

DBMS PROJECT

CASE STUDY

TOPIC-

LIBRARY MANAGEMENT SYSTEM

(Library system makes the work of a person who is in charge of the library more convenient to search, arrange and make an inventory of the contents of the library.)

BY-RAVI VARSHNEY

SEC-C ,ROLL NO.-40

UNI. ROLL NO.-191500642

B.TECH CS - 3RD SEMESTER

DATA REQUIREMENTS-

Book Name

Customer details like as-name,address,email id ,phone number,etc.

Each book will have a unique identification number and other details including a rack number which will help to physically locate the book.

There could be more than one copy of a book, and library members should be able to check-out and reserve any copy. We will call each copy of a book, a book item.

The system should be able to retrieve information like who took a particular book or what are the books checked-out by a specific library member.

There should be a maximum limit (5) on how many books a member can check-out.

There should be a maximum limit (10) on how many days a member can keep a book.

The system should be able to collect fines for books returned after the due date.

Members should be able to reserve books that are not currently available.

The system should be able to send notifications whenever the reserved books become available, as well as when the book is not returned within the due date.

Each book and member card will have a unique barcode. The system will be able to read barcodes from books and members' library cards.

Member: All members can search the catalog, as well as check-out, reserve, renew, and return a book.

System: Mainly responsible for sending notifications for overdue books, canceled reservations, etc.

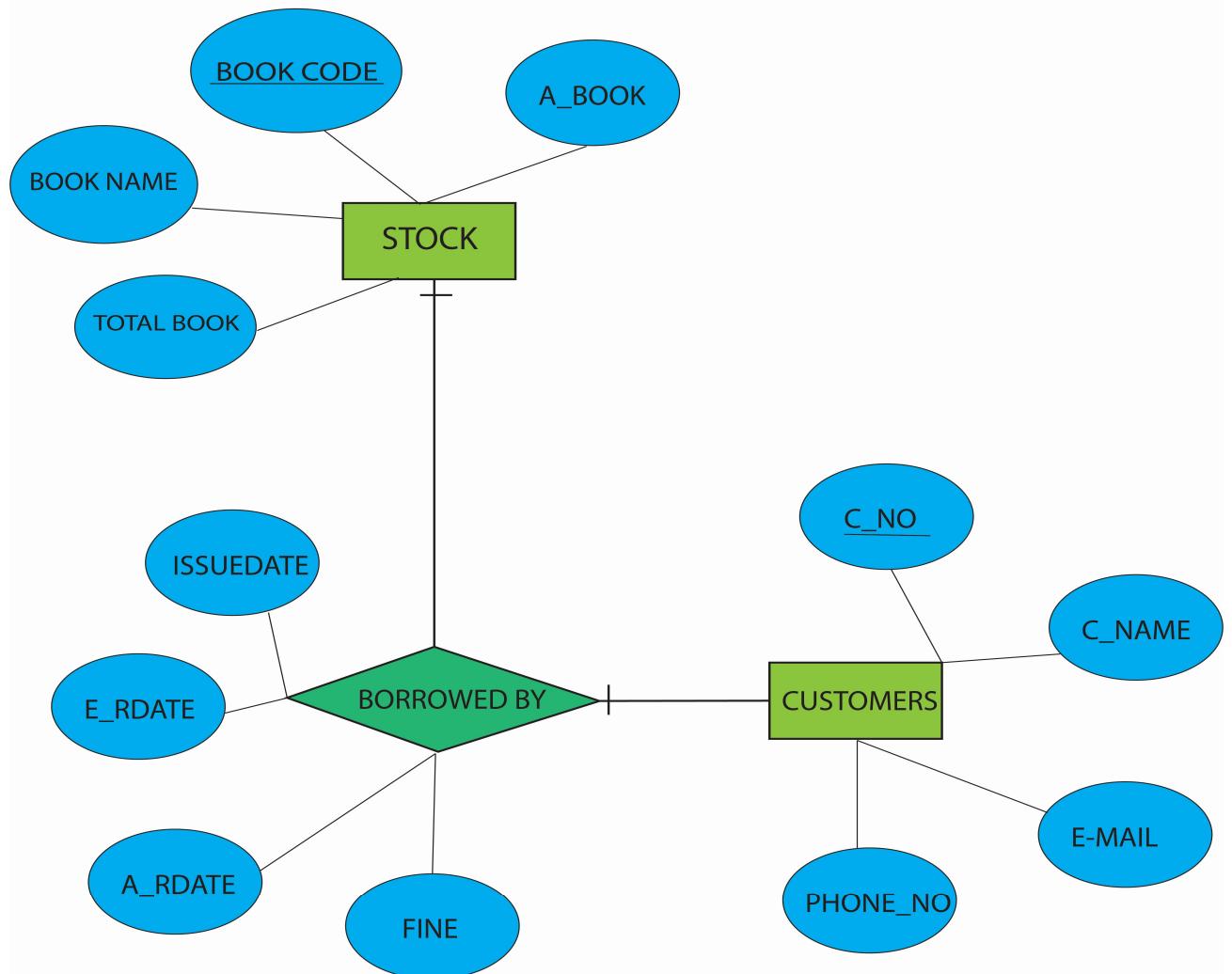
SOME FEATURES ARE-

- Add/Remove/Edit book:** To add, remove or modify a book or book item.

