Problem Statement:

In library management, there is a database managed by the central government keeping a track of the members of the library, the books and the issue of the books.

The database maintains the records of all members of the central library, the books that are available to the members and manages the issuing and return of the books being borrowed.

The database ‘Books’ contains the crucial details of the books such as the ISBN number, book name, book author, publisher and edition.

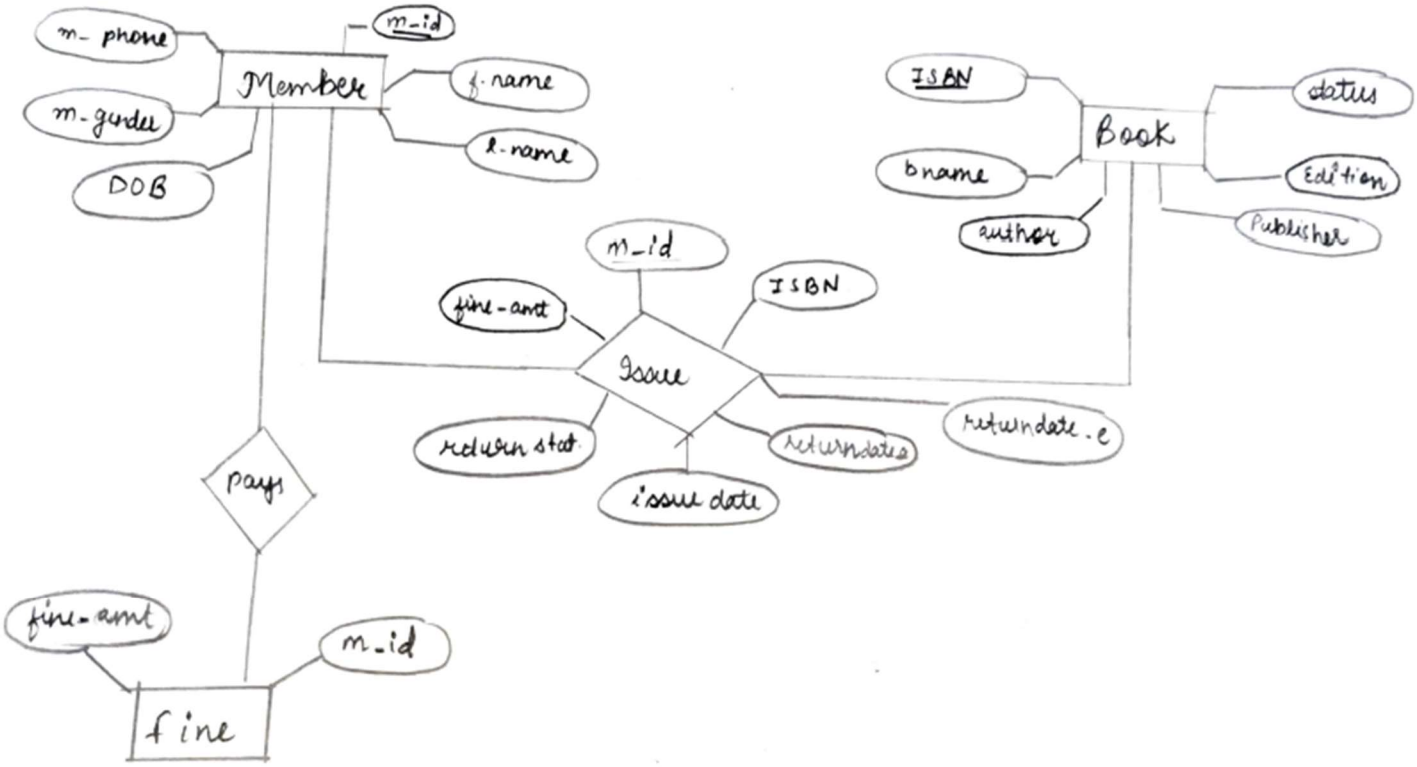
There is a database for the ‘Members’ with their information stored. A member identification number (mid) is assigned by the library to the person and other necessary information is collected by the library.

The library ‘Issues’ a book for a maximum of 21 days before either returning or re-issuing the book (for another 14 days).

If not returned or re-issued before the end date, a penalty will be imposed on the member according to the number of extra days. (Rs 30 per day).If the balance amount (fine amount) is 0, another issue of the book will be allowed, otherwise, not.

