Computer Science Project

2023 – 24

***Movie Database Management System***

***Efforts By:***

Zenil Karia

12 A CBSE

Navrachana International School, Vadodara

**ACKNOWLEDGEMENTS**

In the accomplishment of completing this project we would like to thank our computer science teacher Mrs. Poonam Sonkar for guiding us through all stages of this project. We would also like to thank our higher secondary coordinator Mr. Hussain Kagalwala for his support.

**INDEX**

|  |  |
| --- | --- |
| **PAGE NO.** | **CONTENTS** |
| **4** | **Introduction** |
| **5** | **Future Scope** |
| **6** | **System Requirements** |
| **7** | **Operating System Software** |
| **8** | **Software Developing Platform** |
| **9-24** | **Source Code** |
| **25-37** | **Output Screens** |
| **38** | **Bibliography** |

**INTRODUCTION**

Database creation and management is an important function in the rising computer industry. From banks to movies to company records,

information is important to organise in a formally classified manner and even better should be its accessibility. There is no use of data if you cannot access it.

For this, we have created a software on movie database management. There are thousands of movies releasing every year in every corner of the world and for related information to be stored and accessed, we have created this software by integrating Python with MySQL (Structured Query Language). We have made it as user-friendly as we can and the user, even with minimal knowledge of coding or queries, can easily use this program to their benefit. We decided to store the 2 basic functions of one particular movie - its IMDB rating and box office collections.

Additionally, the user can also choose to play a particular song of the movie, as well as a graph generated by Python itself for comparison among movies. Records, right from the databases, tables to the information, can be created, modified or removed depending on whatever the user wants.

**FUTURE SCOPE**

1. This project in future can be expanded to include many more songs of movies which are in the binary file by connecting it with Spotify so that the user can listen to his/her desired song of the movie he has searched for, without having saved an audio file in the path of the program.

2. This project can also be made much more user-friendly, interactive, and informative by adding more information about each and every movie stored including release, cast information etc. and adding drop down boxes through the tkinter module to ease out users' tasks.

**SYSTEM REQUIREMENTS**

Minimum -

Processors: Intel Atom processor or Intel Core i3 processor

Disk space: 1 GB

O.S: Windows 77, MACOS, UBUNTU

Python versions: 2.7.X, 3.6.X

Designed using the below configuration -

Processor: Intel Core i5 processor

4300M at 2.6 GHz

Disc Space: 2.4 GB

OS: Windows 10, MACOS and UBUNTU

Python versions: 3.X.X or higher

Note: MYSQL Connector Python requires to be in the system's path. Installation fails if it doesn't find Python.

**OPERATING SYSTEM MANAGEMENT**

This project is designed on Windows operating system where in Windows 77 and above versions of Windows can be used.



**SOFTWARE DEVELOPMENT PLATFORM**

This project is designed on Python platform, using IDLE, which is an object- oriented programming language.



**SOURCE CODE**

import pickle

from playsound import playsound

import matplotlib.pyplot as plt

import os

import mysql.connector as my

#Function to create a new file

def New\_File():

fin = open(z, 'wb+')

ans = 'y'

while ans == 'y':

mname = input('Enter Movie Name: ')

rating = float(input("Enter IMDB Rating: "))

boc = input('Enter Box Office Collection (in words): ')

s = {'Movie Name' : mname, 'IMDB Rating' : rating, 'Box Office Collection' : boc}

pickle.dump(s, fin)

ans = input('Do you want to enter more (y/n): ')

print('Thank you all file contents have been added!\n')

fin.close()

#Function to read an existing file

def Read\_File():

fin = open(z, 'rb')

try:

while True:

s = pickle.load(fin)

print(s)

except EOFError:

print('All the file contents have been shown.\n')

fin.close()

#Function to append in an exisiting file

def Append\_File():

fin = open(z, 'ab+')

ans = 'y'

while ans == 'y':

mname = input('Enter Movie Name: ')

rating = float(input("Enter IMDB Rating: "))

boc = input('Enter Box Office Collection (in words): ')

s = {'Movie Name' : mname, 'IMDB Rating' : rating, 'Box Office Collection' : boc}

pickle.dump(s, fin)

ans = input('Do you want to enter more (y/n): ')

print('Thank you. Your contents have been appended!\n')

fin.close()