- Register new account/cancel membership: To add a new member or cancel the membership of an existing member.
- Check-out book: To borrow a book from the library.
- Renew a book: To reborrow an already checked-out book.
- Return a book: To return a book to the library which was issued to a member

ER DIAGRAM-

E-R DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM



ER DIAGRAM INTO TABLES-

STOCK			
<u>BOOK CODE</u>	BOOK NAME	TOTAL BOOK	AVAILABLE BOOK

CUSTOMER			
<u>CUSTOMER NUMBER</u>	NAME	E-MAIL	PHONE NUMBER

MAINTAIN						
<u>C_NO</u>	<u>B_CODE</u>	NAME	ISSUEDATE	E_RDATE	A_RDATE	FINE

IN THIS BOOK CODE IS INTEGER TYPE PRIMARY KEY AND CUSTOMER NUMBER IS INTEGER TYPE
PRIMARY KEY AND NAME IS VARCHAR,E-MAIL IS VARCHAR,PHONE NUMBER IS INTEGER TYPE,BOOK
NAME IS VARCHAR,TOTAL BOOK IS INTEGER TYPE,AVAILABLE BOOK IS INTEGER TYPE,ISSUDATE IS
DATE TYPE,ESTIMATE RETURNDATE IS DATE TYPE,ACTUAL RETURNDATE IS DATE TYPE,FINE IS INTEGER
TYPE ARE NON KEY ATTRIBUTES.SO,ALL TABLES ARE INDEPENDENT IN THE WORKINGSO THESE ARE
NORMALISED TILL 3NF.

CODE-

FOR CREATING STOCK TABLE:

```
import sqlite3
try:
    con=sqlite3.connect('p.db')
    create="create table stock(bcode integer primary key,bname text,tb integer,ab integer)"
    con.execute(create)
    print("table created successfully")
    con.commit()
    con.close()
except sqlite3.Error as error:
    print("Error occurred is",error)
finally:
    if(con):
        con.close()
        print("connection closed")
```

FOR CREATING CUSTOMER TABLE:

```
import sqlite3
try:
    con=sqlite3.connect('p.db')
    create="create table customers(cno integer primary key,cname text,phone integer not null, email text
not null)"
    con.execute(create)
    print("table created successfully")
    con.commit()
    con.close()
except sqlite3.Error as error:
    print("Error occurred is",error)
```

```
finally:  
    if(con):  
        con.close()  
        print("connection closed")
```

FOR INSERTING STOCK TABLE:

```
import sqlite3  
try:  
    con=sqlite3.connect('p.db')  
    cur=con.cursor()  
    q1="""insert into stock values(001,'ENGLISH',10,10);"""  
    q2="""insert into stock values(002,'HINDI',20,20);"""  
    q3="""insert into stock values(003,'PHYSICS',15,15);"""  
    q4="""insert into stock values(004,'BIOLOGY',25,25);"""  
    q5="""insert into stock values(005,'CHEMISTRY',10,10);"""  
    q6="""insert into stock values(006,'MATH',15,15);"""  
    q7="""insert into stock values(007,'SCIENCE',20,20);"""  
    q8="""insert into stock values(008,'SST',10,10);"""  
    q9="""insert into stock values(009,'ART',50,50);"""  
    q10="""insert into stock values(010,'GK',40,40);"""  
    cur.execute(q1)  
    cur.execute(q2)  
    cur.execute(q3)  
    cur.execute(q4)  
    cur.execute(q5)  
    cur.execute(q6)  
    cur.execute(q7)  
    cur.execute(q8)  
    cur.execute(q9)  
    cur.execute(q10)  
    con.commit()
```

```

    cur.close()

except sqlite3.Error as error:
    print("Error occurred is",error)

finally:
    if(con):
        con.close()
        print("connection closed")

