SOURCE CODE

main (2).py - /Users/annanyajain/Downloads/main (2).py (3.7.3)

```
import mysql.connector as sqlt
import pandas as pd
from tabulate import tabulate
import matplotlib.pyplot as plt
import numpy as np

con= sqlt.connect(host= "localhost", user= "root", password = "Annanya@123", database = "library" )
cursor= con.cursor()
def book_input () :
  try:
      Book_id= input("Enter Book Id : ")
      Book_Name= input("Enter Book Name : ")
      Book_Author= input("Enter Book Author : ")
      Book_Genre= input("Enter Book Genre : ")
      Book_Price = float(input("Enter Book Price : "))
      No_Copies= int(input("Enter Number of copies : "))
      qry= "insert into Book values('{}','{}','{}','{}','{}','{}','{}');".format(Book_id,Book_Name,Book_Author,Book_Genre,Book_Price,No_Copies,No_Copies)
      cursor.execute(qry)
      con.commit()
      print(" Book Added successfully \n ")
  except:
      print("Wrong Entry \n ")

def update_book() :
    try:
        x = int(input("Enter Book Id : "))
        qry= "select * from Book where Book_Id = {};".format(x)
        cursor.execute(qry)
        r= cursor.fetchone()
        if r:
            y= float(input("Enter New Book Price : "))
            z= float(input("Enter Number of Book copies : "))
            w= float(input("Enter Number of Remaining Book copies : "))
            qry= "update Book set Book_Price = {}, No_Copies = {},  rem_copies = {} where Book_Id = {};".format(y,z,w,x)

            cursor.execute(qry)
            con.commit()
            print("Book details Updated Successfully \n ")
        else:
            print("Please enter a valid Book ID ")
    except:
        print("Wrong Entry\n")
```

```
def book_delete():
    try:
        x = int(input("Enter Book Id : "))
        qry= "select * from Book where Book_Id = {};".format(x)
        cursor.execute(qry)
        r= cursor.fetchone()
        if r:
            qry= "delete from Book where Book_Id = {};".format(x)
            cursor.execute(qry)
            con.commit()
            print("Book Deleted Successfully \n ")
        else:
            print("Please enter a valid Book ID ")
    except:
        print("Wrong Entry\n")

def search_book():
    try:
        x = int(input("Enter Book Id : "))
        qry= "select * from Book where Book_Id = {};".format(x)
        cursor.execute(qry)
        r= cursor.fetchone()
        if r:
            df=pd.read_sql(qry,con)
            print(tabulate(df,headers='keys',tablefmt= 'psql', showindex= False))

        else:
            print("Please enter a valid Book ID ")
    except:
        print("Wrong Entry\n")
```

main (2).py - /Users/annanyajain/Downloads/main (2).py

```
def member_input () :
  try:
      Mem_id = input("Enter Member Id : ")
      Mem_Name= input("Enter Member Name : ")
      Mem_Add= input("Enter Member Address : ")
      Mem_phno = input("Enter Member Phone No : ")

      qry= "insert into member values('{}','{}','{}','{}');".format(Mem_id,Mem_Name,Mem_Add,Mem_phno)
      cursor.execute(qry)
      con.commit()
      print(" Member Added successfully \n ")
  except:
       print("Wrong Entry\n")

def update_member() :
   try:
       x = int(input("Enter Member Id: "))
       qry= "select * from member where Mem_id = {};".format(x)
       cursor.execute(qry)
       r= cursor.fetchone()
       if r:
           y= input("Enter New Address : ")
           z= int(input("Enter New phone No: "))
           qry= "update member set Mem_Add = '{}', Mem_phno = '{}' where Mem_id = {};".format(y,z,x)

           cursor.execute(qry)
           con.commit()
           print("Member details Updated Successfully \n ")
       else:
           print("Please enter a valid Member ID ")
   except:
        print("Wrong Entry \n")

def Member_delete():
    try:
        x = int(input("Enter Member Id : "))
        qry= "select * from member where Mem_Id = {};".format(x)
        cursor.execute(qry)
        r= cursor.fetchone()
        if r:
            qry= "delete from member where Mem_id = {};".format(x)
            cursor.execute(qry)
            con.commit()
            print("Deleted Successfully \n ")
        else:
            print("Please enter a valid Member ID ")
    except:
        print("Wrong Entry \n")

def search_member():
   try:
       x = int(input("Enter Member Id : "))
       qry= "select * from member where Mem_id = {};".format(x)
       cursor.execute(qry)
       r= cursor.fetchone()
       if r:
           df=pd.read_sql(qry,con)
           print(tabulate(df,headers='keys',tablefmt= 'psql', showindex= False))

       else:
           print("Please enter a valid Member ID ")
   except:
       print("Wrong Entry \n")
```

main (2).py - /Users/annanyajain/Downloads/main (2).py

```
def book_output():
    df=pd.read_sql("select * from Book",con)
    print(tabulate(df,headers='keys',tablefmt='psql', showindex= False))

def member_output():
    df=pd.read_sql("select * from member ",con)
    print(tabulate(df,headers='keys',tablefmt='psql', showindex= False))

def return_output():
    df=pd.read_sql("select * from returnbook ",con)
    print(tabulate(df,headers='keys',tablefmt='psql', showindex= False))

def issue_output():
    df=pd.read_sql("select * from issuebook",con)
    print(tabulate(df,headers='keys',tablefmt='psql', showindex= False))

def book_issue():
    q = "select max(issue_id) from issuebook;"
    cursor.execute(q)
    r= cursor.fetchone()[0]
    if r:
        issue_id = r+1
    else:
        issue_id = 1
    x=int(input ("Enter Member ID"))
    q1="select * from member where  Mem_id = {};".format(x)
    cursor.execute(q1)
    r=cursor.fetchone()
    if r:
        y=int(input ("Enter Book ID"))
        q2= "select Book_id,rem_copies from Book where Book_id = {};".format(y)
        cursor.execute(q2)
        r=cursor.fetchone()
        if r:
            if r[1]>0:
                issue_date = input("Enter Issue Date")
                No_Copies = int(input("Enter Number of Copies"))
                rem_copies = r[1] - No_Copies
                q3 = "insert into issuebook values('{}','{}','{}','{}','{}');".format(issue_id, issue_date, x,y, No_Copies)
                cursor.execute(q3)
                q4 = "update Book set rem_copies = {} where Book_id ={};".format(rem_copies,y)
                cursor.execute(q4)
                con.commit()
                print("Book Issued ....")

            else:
                print ("Book is not available")
        else:
            print("Wrong Book id")
    else:
        print("Wrong Member id")
```

```
def book_return():
    q = "select max(return_id) from returnbook;"
    cursor.execute(q)
    r= cursor.fetchone()[0]
    if r:
        return_id = r+1
    else:
        return_id = 1
    x=int(input("Enter Member ID : "))
    q1 = "select * from member where Mem_id = {};".format(x)
    cursor.execute(q1)
    r = cursor.fetchone()
    if r:
        y=int(input ("Enter Book ID"))
        q2= "select Book_id,rem_copies from Book where Book_id = {};".format(y)
        cursor.execute(q2)
        r=cursor.fetchone()
        if r:
            return_date = input("Enter Return Date")
            No_Copies = int(input("Enter Number of Copies"))
            rem_copies = r[1] + No_Copies
            q3 = "insert into returnbook values('{}','{}','{}','{}','{}');".format(return_id, return_date, x,y, No_Copies)
            cursor.execute(q3)
            q4 = "update Book set rem_copies = {} where Book_id ={};".format(rem_copies,y)
            cursor.execute(q4)
            con.commit()
            print("Book Returned...")