#Function to search for a movie in an exisiting file

def Search\_File():

fin = open(z, 'rb')

movie = input('Enter the movie name you would like to search for: ')

try:

while True:

s = pickle.load(fin)

if s['Movie Name'] == movie:

print(s)

y = input('Would you like to also hear a song from this movie (yes/no): ')

if y.lower() == 'yes':

print('Enjoy the song!\n')

playsound(movie+'.MP3')

fin.close()

break

else:

print()

break

except EOFError:

print('No such movie exists in the file.\n')

fin.close()

#Function to modify an exisiting file

def Modify\_File():

fin = open(z, 'rb+')

ans = 'y'

mod = input('Enter the movie name whose record you want to modify: ')

try:

while True:

pos = fin.tell()

s = pickle.load(fin)

if s['Movie Name'] == mod:

nmovie = input('Enter new movie name: ')

nrating = float(input('Enter its IMDB rating: '))

ncol = input('Enter its box office collection (in words): ')

s['Movie Name'] = nmovie

s['IMDB Rating'] = nrating

s['Box Office Collection'] = ncol

fin.seek(pos)

pickle.dump(s, fin)

print('Record updated.\n')

fin.close()

break

except EOFError:

print('Movie Name not found.\n')

fin.close()

#Function to delete any record from a file

def Delete\_File():

fin = open(z, 'rb')

fout = open('1.dat', 'wb')

rec = input('Enter the movie name whose record you want to delete: ')

try:

while True:

s = pickle.load(fin)

if s['Movie Name'] == rec:

continue

else:

pickle.dump(s, fout)

except EOFError:

fin.close()

fout.close()

os.remove(z)

os.rename('1.dat', z)

print('The requested record has been deleted.\n')

#Function to plot graph for comparison

def Graph\_File():

x1 = []

x2 = []

fin = open(z, 'rb')

try:

while True:

s = pickle.load(fin)

x1.append(s['Movie Name'])

x2.append(s['IMDB Rating'])

except EOFError:

plt.barh(x1, x2, label = 'IMDB Ratings')

plt.title('Movies vs Ratings')

plt.xlabel('IMDB Ratings')

plt.ylabel('Movie Names')

plt.legend()

for index, value in enumerate(x2):

plt.text(value, index, str(value))

plt.show()

print('Graph plotted.\n')

fin.close()

#Function to work with MySQL using Python

def Work\_MySQL\_File():

#Function to create a database

def Create\_Database():

dataname = input('Enter the name of the database you want to create: ')

cur.execute('create database '+dataname)

mycon.commit()

print('Database created!\n')

#Function to use a database

def Use\_Database():

cur.execute('show databases')

sdata = cur.fetchall()

print(sdata)

duse = input('Which database do you want to use: ')

duse1 = 'use '+duse

cur.execute(duse1)

mycon.commit()

print('Database changed!\n')

#Function to show tables

def Show\_Tables():

cur.execute('show tables')

sdata = cur.fetchall()

print(sdata)

mycon.commit()

print('All tables shown!\n')

#Function to create a table

def Create\_Table():

tname = "create table "+z[0:9]+" (Movie\_name varchar(50) primary key, IMDB\_Rating float not null, Box\_Office\_Collection varchar(20) not null)"

cur.execute(tname)

mycon.commit()

print('Table created!\n')

#Function to describe a table

def Describe\_Table():

cur.execute('describe '+z[0:9])

des = cur.fetchall()

print(des)

mycon.commit()

print('Table described!\n')

#Function to insert values into a table

def Insert\_Values():

values = input('Do you want to insert values yourself or you would like to copy them from '+z+' (Enter y/c) : ')

if values == 'y':

ans = 'yes'

while ans == 'yes':

mname = input('Enter Movie name: ')

mimdb = float(input('IMDB Rating: '))

mbox = input("Enter box office collection in words: ")

cur.execute('insert into '+z[0:9]+' values("{}, {}, "{}")').format(mname, mimdb, mbox)

mycon.commit()

ans = input('Do you want to enter more (yes/no): ')

print('Entries added!\n')

else:

fin = open(z, 'rb')

try:

while True:

s = pickle.load(fin)

com = "insert into "+z[0:9]+" values('{}', {}, '{}')".format(s['Movie Name'], s['IMDB Rating'], s['Box Office Collection'])

cur.execute(com)

except EOFError:

mycon.commit()

fin.close()

print('Entries Copied!\n')