```

FOR INSERTING CUSTOMER TABLE:

```

import sqlite3

try:
    con=sqlite3.connect('p.db')
    cur=con.cursor()

    q1="""insert into customers values(1,'Akash',9874561230,'akash@gmail.com');"""
    q2="""insert into customers values(2,'Amit',9875642301,'amit@gmaiil.com');"""
    q3="""insert into customers values(3,'Anurag',9999999999,'anurag@gmail.com');"""
    q4="""insert into customers values(4,'Abhishek',9898989898,'abhishek@gmaiil.com');"""
    q5="""insert into customers values(5,'Sneha',9879879870,'sneha@gmaiil.com');"""
    q6="""insert into customers values(6,'Aman',9869869860,'aman@gmaiil.com');"""
    q7="""insert into customers values(7,'Ashutosh',9859859850,'ashutosh@gmaiil.com');"""
    q8="""insert into customers values(8,'Avinash',9849849840,'avinash@gmaiil.com');"""
    q9="""insert into customers values(9,'Mohini',9839839830,'mohini@gmaiil.com');"""
    q10="""insert into customers values(10,'Savita',8745895620,'savita@gmail.com');"""

    cur.execute(q1)
    cur.execute(q2)
    cur.execute(q3)
    cur.execute(q4)
    cur.execute(q5)
    cur.execute(q6)
    cur.execute(q7)
    cur.execute(q8)

```

```

        cur.execute(q9)
        cur.execute(q10)
        con.commit()
        cur.close()

    except sqlite3.Error as error:
        print("Error occurred is",error)

    finally:
        if(con):
            con.close()
            print("connection closed")

```

FOR CREATING MAINTAIN TABLE:

```

import sqlite3
try:
    con=sqlite3.connect('p.db')
    create="create table maintain(cno integer,bcode integer,cname text,issuedate date,E_returndate
date,R_returndate date,fine integer,foreign key(cno) references customers,foreign key(bcode)
references stock)"
    con.execute(create)
    print("table created successfully")
    con.commit()
    con.close()

except sqlite3.Error as error:
    print("Error occurred is",error)
finally:
    if(con):
        con.close()
        print("connection closed")

```

MAIN CODE FOR RUNNING THE PROGRAM:

```
from datetime import date,timedelta,datetime
import sqlite3
try:

#RAVI VARSHNEY SEMESTER 3 DBMS PROJECT
#UNIVERSITY ROLL NO-191500642
#SECTION-C  ROLL NO-40
#TOPIC-LIBRARY MANAGEMENT SYSTEM
con=sqlite3.connect('p.db')
cur=con.cursor()
print("-----WELCOME TO LIBRARY-----")
print("PRESS 1 FOR ISSUE A BOOK")
print("PRESS 2 FOR RETURN A BOOK")
print("PRESS 3 FOR ENTER NEW USER")
print("PRESS 4 FOR ENTER NEW BOOK ")
print("PRESS 5 FOR BOOK STOCK")
print("PRESS 6 FOR CUSTOMER DETAIL")
print("PRESS 7 FOR TARNSACTIONS DETAILS")
a=int(input())
l1=[]
l2=[]
q5=cur.execute("select cno,cname from customers;")
for i in q5:
    l1.append(i[0])
    l2.append(i[1])
dct1={l1[i]:l2[i] for i in range(len(l1))}

l3=[]
l4=[]
q7=cur.execute("select bcode,bname from stock;")
for i in q7:
    l3.append(i[0])
    l4.append(i[1])
dct2={l3[i]:l4[i] for i in range(len(l3))}

# FOR ISSUE A BOOK
if a==1:
    q="""insert into maintain(cno,bcode,cname,issuedate,e_returndate) values(?,?,?,?,?);"""
    cno=int(input("Enter the id of a person:-"))

    cname=""
```

```

if cno in dct1:
    cname=dct1[cno]
    bcode=int(input("Enter the Book Code:-"))
    if bcode in dct2:
        q1=cur.execute("select ab from stock where bcode=?;",(bcode,))
        for i in q1:
            q12=cur.execute("select r_returndate from maintain where cno=? and bcode=? and r_returndate is NULL;",(cno,bcode))
            flag=0
            for r_returndate in q12:
                flag=1
                print("YOU ALREADY HAVE A SAME BOOK")
            if flag==0:
                if i[0]>0:
                    cur.execute("update stock set ab=ab-1 where bcode=?;",(bcode,))
                    issuedate=date.today()
                    e_returndate=date.today()+timedelta(12)
                    datatuple=(cno,bcode,cname,issuedate,e_returndate)
                    cur.execute(q,datatuple)
                    cur.execute("select * from maintain where cno=? and bcode=? and cname=? and issuedate=? and e_returndate=?;",(cno,bcode,cname,issuedate,e_returndate))
                    rows=cur.fetchall()
                    for row in rows:
                        print("Customer ID:-",row[0])
                        print("BOOK CODE:-",row[1])
                        print("Customer Name:-",row[2])
                        print("Issue Date:-",row[3])
                        print("Estimate Return Date:-",row[4])
                    break
                con.commit()
            else:
                print("NOT SUFFICIENT BOOKS")
        else:
            print("NO SUCH BOOK AVAILABLE")
    else:
        print("NO SUCH USER")