To avoid overlapped issuing of books, status of the book is maintained (T for Taken and F for Free).

ER MODEL


m_phone
m_id
Member
f_name
m_gender
l_name
DOB
ISBN
status
Book
bname
Edition
author
Publisher
m_id
fine_amt
ISBN
Issue
returnstat
returndate
returndate_e
issue date
pays
fine_amt
m_id
fine

Relational Model :

Members (m_id, fname, lname, DOB, mgender, mphone);

Books (ISBN, bname, author, publishers, edition, status);

Issue (m_id, ISBN, issue_date, returndate_e, returndate_a, fine_amt, return_stat);

Functional Dependencies:

m_id → fname, lname, DOB, mgender, mphone, fine_amt

ISBN → bname, author, publisher, edition, status

m_id, **ISBN**, **issue_date** → returndate_e, returndate_a, return_stat

Normalized Relational Model

All the functional dependencies satisfy 1st Normal Form , 2nd Normal Form , 3rd Normal Form and BCNF. Therefore, our Normalised Relational Model is as follows

Members (m_id, fname, lname, DOB, mgender, mphone);

Books (ISBN, bname, author, publisher, edition, status);

Issue (m_id, ISBN, issue_date, returndate_e, returndate_a, return_stat);

Fine (m_id, fine_amt);

SQL QUERIES USED

MEMBERS:

```
create table MEMBERS (
m_id int primary key,
fname varchar(25),
lname varchar(35),
DOB date,
mgender varchar(10),
mphone bigint,
check (floor(datediff('2022-01-01', DOB)/365)>13)
);
```

BOOKS:

```
Create table BOOKS (
ISBN int primary key,
bname varchar(125),
Author varchar(55),
Publisher varchar(45),
Edition int
Status char(1) default 'f',
);
```

ISSUE:

```
create table Issue (
m_id int,
ISBN int,
issue_date date,
return_date_e date default '0000-00-00',
return_date_a date default '0000-00-00',
return_stat varchar(10) default="NR"
primary key (m_id, ISBN, issue_date),
foreign key (m_id) references MEMBERS(m_id) on delete
cascade,
foreign key (ISBN) references BOOKS(ISBN) on delete cascade
);
```

FINE:

```
create table FINE (
m_id int primary key,
fine_amt int default 0,
foreign key(m_id) references MEMBERS(m_id) on delete
cascade
);
```

TRIGGER:

When A book is issued, change the status of the book:

```
create trigger change_stat before insert on issue
for each row update books b set
b.status='t'
where b.ISBN=new.ISBN;
```

SQL SCHEMAS

```
mysql> describe members;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| m_id    | int         | NO   | PRI | NULL    |       |
| fname   | varchar(25) | YES  |     | NULL    |       |
| lname   | varchar(35) | YES  |     | NULL    |       |
| DOB     | date        | YES  |     | NULL    |       |
| mgender | varchar(10) | YES  |     | NULL    |       |
| mphone  | bigint      | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
6 rows in set (0.03 sec)
```

```
mysql> describe books;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| ISBN      | int          | NO   | PRI | NULL    |       |
| bname     | varchar(125) | YES  |     | NULL    |       |
| author    | varchar(55)  | YES  |     | NULL    |       |
| publisher | varchar(45)  | YES  |     | NULL    |       |
| edition   | int          | YES  |     | NULL    |       |
| Status    | char(1)      | YES  |     | f       |       |
+-----------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

```
mysql> describe issue;
+---------------+-------------+------+-----+------------+-------+
| Field         | Type        | Null | Key | Default    | Extra |
+---------------+-------------+------+-----+------------+-------+
| m_id          | int         | NO   | PRI | NULL       |       |
| ISBN          | int         | NO   | PRI | NULL       |       |
| issue_date    | date        | NO   | PRI | NULL       |       |
| return_date_e | date        | YES  |     | 0000-00-00 |       |
| return_date_a | date        | YES  |     | 0000-00-00 |       |
| return_stat   | varchar(10) | YES  |     | NR         |       |
+---------------+-------------+------+-----+------------+-------+
6 rows in set (0.00 sec)
```

```
mysql> describe fine;
+----------+------+------+-----+---------+-------+
| Field    | Type | Null | Key | Default | Extra |
+----------+------+------+-----+---------+-------+
| m_id     | int  | NO   | PRI | NULL    |       |
| fine_amt | int  | YES  |     | 0       |       |
+----------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