#Function to alter a table

def Alter\_Table():

alttable = input('What do you want to do (modify a column (Enter m)/change a column name (Enter c): ')

if alttable == 'm':

mcol = input('Enter the name of the column you want to modify: ')

dcol = input('Enter datatype and size (Eg: varchar(5)): ')

cur.execute('alter table '+z[0:9]+' modify '+mcol+' '+dcol)

mycon.commit()

print('Column modified!\n')

else:

cold = input('Enter old column name: ')

cnew = input('Enter new column name: ')

ccol = input('Enter datatype and size (Eg: varchar(5)): ')

cur.execute('alter table '+z[0:9]+' change '+cold+' '+cnew+' '+ccol)

mycon.commit()

print('Column name changed!\n')

#Function to update a table

def Update\_Table():

upd = input('What do you want to update (IMDB Rating (Enter i)/Box Office collection (Enter bo): ')

if upd == 'i':

mname = input('Enter the movie whose record you want to update: ')

newimdb = float(input('Enter new imdb rating: '))

upd1 = 'update '+z[0:9]+' set IMDB\_Rating = {} where Movie\_name = "'+mname+'"'

upd2 = upd1.format(newimdb)

cur.execute(upd2)

mycon.commit()

print("Values updated!\n")

elif upd =='bo':

mname = input('Enter the movie whose record you want to update: ')

newbox = input('Enter new box office collection (in words): ')

upd1 = 'update '+z[0:9]+' set Box\_Office\_Collection = "{}" where Movie\_name = "'+mname+'"'

upd2 = upd1.format(newbox)

cur.execute(upd2)

mycon.commit()

print("Box Office Value updated!\n")

else:

print('Enter valid input!')

#Function to display records

def Display\_Records():

disp = input('Do you want to display all records (Enter all) or any specific record (Enter sp): ')

if disp == 'all':

cur.execute('select \* from '+z[0:9])

data = cur.fetchall()

for i in data:

print(i)

print()

else:

mname = input('Enter the movie whose record you want to see: ')

cur.execute('select \* from '+z[0:9]+' where Movie\_name = "'+mname+'"')

data = cur.fetchall()

print(data+'\n')

#Function to delete records

def Delete\_Records():

delete = input('Do you want to delete all records (Enter all) or any specific record (Enter sp): ')

if delete == 'all':

cur.execute('delete from '+z[0:9])

mycon.commit()

print('Contents deleted!\n')

else:

mname = input('Enter the movie whose record you want to delete: ')

cur.execute('delete from '+z[0:9]+' where Movie\_name = "'+mname+'"')

mycon.commit()

print('Record deleted!\n')

#Function to drop a table

def Drop\_Table():

tab = input('Which table do you want to delete: ')

d = input('Are you sure you want to delete the table (y/n): ')

if d == 'y':

cur.execute('drop table '+tab)

mycon.commit()

print('Table deleted!\n')

else:

print('No problem!\n')

#Function to drop a database

def Drop\_Database():

duse = input('Which database do you want to delete: ')

d = input('Are you sure you want to delete the database (y/n): ')

if d == 'y':

cur.execute('drop database '+duse)

mycon.commit()

print('Database deleted!\n')

else:

print('No problem!\n')

p = input('Enter your MySQL password: ')

mycon = my.connect(host='localhost', user='root',passwd=p,use\_pure=True,charset='utf8')

print()

cur = mycon.cursor()

o = 14

while o != 13:

print("1. Create Database \n"+"2. Use Database \n"+"3. Show Tables \n"+"4. Create table \n"+"5. Describe Table \

\n"+"6. Insert values \n"+"7. Alter a table \n"+"8. Update a table \n"+"9. Display all or any specific record \

\n"+"10. Delete a record or all records \n"+"11. Drop Table \n"+"12. Drop Database \n"+"13. Return to main program")

o = int(input('Which of the following would you like to do on MySQL: '))

if o == 1:

Create\_Database()

elif o == 2:

Use\_Database()

elif o == 3:

Show\_Tables()

elif o == 4:

Create\_Table()

elif o == 5:

Describe\_Table()

elif o == 6:

Insert\_Values()

elif o == 7:

Alter\_Table()

elif o == 8:

Update\_Table()

elif o == 9:

Display\_Records()

elif o == 10:

Delete\_Records()

elif o == 11:

Drop\_Table()

elif o == 12:

Drop\_Database()

elif o == 13:

mycon.close()

print()

break

#Menu Option Function

def Menu(ch=1):

while ch!=9:

print("\t", '-'\*10, z, '-'\*10)

print("\n1. New File \n"+"2. Show all records \n"+"3. Append \n"+"4. Search for a movie \

\n"+"5. Modify \n"+"6. Delete any record \n"+"7. Plot graph \n"+"8. Work with MySQL \n"+"9. Back to main menu \n")

ch = int(input('Please choose one option: '))

if ch == 9:

break

else:

if ch == 1:

New\_File()

elif ch == 2:

Read\_File()

elif ch == 3:

Append\_File()

elif ch == 4:

Search\_File()

elif ch == 5:

Modify\_File()

elif ch == 6:

Delete\_File()

elif ch == 7:

Graph\_File()

elif ch == 8:

Work\_MySQL\_File()

else:

print('Please enter a valid input.\n')

#Main Program

print('\t\tWelcome to International Movie Database Collection (IMDB).\n\n\t This program will help you to perform the following functions on your file.\n\n')

while ch != 3:

print("File Options:\n1. Bollywood.dat\n2. Hollywood.dat\n3. Exit")

fname = int(input('Choose: '))

if fname == 1:

z = "Bollywood.dat"

Menu()

elif fname == 2:

z = "Hollywood.dat"

Menu()

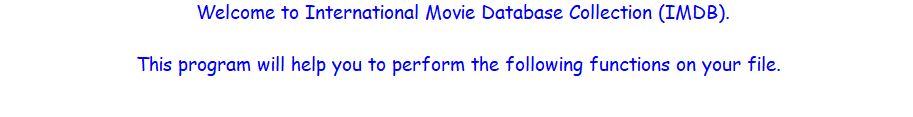
else:

print('Thanks for using this software. Have a Good Day!')

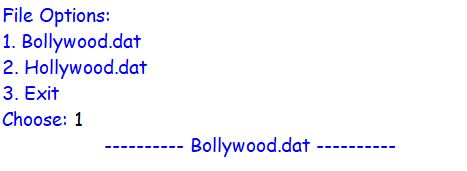
break

**OUTPUT SCREENS**

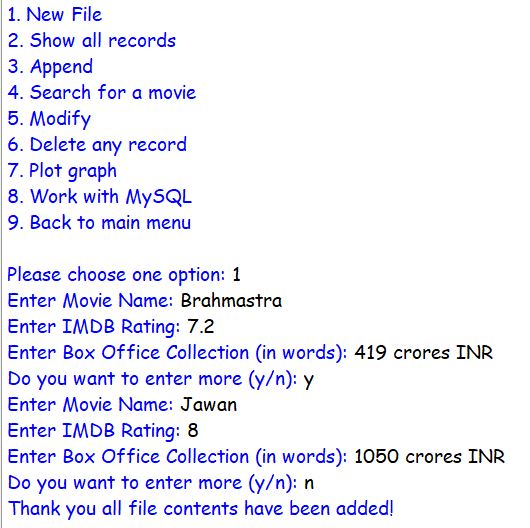
Welcome Screen



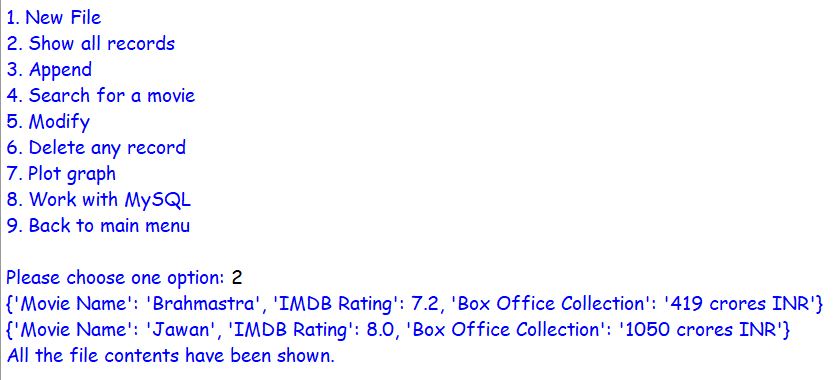
Using any one File



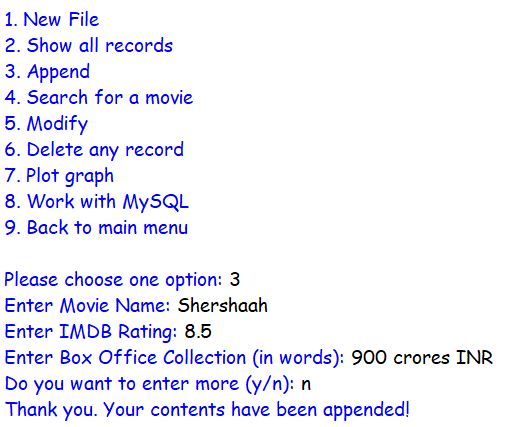