```

```

# FOR RETURN A BOOK
elif a==2:
    q="""insert into maintain(cno,bcode,cname,issuedate,e_returndate,r_returndate,fine)
values(?,?,?,?,?,?)"""
    cno=int(input("Enter the id of a person:-"))

```

```

if cno in dct1:
    cname=dct1[cno]
    bcode=int(input("Enter the Book Code:-"))
    if bcode in dct2:
        q1=cur.execute("select ab from stock where bcode=?;",(bcode,))
        for i in q1:
            cur.execute("update stock set ab=ab+1 where bcode=?;",(bcode,))
        q3=cur.execute("select issuedate from maintain where cno=? and bcode=?;",(cno,bcode))
        issuedate=""
        for i in q3:
            issuedate=i[0]

        q4=cur.execute("select e_returndate from maintain where cno=? and bcode=?;",(cno,bcode))
        e_returndate=""
        for i in q4:
            e_returndate=i[0]

        q11=cur.execute("select r_returndate from maintain where cno=? and bcode=? and
issuedate=? and e_returndate=? and r_returndate is NULL;",(cno,bcode,issuedate,e_returndate))
        flag=0
        for r_returndate in q11:
            flag=1
            r_returndate=input('Enter a date in YYYY-MM-DD format:-')
            year1,month1,dt1=map(int,r_returndate.split("-"))
            f_date=date(year1,month1,dt1)
            year2,month2,dt2=map(int,e_returndate.split("-"))
            l_date=date(year2,month2,dt2)
            d=f_date-l_date
            if d.days<=0:
                fine=0
            else:
                fine=d.days*10

        datatuple=(cno,bcode,cname,issuedate,e_returndate,r_returndate,fine)
        cur.execute(q,datatuple)
        cur.execute("select * from maintain where cno=? and bcode=? and cname=? and
issuedate=? and e_returndate=? and r_returndate=? and
fine=?;",(cno,bcode,cname,issuedate,e_returndate,r_returndate,fine))
        rows=cur.fetchall()
        for row in rows:
            print("Customer ID:-",row[0])
            print("BOOK CODE:-",row[1])
            print("Customer Name:-",row[2])
            print("Issue Date:-",row[3])
            print("Estimate Return Date:-",row[4])

```

```

        print("Return Date:-",row[5])
        print("Fine:-",row[6])
        break

    cur.execute("delete from maintain where cno=? and bcode=? and cname=? and
issuedate=? and e_returndate=? and r_returndate is NULL and fine is
NULL;",(cno,bcode,cname,issuedate,e_returndate))
    con.commit()

    if flag==0:
        print("NO SUCH ENTRY")
    else:
        print("NO SUCH BOOK AVAILABLE")
    else:
        print("NO SUCH USER")

```

```

# FOR ENTER NEW USER
elif a==3:
    l1=[]
    l2=[]
    q5=cur.execute("select cno,cname from customers;")
    q6="""insert into customers(cno,cname,phone,email) values(?, ?, ?, ?);"""
    for i in q5:
        l1.append(i[0])
        l2.append(i[1])
    dct1={l1[i]:l2[i] for i in range(len(l1))}
    cno=int(input("Enter the id of a person:-"))
    cname=""
    if cno in dct1:
        print("USER ALREADY EXISTS")
    else:
        cname=input("Enter the name of the person:-")
        phone=int(input("Enter the phone number of new user:-"))
        email=input("Enter new user email:-")
        datatuple=(cno,cname,phone,email)
        cur.execute(q6,datatuple)
        con.commit()

```

```

# FOR ENTER NEW BOOK
elif a==4:
    l3=[]
    l4=[]
    q7=cur.execute("select bcode,bname from stock;")