PYTHON PROGRAM

```
import mysql.connector
from datetime import date
from datetime import datetime
from datetime import timedelta
import random
import time

mydb=mysql.connector.connect(host="localhost", user="root",
password="saanjysh", database="dbmsproject")

cur=mydb.cursor()

def memnew():
    insert_stmt="Insert into MEMBERS values (%s, %s, %s, %s, %s, %s)"
    memid=int(input("Member Identification Number(m_id): "))
    fname=input("First Name: ")
    lname=input("Last Name: ")
    dobinp=input("Date Of Birth (yyyy-mm-dd): ")
    dob= datetime.strptime(dobinp, "%Y-%m-%d")
    gen=input("Gender: ")
    phone=int(input("Phone Number: "))
    data=(memid, fname, lname, dob, gen, phone)

    try:
        cur.execute(insert_stmt, data)
        mydb.commit()
        print("Data Inserted")
    except:
        mydb.rollback()
        print("Data Not Inserted")
```

```
def booknew():
    insert_stmt="Insert into BOOKS (isbn, bname, author, publisher, edition)
values (%s, %s, %s, %s, %s)"
    isbn=int(input("Enter ISBN of book: "))
    name=input("Name of the Book: ")
    author=input("Author: ")
    pub=input("Publisher: ")
    ed=int(input("Edition: "))
    data=(isbn, name, author, pub, ed)

    try:
        cur.execute(insert_stmt, data)
        mydb.commit()
        print("Data Inserted")
    except:
        mydb.rollback()
        print("Data Not Inserted")

def payfine():
    mid=int(input("Membership Identification Number(m_id): "))
    cur.execute("Select m_id, fine_amt from fine")
    famt=cur.fetchall()
    l=[]
    for i in famt:
        l.append(i[0])
    if mid in l:
        cur.execute("Select m_id, fine_amt from fine where m_id="+str(mid))
        famt=cur.fetchall()
        for i in famt:
            fineamt=i[1]
        print("Total fine to pay:",fineamt)
        if fineamt==0:
            print("No Fine to Pay")
```

```
        else:
          paying=int(input("Fine amount you will be paying: "))
          if paying>fineamt:
                print("This amount can not be accepted")
          else:
             try:
                cnum=int(input("Card Number: "))
                cname=input("Name on Card: ")
                ccvv= int(input("CVV: "))
                randpin=random.randint(10000, 99999)
                print("MESSAGE: Pin Generated:", randpin)
                pinent=int(input("Enter Pin sent in the message in registered
number: "))
                time.sleep(2)
                if pinent==randpin:
                   cur.execute("Update fine set fine_amt=fine_amt-
"+str(paying)+" where m_id="+str(mid))
                   mydb.commit()
                   print("Transaction Successfull")
                else:
                   print("Transaction Unsuccessfull")
             except:
                mydb.rollback()
                print("Trasaction Unsuccessfull ")
    else:
       print("No Fine to Pay")
```

```
def viewinfo():
    mid=int(input("Membership Identification Number(m_id): "))
    cur.execute("Select * from Members")
    mems = cur.fetchall()
    for i in mems:
        if i[0]==mid:
            print("Member Identification Number:", i[0])
            print("Name:", i[1], i[2])
            print("Date of Birth:", i[3])
            print("Gender:", i[4])
            print("Phone Number:", i[5])


def namechange():
    l=[];
    mid=int(input("Membership Identification Number(m_id): "))
    cur.execute("Select m_id, fname, lname from Members")
    mems = cur.fetchall()
    for i in mems:
        l.append(i)
    fnnew=input("First Name: ")
    lnnew=input("Last Name: ")
    for i in l:
        if i[0]==mid:
            stmt="Update MEMBERS set fname=%s, lname=%s where
m_id=%s"
            data=(fnnew, lnnew, mid)
            try:
                cur.execute(stmt, data)
                mydb.commit()
                print("Name Updated Successfully")
            except:
                mydb.rollback()
                print("Update Not Possible")
```

```
def numchange():
    l=[];
    mid=int(input("Membership Identification Number(m_id): "))
    cur.execute("Select m_id, mphone from Members")
    mems = cur.fetchall()
    for i in mems:
        l.append(i)
    numnew=input("New Phone Number: ")
    for i in l:
        if i[0]==mid:
            stmt="Update MEMBERS set mphone=%s where m_id=%s"
            data=(numnew, mid)
            try:
                cur.execute(stmt, data)
                mydb.commit()
                print("Phone Number Updated Successfully")
            except:
                mydb.rollback()
                print("Update Not Possible")

def bookissue():
    numbook=0
    mid=int(input("Membership Identification Number(m_id): "))
    isbn=int(input("ISBN Number: "))
    cur.execute("Select m_id, count(*) from issue i where i.return_stat='NR'
group by m_id having i.m_id="+str(mid))