Creating a new file



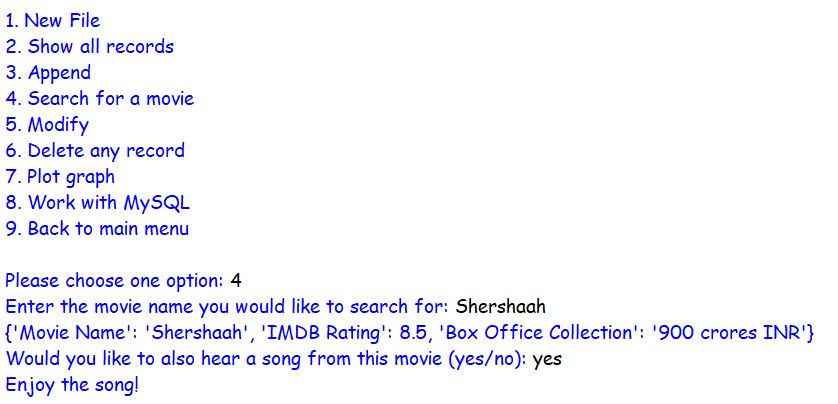
Reading a File



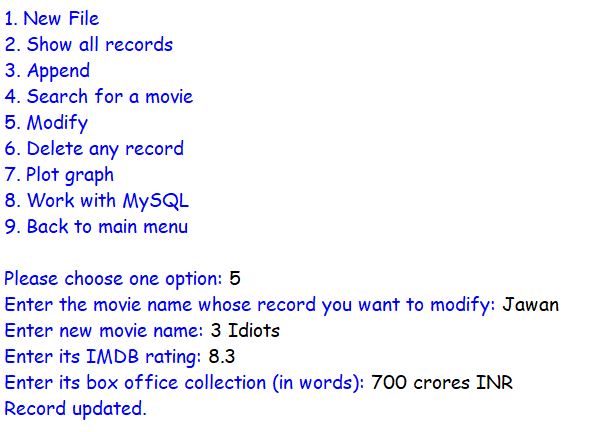
Appending in an existing file



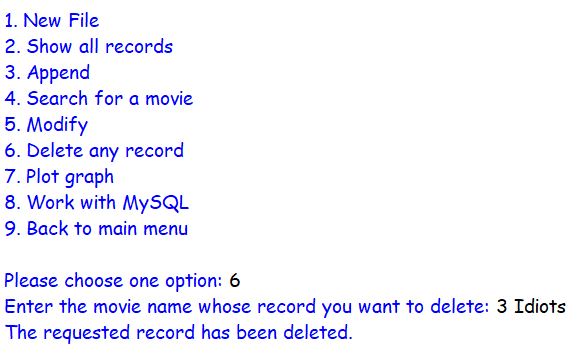
Searching for a movie in a file



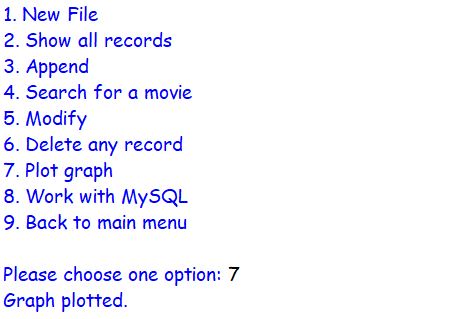
Modify an existing file

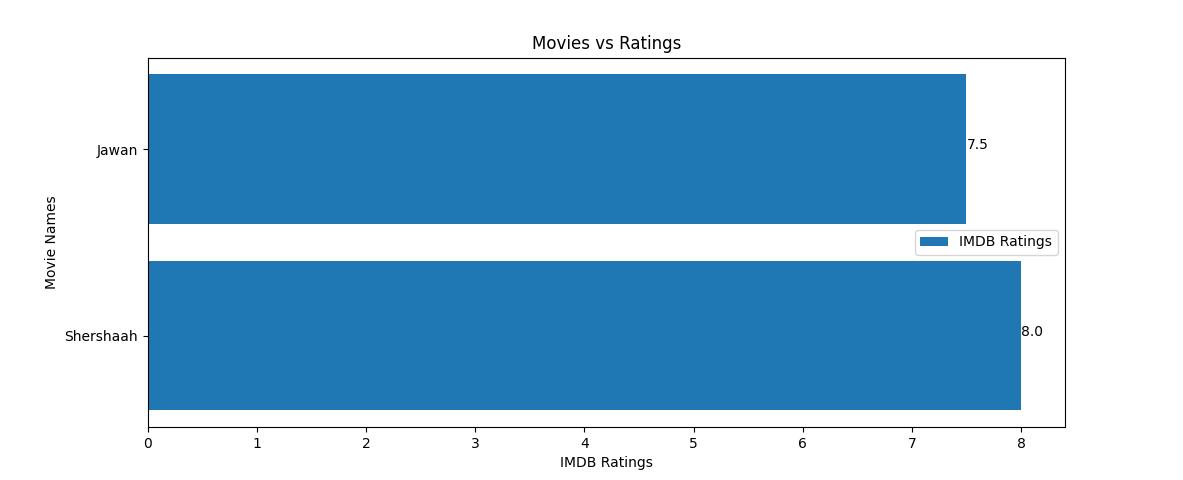


Delete a specific record

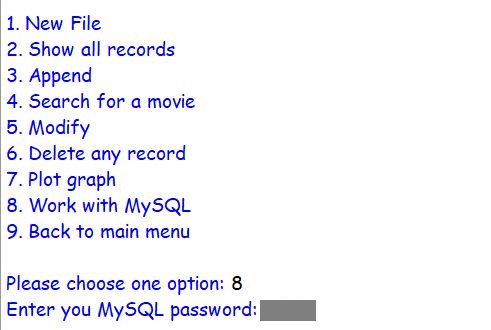


Graph Plotting

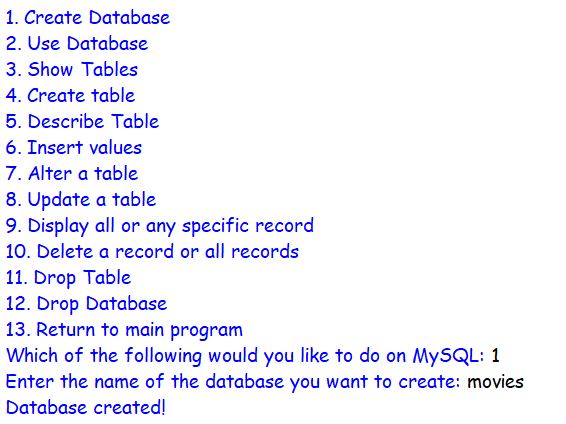