```

```

q8="""insert into stock(bcode,bname,tb,ab) values(?, ?, ?, ?);"""
for i in q5:
    l3.append(i[0])
    l4.append(i[1])
dct2={l3[i]:l4[i] for i in range(len(l3))}
bcode=int(input("Enter the code of book:-"))
cname=""
if bcode in dct2:
    print("BOOK ALREADY EXISTS")
else:
    bname=input("Enter the name of Book:-")
    tb=int(input("Enter the total number of books:-"))
    ab=tb
    datatuple=(bcode,bname,tb,ab)
    cur.execute(q8,datatuple)
    con.commit()

# FOR BOOK STOCK
elif a==5:
    print("-----BOOK STOCK-----")
    q7=cur.execute("select * from stock;")
    rows=cur.fetchall()
    for r in range(31):
        print("-",end="")
        print()
        print("{:<1}{:<4}{:<1}{:<10}{:<1}{:<6}{:<1}{:<6}{:<1}".format(
            "|","CODE","|","BOOKNAME","|","T_BOOK","|","A_BOOK","|"))
        for r in range(31):
            print(" -",end="")
            print()
            for row in rows:
                print("{:<1}{:>4}{:<1}{:<10}{:<1}{:<6}{:<1}{:<6}{:<1}".format(
                    "|",row[0],"|",row[1],"|",row[2],"|",row[3],"|"))
                for r in range(31):
                    print(" -",end="")
                    print()
                    con.commit()

# FOR CUSTOMER DETAIL
elif a==6:
    print("-----CUSTOMER DETAIL-----")
    q7=cur.execute("select * from customers;")
    rows=cur.fetchall()
    for r in range(50):
        print("-",end="")

```

```

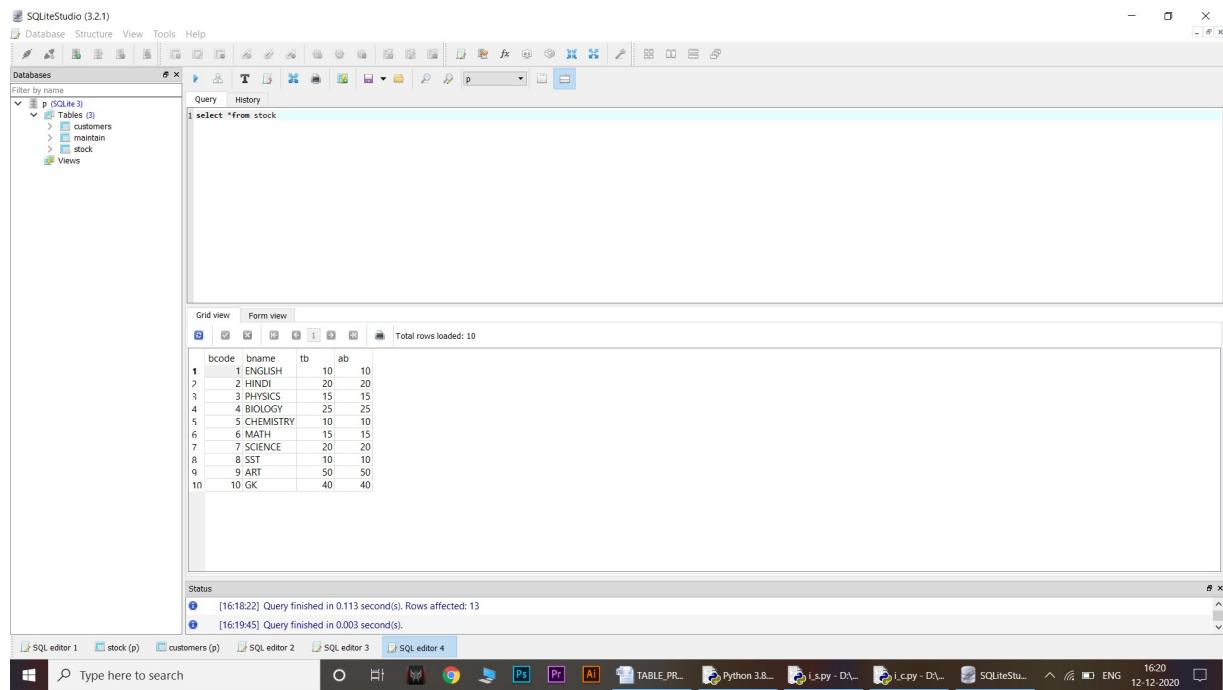
print()
print("{:<1}{:<3}{:<1}{:<10}{:<1}{:<11}{:<1}{:<21}{:<1}".format(
    "|","ID","|","NAME","|","PHONE","|","EMAIL","|")
for r in range(50):
    print("-",end="")
print()
for row in rows:
    print("{:<1}{:>3}{:<1}{:<10}{:<1}{:<11}{:<1}{:<21}{:<1}".format(
        "|",row[0],"|",row[1],"|",row[2],"|",row[3],"|"))
    for r in range(50):
        print("-",end="")
    print()
con.commit()

# FOR TARNSACTIONS DETAILS
elif a==7:
    print("-----TARNSACTIONS DETAILS-----")
    q7=cur.execute("select * from MAINTAIN;")
    rows=cur.fetchall()
    for r in range(60):
        print("-",end="")
    print()
    print("{:<1}{:<3}{:<1}{:<4}{:<1}{:<10}{:<1}{:<10}{:<1}{:<10}{:<1}{:<4}{:<1}".format(
        "|","ID","|","CODE","|","NAME","|","ISSUEDATE","|","E_R_DATE","|","RETURN DATE","|","FINE","|"))
    for r in range(60):
        print("-",end="")
    print()
    for row in rows:
        print("{:<1}{:<3}{:<1}{:<4}{:<1}{:<10}{:<1}{:<10}{:<1}{:<11}{:<1}{:<4}{:<1}".format(
            "|",row[0],"|",row[1],"|",row[2],"|",str(row[3]),"|",str(row[4]),"|",str(row[5]),"|",str(row[6]),"|"))
        for r in range(60):
            print("-",end="")
        print()
    con.commit()

else:
    print("WRONG CHOICE")
    con.commit()
    cur.close()
except sqlite3.Error as error:
    print("Error occurred is",error)
finally:
    if(con):
        con.close()
        print("connection closed")