    countl=cur.fetchall()
    for i in countl:
        numbook=i[1]
    if numbook>=3:
        print("Maximum limit of issues attained")
    else:
        cur.execute("Select Status from books where isbn="+str(isbn))
        bstat=cur.fetchall()
        status= bstat[0][0]
```

```
        if status=='t':
            print("Book can not be issued")
            print("Please select another book")
        else:
            isda=input("Date of Issue (yyyy-mm-dd): ")
            issdate= datetime.strptime(isda, "%Y-%m-%d").date()
            retdatee = issdate + timedelta(days=21)
            stmt=("Insert into issue(m_id, ISBN, issue_date, return_date_e)"
            "values (%s, %s, %s, %s)")
            data=(mid, isbn, issdate, retdatee)
            try:
                cur.execute(stmt, data)
                mydb.commit()
                print("Book Issued Successfully")
            except:
                mydb.rollback()
                print("Book can not be issued")
                print("Please select another book")

def bookreturn():
    mid=int(input("Membership Identification Number(m_id): "))
    cur.execute("Select ISBN,bname,issue_date, return_date_e from issue i
natural join books b where i.return_stat='NR' and b.status='t'and
i.m_id="+str(mid))
    info=cur.fetchall()
    l=[]
    print("Books Currently Issued:")
    print("ISBN\t\t Book Name")
    for i in info:
        print(i[0],"\t", i[1])
        l.append(i[0])
    retisbn=int(input("ISBN of book to return: "))
    for i in info:
        if i[0]==retisbn:
```

```
            retdate_e =i[3]
            issdate=i[2]
            print("Issue Date: ",issdate)
            break;
    if retisbn in l:
        retdate=input("Return Date: ")
        stmt=("Update issue set return_date_a=%s, return_stat='R' where
m_id=%s and isbn=%s and issue_date=%s")
        data=(retdate, mid, retisbn, issdate)
        try:
            cur.execute(stmt, data)
            mydb.commit()
            cur.execute("Update books set status='f' where isbn="+str(retisbn))
            mydb.commit()
            print("Book Returned Successfully")
        except:
            mydb.rollback()
            print("Unexpected Error!")
            print("Please try again later")
        retdate= datetime.strptime(retdate, "%Y-%m-%d").date()
        if (retdate>retdate_e):
            print("Fine is Imposed for Late Return")
            delta=(retdate-retdate_e).days
            fine=30*delta
            print("Fine Value: ", fine)
            try:
                cur.execute("Insert into Fine values ("+str(mid)+", "+str(fine)+")")
                mydb.commit()
            except:
                mydb.rollback()
                cur.execute("Update fine set fine_amt=fine_amt+"+str(fine)+"
where m_id="+str(mid))
                mydb.commit()
    else:
        print("ISBN Entered Invalid")
```

```
def bookstat():
    print("Books Available:")
    print("ISBN\t\t Book Name")
    cur.execute("Select ISBN, bname from books")
    l=[]
    books=cur.fetchall()
    for i in books:
        print(i[0],"\t", i[1])
        l.append(i[0])
    isbn=int(input("ISBN of book: "))
    if isbn in l:
        cur.execute("Select isbn, status from books where isbn="+str(isbn))
        statusl=cur.fetchall()
        for i in statusl:
            status=i[1]
        if status=='t':
            print("Book is Taken")
        else:
            print("Book is Available")
    else:
        print("ISBN Entered Invalid")

ch=1;
while ch==1:
    print("WELCOME TO PERKINSONS LIBRARY\n")
    print("Main Menu:")
    print("MAINTENANCE\n\t1. New Member\n\t2. New Book Entry\n\t3.
Fine Payment\n")
    print("MEMBERS\n\t4. View Information\n\t5. Change Name\n\t6.
Change Number")
    print("BOOKS\n\t7. Book Issue\n\t8. Book Return\n\t9. Book Status")
    print("10. Exit")
    choice=int(input("Enter choice (1-9): "))
    if choice==1:
```

```
        memnew()
    elif choice==2:
        booknew()
    elif choice==3:
        payfine()
    elif choice==4:
        viewinfo()
    elif choice==5:
        namechange()
    elif choice==6:
        numchange()
    elif choice==7:
        bookissue()
    elif choice==8:
        bookreturn()
    elif choice==9:
        bookstat()
    elif choice==10:
        print("Thank You")
        print("We hope to see you again")
        break;
    else:
        print("Incorrect choice")
    print("Do You wish to continue?")
    ch=int(input("1 for Yes, 0 for No: "))
```

```
WELCOME TO PERKINSONS LIBRARY

Main Menu:
MAINTENANCE
        1. New Member
        2. New Book Entry
        3. Fine Payment

MEMBERS
        4. View Information
        5. Change Name
        6. Change Number
BOOKS
        7. Book Issue
        8. Book Return
        9. Book Status
10. Exit
Enter choice (1-9):
```