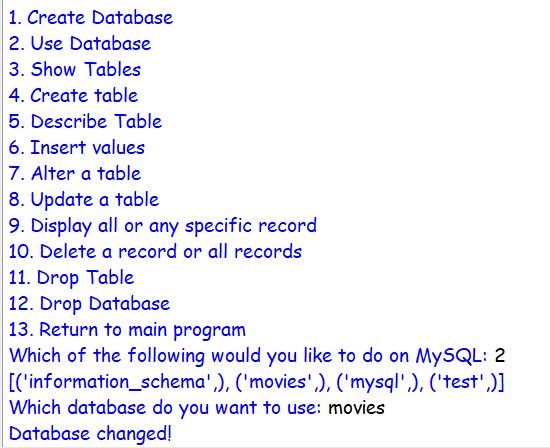
Work with MySQL



Creating a database



Using a database



Show Tables



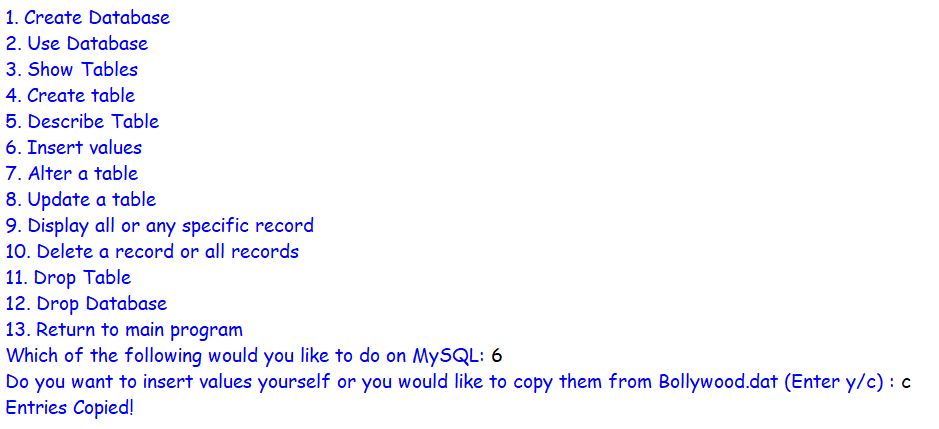
Create a Table



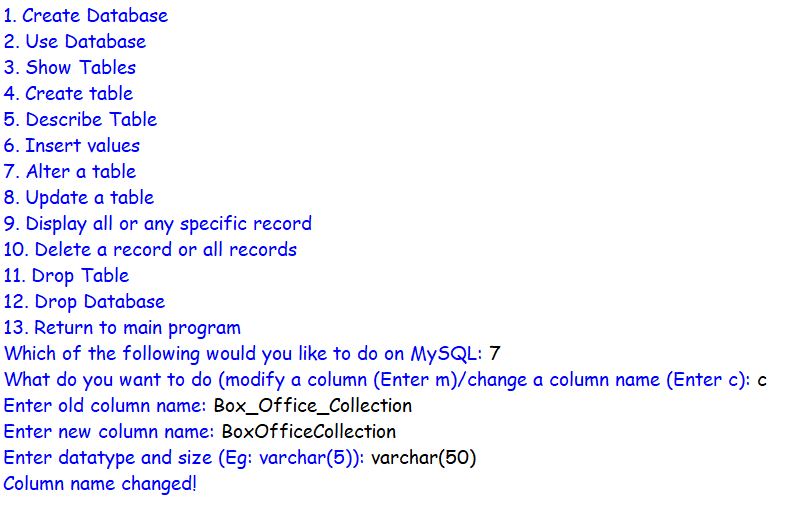
Describe Table



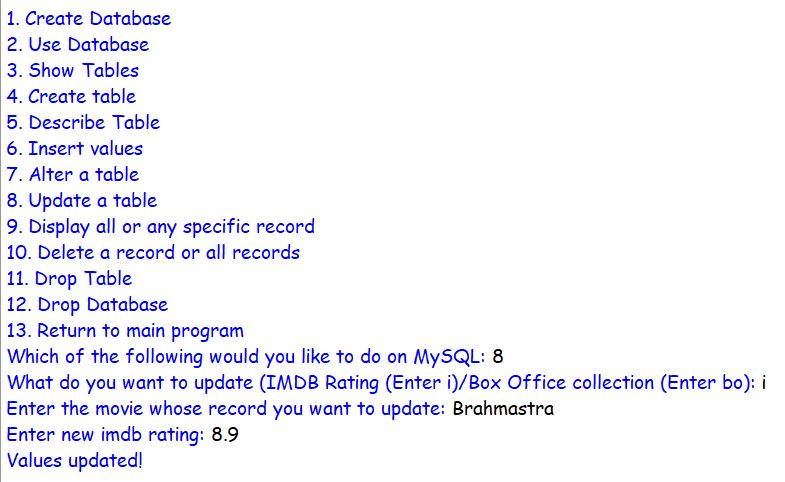
Inserting values into a Table



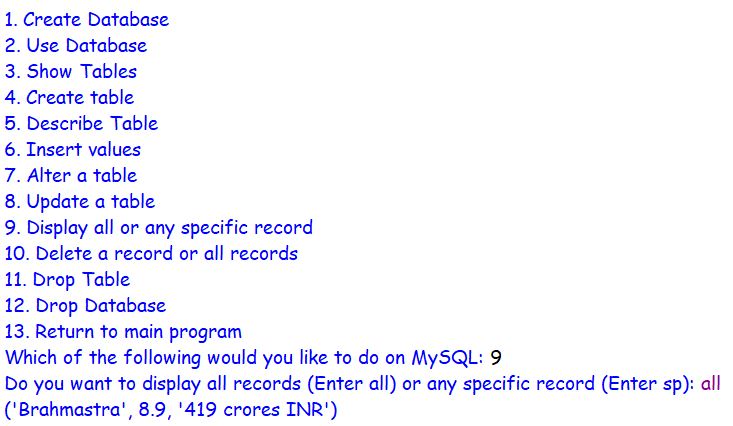
Alter a Table



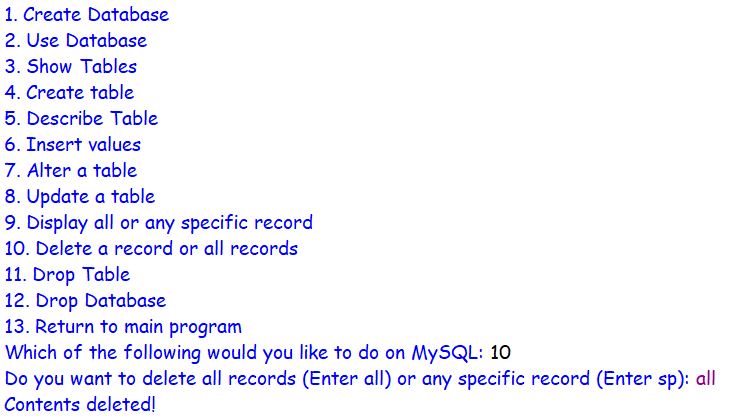
Update a Table: IMDB Ratings



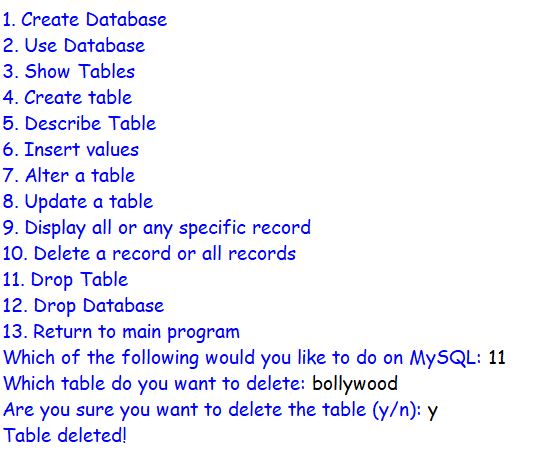
