\_account = (id, name, DOB, username, address)

                mycursor.execute(sqlform, steam\_account)

                mydb.commit()

                print(mycursor.rowcount, "was inserted")

            elif choice == 2:

                while True:

                    id = int(input("Please enter the Id of the user: "))

                    name = input("Please enter the full name of the user: ")

                    DOB = input("Please enter the date of birth of the user: ")

                    username = input("Please enter the username of the user: ")

                    address = input("Please enter the address of the user: ")

                    sqlform = "Insert into steam\_account(steamId, name, DOB, username, address) values(%s,%s,%s,%s,%s)"

                    steam\_account = (id, name, DOB, username, address)

                    mycursor.execute(sqlform, steam\_account)

                    mydb.commit()

                    print(mycursor.rowcount, "was inserted")

                    choice = input("Do you want to continue adding steam users? (y/n)")

                    if choice == 'n':

                        break

            elif choice == 3:

                continue

            else:

                print("Please enter option 1-3.")

        elif choice == 2:

            mycursor.execute("Select \* from steam\_account")

            myresult = mycursor.fetchall()

            for row in myresult:

                print(row)

        elif choice == 3:

            print("Please choose one of the following parameters you want to be changed to a user.")

            print("1. Update by Identification. \n2. Update by name. \n3. Update by DOB. \n4. Update by Username. \n5. Update by Address \n6. Back to main menu.")

            choice = int(input("==>"))

            if choice == 1:

                Id = int(input("Please enter the Id of the user you want to update the information for: "))

                newId = int(input("Please enter the updated Id of the user: "))

                sql = "UPDATE steam\_account SET steamId= %s WHERE steamId= %s"

                steam\_account = (newId, Id)

                mycursor.execute(sql, steam\_account)

                mydb.commit()

            elif choice == 2:

                Id = int(input("Please enter the Id of the user you want to update the information for: "))

                newName = input("Please enter the updated full name of the user: ")

                sql = "UPDATE steam\_account SET name= %s WHERE name= %s"

                steam\_account = (newName, Id)

                mycursor.execute(sql, steam\_account)

                mydb.commit()

            elif choice == 3:

                Id = int(input("Please enter the Id of the user you want to update the information for: "))

                newDOB = input("Please enter the updated date of birth of the user: ")

                sql = "UPDATE steam\_account SET DOB= %s WHERE DOB= %s"

                steam\_account = (newDOB, Id)

                mycursor.execute(sql, steam\_account)

                mydb.commit()

            elif choice == 4:

                Id = int(input("Please enter the Id of the user you want to update the information for: "))

                newUsername = input("Please enter the updated username of the user: ")

                sql = "UPDATE steam\_account SET username= %s WHERE username= %s"

                steam\_account = (newUsername, Id)

                mycursor.execute(sql, steam\_account)

                mydb.commit()

            elif choice == 5:

                Id = int(input("Please enter the Id of the user you want to update the information for: "))

                newAddress = input("Please enter the updated address of the user: ")

                sql = "UPDATE steam\_account SET address= %s WHERE address= %s"

                steam\_account = (newAddress, Id)

                mycursor.execute(sql, steam\_account)

                mydb.commit()

            elif choice == 6:

                continue

            else:

                print("Please enter option 1-6")

        elif choice == 4:

            print("Please choose one of the following parameters you want to be changed to a user.")

            print("1. Delete a user by Identification. \n2. Delete a user by name. \n3. Delete a user by DOB. \n4. Delete a user by Username.")

            print("5. Delete a user by Address \n6. Back to main menu.")

            choice = int(input("==>"))

            if choice == 1:

                id = int(input("Please enter the Id of the user: "))

                sql = "DELETE FROM steam\_account WHERE steamId= %s"

                steam\_account = (id,)

                mycursor.execute(sql, steam\_account)

                mydb.commit()

            elif choice == 2:

                name = input("Please enter the full name of the user: ")

                sql = "DELETE FROM steam\_account WHERE name= %s"

                steam\_account = (name,)

                mycursor.execute(sql, steam\_account)

                mydb.commit()

            elif choice == 3:

                DOB = input("Please enter the date of birth of the user: ")

                sql = "DELETE FROM steam\_account WHERE DOB= %s"

                steam\_account = (DOB,)

                mycursor.execute(sql, steam\_account)

                mydb.commit()

            elif choice == 4:

                username = input("Please enter the username of the user: ")

                sql = "DELETE FROM steam\_account WHERE username= %s"

                steam\_account = (username,)

                mycursor.execute(sql, steam\_account)

                mydb.commit()

            elif choice == 5:

                address = input("Please enter the address of the user: ")

                sql = "DELETE FROM steam\_account WHERE address= %s"

                steam\_account = (address,)

                mycursor.execute(sql, steam\_account)

                mydb.commit()

            elif choice == 6:

                continue

            else:

                print("Please enter option 1-6 only.")

        elif choice == 5:

            print("")

            print("Thanks for using the Steam database management program.")

            mydb.close()

            break

        else:

            print("You have not entered a choice or put a number which is not from 1-5!!!")

    except ValueError:

        print("Please make sure you're using the correct format of entering data.")