```

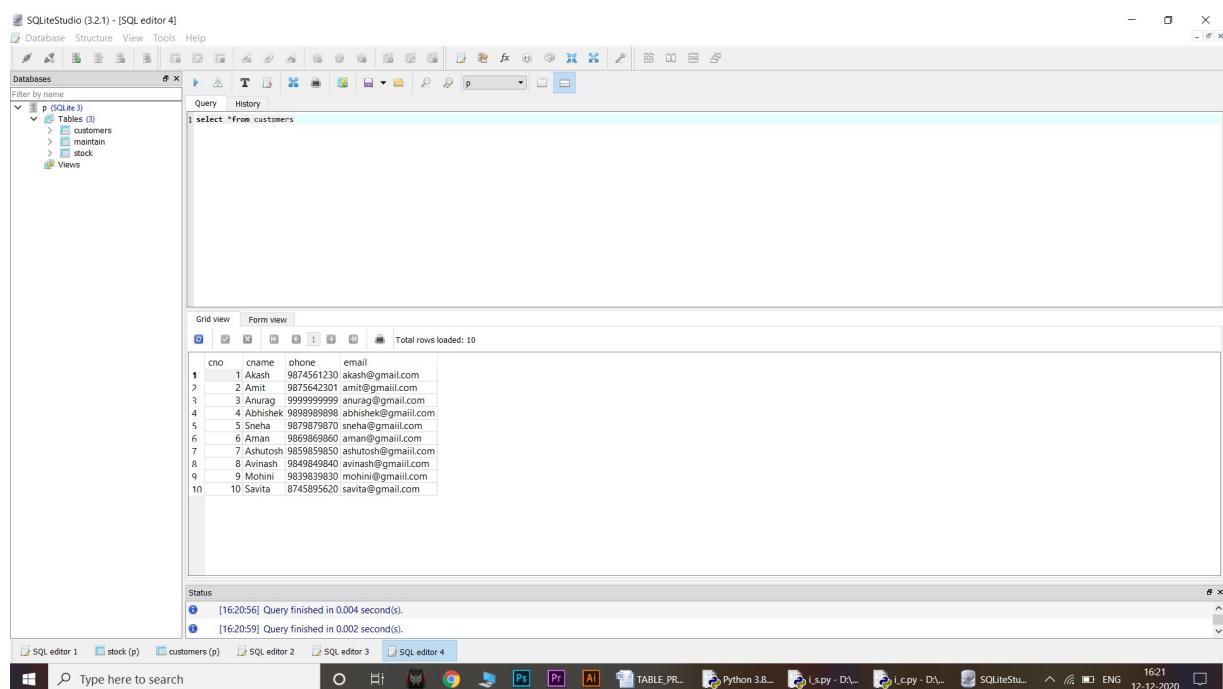
IMPLEMENTATION-



The screenshot shows the SQLiteStudio interface with the database 'p (SQLite 3)' selected. The 'Tables' tab is open, displaying the 'stock' table. A query 'select * from stock' is run, and the results are shown in a grid view:

bcode	bname	tb	ab
1	ENGLISH	10	10
2	HINDI	20	20
3	PHYSICS	15	15
4	MATHS	25	25
5	CHEMISTRY	10	10
6	MATHS	15	15
7	SCIENCE	20	20
8	SST	10	10
9	ART	50	50
10	GK	40	40

The status bar at the bottom indicates: [16:18:22] Query finished in 0.113 second(s). Rows affected: 13 [16:19:45] Query finished in 0.003 second(s).