        else:
            print("Wrong Book id")

    else:
        print("Wrong Member id")
```

```
def col_chart():
    q= "select Book_id, sum(No_Copies) as totalcopies from issuebook group by Book_id"
    df= pd.read_sql(q,con)
    print(df)
    plt.bar(df.Book_id, df.totalcopies, color = 'b')
    plt.xlabel("Book ID ")
    plt.ylabel("Number of Copies Issued")
    plt.title("BEST READING BOOK")
    plt.xticks(df.Book_id)
    plt.grid()
    plt.show()

def copies_balanced():
    q= "select Book_Name, No_Copies from Book"
    df= pd.read_sql(q,con)
    print(df)
    typ=input("Graph Type [Line-L,Bar-B]:")
    if typ=='L':
        x1=df.Book_Name
        y1=df.No_Copies
        n=np.arange(len(df))
        plt.plot(n, y1,color='m', linestyle='-.',linewidth=2,marker='o')
        plt.xticks(n,x1,rotation=20)
        plt.xlabel("Book Name-->",fontsize=12,color='m')
        plt.ylabel("Number of Copies-->",fontsize=12,color='m')
        plt.title(" BOOK INVENTORY ", fontsize=16,color='r')
        plt.show()

    elif typ=='B':
        df.plot('Book_Name','No_Copies', kind='bar', color='g',rot=10)
        plt.title("Plot: BOOK INVENTORY  ",fontsize=16,color='r')
        plt.xlabel("Book Name-->",fontsize=12,color='m')
        plt.ylabel("No.of Copies-->",fontsize=12,color='m')
        plt.grid()
        plt.show()
```

```
print("WELCOME TO OUR ONLINE PORTAL ")
while(True):
    print("="*80)
    print("\t\t\t-------LIBRARY MANAGEMENT SYSTEM--------\n")
    print("="*80)
    print('''\t\t\t\t Enter your choice : \n\t\t\t\t 1. Book Details \n\t\t\t\t
                                                    2. Member details \n\t\t\t\t
                                                    3. Transaction \n\t\t\t\t
                                                    4. Reports \n\t\t\t\t
                                                    5. Exit ''')
    choice=int(input("Enter your numerical choice:"))
    if choice==1:
         while (True):
             print( ''' Enter your choice : \n\t\t\t\t 1. Add Book Details \n\t\t\t\t
                                                      2. Update Book Details \n\t\t\t\t
                                                      3. Delete A Book \n\t\t\t\t
                                                      4. Search A Book \n\t\t\t\t
                                                      5. Back to main menu \n\t\t\t\t ''')
             ch=int(input("Enter your numerical choice:"))
             if ch==1:
                 book_input ()
             elif ch==2:
                 update_book()
             elif ch==3:
                 book_delete()
             elif ch==4:
                 search_book()
             elif ch==5:
                 break

    elif choice==2:
        while (True):
             print( '''\t\t\t\t Enter your choice : \n\t\t\t\t 1. Add Member Details \n\t\t\t\t
                                                    2. Update Member Details \n\t\t\t\t
                                                    3. Delete A Member \n\t\t\t\t
                                                    4. Search A Member \n\t\t\t\t
                                                    5. Back to Main Menu \n\t\t\t\t ''')
             ch=int(input("Enter your numerical choice:"))
             if ch==1:
                 member_input()
             elif ch==2:
                 update_member()
             elif ch==3:
                 Member_delete()
             elif ch==4:
                 search_member()
             elif ch==5:
                 break

    elif choice==3:
        while (True):
             print( '''\t\t\t\t Enter your choice : \n\t\t\t\t 1. Issue Book \n\t\t\t\t
                                                    2. Return Book \n\t\t\t\t
                                                    3. Back to Main Menu \n\t\t\t\t ''')
             ch=int(input("Enter your numerical choice:"))
             if ch==1:
                 book_issue()
             elif ch==2:
                book_return()
             elif ch==3:
                 break
```

```
elif choice==4:
    while (True):
        print('''\t\t\t\t Enter your choice : \n\t\t\t\t 1. Book Details \n\t\t\t\t
                                                    2. Member Details  \n\t\t\t\t
                                                    3. Issue Details \n\t\t\t\t
                                                    4. Return Details \n\t\t\t\t
                                                    5. Best Reading Book(Chart) \n\t\t\t\t
                                                    6. No of copies balanced in Library \n\t\t\t\t
                                                    7. Back to Main Menu \n\t\t\t\t''')
        ch=int(input("Enter your numerical choice:"))
        if ch==1:
            book_output()
        elif ch==2:
            member_output()
        elif ch==3:
            issue_output()
        elif ch==4:
            return_output()
        elif ch==5:
            col_chart()
        elif ch==6:
            copies_balanced()
        elif ch==7:
            break


elif choice ==5:
    break
```