Display Records



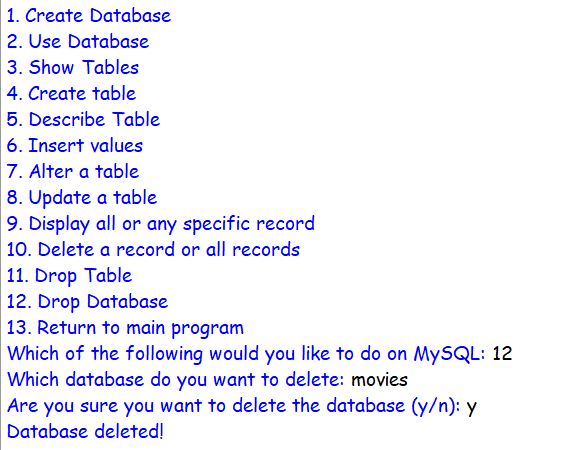
Delete Records from a Table



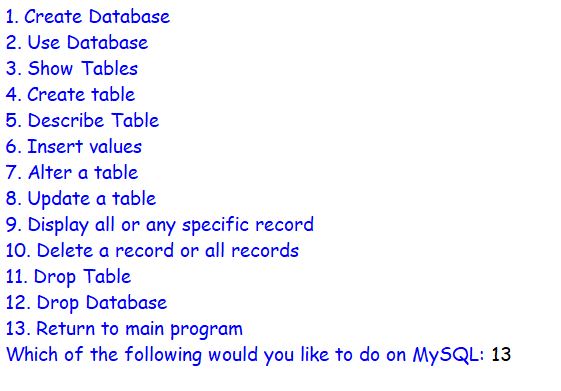
Drop Table



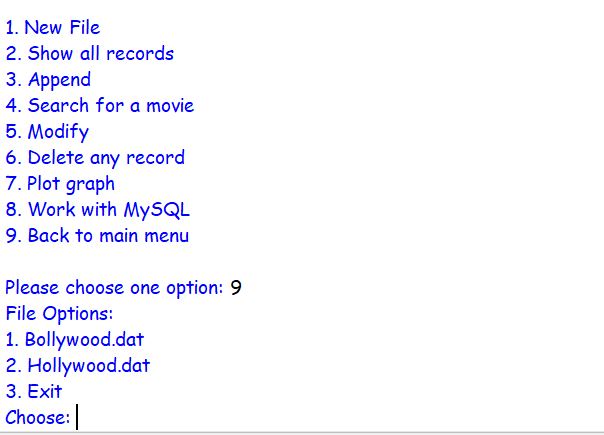
Drop Database



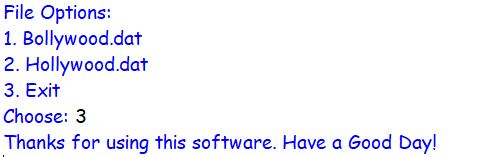
Return to main program



Back to main menu



Exit



**BIBLIOGRAPHY**

* Class XII Computer Science with Python Textbook by Sumita Arora
* <https://www.geeksforgeeks.org/play-sound-in-python/>
* <https://www.w3schools.com/python/matplotlib_pyplot.asp>