The screenshot shows the SQLiteStudio interface with the database 'p (SQLite 3)' selected. The 'Tables' tab is open, displaying the 'customers' table. A query 'select * from customers' is run, and the results are shown in a grid view:

cno	cname	phone	email
1	Akash	9874561230	akash@gmail.com
2	Amit	9875642301	amit@gmail.com
3	Anurag	9999999999	anurag@gmail.com
4	Abhishek	9876543210	abhishek@gmail.com
5	Sneha	9878978970	sneha@gmail.com
6	Aman	9869696960	aman@gmail.com
7	Ashutosh	9859859850	ashutosh@gmail.com
8	Avinash	9849849840	avinash@gmail.com
9	Mohini	9839839830	mohini@gmail.com
10	Savita	8745895620	savita@gmail.com

The status bar at the bottom indicates: [16:20:56] Query finished in 0.004 second(s). [16:20:59] Query finished in 0.002 second(s).

```
*Python 3.8.1 Shell*
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\RAVI\sem 3\DBMS\project\mainDEmo1.py =====
-----WELCOME TO LIBRARY-----
PRESS 1 FOR ISSUE A BOOK
PRESS 2 FOR RETURN A BOOK
PRESS 3 FOR ENTER NEW USER
PRESS 4 FOR ENTER NEW BOOK
PRESS 5 FOR BOOK STOCK
PRESS 6 FOR CUSTOMER DETAIL
PRESS 7 FOR TARNSACTIONS DETAILS
|
```

FOR ENTERING 1:

```
* Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\RAVI\sem 3\DBMS\project\mainDEmo1.py =====
-----WELCOME TO LIBRARY-----
PRESS 1 FOR ISSUE A BOOK
PRESS 2 FOR RETURN A BOOK
PRESS 3 FOR ENTER NEW USER
PRESS 4 FOR ENTER NEW BOOK
PRESS 5 FOR BOOK STOCK
PRESS 6 FOR CUSTOMER DETAIL
PRESS 7 FOR TARNSACTIONS DETAILS
1
Enter the id of a person:-1
Enter the Book Code:-1
Customer ID: 1
BOOK CODE: 1
Customer Name:- Akash
Issue Date:- 2020-12-12
Estimate Return Date:- 2020-12-24
connection closed
>>>
```

FOR ENTERING 2:

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\RAVI\sem 3\DBMS\project\mainDemol.py =====
-----WELCOME TO LIBRARY-----
PRESS 1 FOR ISSUE A BOOK
PRESS 2 FOR RETURN A BOOK
PRESS 3 FOR ENTER NEW USER
PRESS 4 FOR ENTER NEW BOOK
PRESS 5 FOR BOOK STOCK
PRESS 6 FOR CUSTOMER DETAIL
PRESS 7 FOR TARNSACTIONS DETAILS
2
Enter the id of a person:-1
Enter the Book Code:-1
Enter the date in YYYY-MM-DD format:-2021-1-1
Customer Id:- 1
BOOK CODE:- 1
Customer Name:- Akash
Issue Date:- 2020-12-12
Estimate Return Date:- 2020-12-24
Return Date:- 2021-1-1
Fine:- 80
connection closed
>>> |
```

The screenshot shows a Windows desktop environment with a Python 3.8.1 Shell window open. The window title is 'Python 3.8.1 Shell'. The code entered is a script for a library management system. It prompts the user to enter a customer ID (1), a book code (1), and a date (2021-1-1). It then displays the customer details (Akash, Issue Date: 2020-12-12, Estimate Return Date: 2020-12-24, Return Date: 2021-1-1, Fine: 80). Finally, it prints 'connection closed'. The status bar at the bottom right of the shell window shows 'Ln: 25 Col: 4'.

FOR ENTERING 3:

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\RAVI\sem 3\DBMS\project\mainDemol.py =====
-----WELCOME TO LIBRARY-----
PRESS 1 FOR ISSUE A BOOK
PRESS 2 FOR RETURN A BOOK
PRESS 3 FOR ENTER NEW USER
PRESS 4 FOR ENTER NEW BOOK
PRESS 5 FOR BOOK STOCK
PRESS 6 FOR CUSTOMER DETAIL
PRESS 7 FOR TARNSACTIONS DETAILS
3
Enter the id of a person:-12
Enter the name of the person:-Lakhan
Enter the phone number of new user:-1234567891
Enter new user email:-lakan@gmail.com
connection closed
>>> |
```

The screenshot shows a Windows desktop environment with a Python 3.8.1 Shell window open. The window title is 'Python 3.8.1 Shell'. The code entered is a script for a library management system. It prompts the user to enter a customer ID (12), a name (Lakhan), a phone number (1234567891), and an email address (lakan@gmail.com). Finally, it prints 'connection closed'. The status bar at the bottom right of the shell window shows 'Ln: 19 Col: 4'.

FOR ENTERING 4:

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\RAVI\sem 3\DBMS\project\mainDemo1.py =====
-----WELCOME TO LIBRARY-----
PRESS 1 FOR ISSUE A BOOK
PRESS 2 FOR RETURN A BOOK
PRESS 3 FOR ENTER NEW USER
PRESS 4 FOR ENTER NEW BOOK
PRESS 5 FOR BOOK STOCK
PRESS 6 FOR CUSTOMER DETAIL
PRESS 7 FOR TARNSACTIONS DETAILS
4
Enter the code of book:-15
Enter the name of Book:-PSYCHOLOGY
Enter the total number of books:-15
connection closed
>>>
```

FOR ENTERING 5:

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\RAVI\sem 3\DBMS\project\mainDemo1.py =====
-----WELCOME TO LIBRARY-----
PRESS 1 FOR ISSUE A BOOK
PRESS 2 FOR RETURN A BOOK
PRESS 3 FOR ENTER NEW USER
PRESS 4 FOR ENTER NEW BOOK
PRESS 5 FOR BOOK STOCK
PRESS 6 FOR CUSTOMER DETAIL
PRESS 7 FOR TARNSACTIONS DETAILS
5
-----BOOK STOCK-----
|CODE|BOOKNAME|T_BOOK[A_BOOK]
| 1 |ENGLISH |10  |10 |
| 2 |HINDI  |20  |20 |
| 3 |PHYSICS|15  |15 |
| 4 |BIOLOGY|25  |25 |
| 5 |CHEMISTRY|10  |10 |
| 6 |MATH   |15  |15 |
| 7 |SCIENCE|20  |20 |
| 8 |SST    |10  |10 |
| 9 |ART    |50  |50 |
| 10 |GRT   |40  |40 |
| 15 |PSYCHOLOGY|15  |15 |
connection closed
>>> |
```

FOR ENTERING 6:

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\RAVI\sem 3\DBMS\project\mainDemo1.py =====
-----WELCOME TO LIBRARY-----
PRESS 1 FOR ISSUE A BOOK
PRESS 2 FOR RETURN A BOOK
PRESS 3 FOR ENTER NEW USER
PRESS 4 FOR ENTER NEW BOOK
PRESS 5 FOR BOOK STOCK
PRESS 6 FOR CUSTOMER DETAIL
PRESS 7 FOR TARNSACTIONS DETAILS
6
-----CUSTOMER DETAIL-----
|ID |NAME |PHONE |EMAIL |
|1|Akash |9874561230 |akash@gmail.com |
|2|Amit |987542301 |amit@gmail.com |
|3|Anurag |9999999999 |anurag@gmail.com |
|4|Abhishek |9898989899 |abhishek@gmail.com |
|5|Sneha |9879879879 |sneha@gmail.com |
|6|Aman |9869696966 |aman@gmail.com |
|7|Ashutosh |9859859858 |ashutosh@gmail.com |
|8|Avinash |9849849840 |avinash@gmail.com |
|9|Mohni |9839839830 |mohni@gmail.com |
|10|Savita |9874595620 |savita@gmail.com |
|12|Lakan |1234567891 |lakan@gmail.com |
connection closed
>>> |
```

FOR ENTERING 7:

```
Python 3.8.1 Shell
File Edit Shell Debug Options Window Help
Python 3.8.1 (tags/v3.8.1:1b293b6, Dec 18 2019, 23:11:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:\RAVI\sem 3\DBMS\project\mainDemo1.py =====
-----WELCOME TO LIBRARY-----
PRESS 1 FOR ISSUE A BOOK
PRESS 2 FOR RETURN A BOOK
PRESS 3 FOR ENTER NEW USER
PRESS 4 FOR ENTER NEW BOOK
PRESS 5 FOR BOOK STOCK
PRESS 6 FOR CUSTOMER DETAIL
PRESS 7 FOR TARNSACTIONS DETAILS
7
-----TARNSACTIONS DETAILS-----
|ID |CODE|NAME |ISSUEDATE |E_R_DATE |RETURN DATE|FINE|
|1| 1 |Akash |2020-12-12|2020-12-24|2021-1-1 |80 |
|2| 15 |Amit |2020-12-12|2020-12-24|None |None|
|2| 10 |Amit |2020-12-12|2020-12-24|None |None|
|4| 13 |Abhishek |2020-12-12|2020-12-24|None |None|
|2| 16 |Amit |2020-12-12|2020-12-24|2020-12-12 |0 |
|6| 19 |Aman |2020-12-12|2020-12-24|2021-2-1 |390 |
|1| 15 |Akash |2020-12-12|2020-12-24|None |None|
connection closed
>>> |
```