QUERIES TO ALL FUNCTIONS USED

1. Add a new member memnew()

Insert into MEMBERS values (%s, %s, %s, %s, %s, %s);

Before:

```
mysql> select * from members;
+--------+---------+--------+------------+---------+------------+
| m_id   | fname   | lname  | DOB        | mgender | mphone     |
+--------+---------+--------+------------+---------+------------+
| 435981 | Meena   | Kumari | 2004-05-17 | Female  | 8403365937 |
| 649289 | Johanna | Varkey | 2003-04-28 | Female  | 9357821225 |
| 674537 | Sumit   | Kumar  | 2000-11-22 | Male    | 9256792444 |
| 739758 | Tarun   | Mittal | 2002-12-26 | Male    | 8392658296 |
| 894918 | Mahinat | Khan   | 1999-08-12 | Female  | 8294728955 |
+--------+---------+--------+------------+---------+------------+
5 rows in set (0.01 sec)
```

Program:

```
Enter choice (1-9): 1
Member Identification Number(m_id): 849681
First Name: Jyotshna
Last Name: Jha
Date Of Birth (yyyy-mm-dd): 2003-08-17
Gender: Female
Phone Number: 7303354767
Data Inserted
Do You wish to continue?
1 for Yes, 0 for No:
```

After:

```
mysql> select * from members;
+--------+---------+--------+------------+---------+------------+
| m_id   | fname   | lname  | DOB        | mgender | mphone     |
+--------+---------+--------+------------+---------+------------+
| 435981 | Meena   | Kumari | 2004-05-17 | Female  | 8403365937 |
| 649289 | Johanna | Varkey | 2003-04-28 | Female  | 9357821225 |
| 674537 | Sumit   | Kumar  | 2000-11-22 | Male    | 9256792444 |
| 739758 | Tarun   | Mittal | 2002-12-26 | Male    | 8392658296 |
| 849681 | Jyotshna| Jha    | 2003-08-17 | Female  | 7303354767 |
| 894918 | Mahinat | Khan   | 1999-08-12 | Female  | 8294728955 |
+--------+---------+--------+------------+---------+------------+
6 rows in set (0.00 sec)
```

2.Add a new book in the db

Insert into BOOKS (isbn, bname, author, publisher, edition) values (%s, %s, %s, %s, %s);

Before:

```
+----------+---------------------+----------------+-----------------------+---------+--------+
| ISBN     | bname               | author         | publisher             | edition | Status |
+----------+---------------------+----------------+-----------------------+---------+--------+
| 28491246 | Equations on Paper  | Verditri Mathur| HGKJ Publishers       |       1 | f      |
| 35768590 | Concept of Physics  | Kunandar Thakur| RSK Publishers        |       4 | f      |
| 51895189 | The Question of Life| Niladri James  | Goyal Publishing House|       2 | f      |
| 53785795 | Tom Cruise          | Tommy Hilfiger | JK Publishers         |       1 | f      |
| 89471049 | Tales of Nirvana    | Philip Jones   | RSK Publishers        |       1 | f      |
+----------+---------------------+----------------+-----------------------+---------+--------+
5 rows in set (0.01 sec)
```

Program:

```
Enter choice (1-9): 2
Enter ISBN of book: 42781941
Name of the Book: Wizard of Oz
Author: Hilli Perk
Publisher: Royal Publishing House
Edition: 3
Data Inserted
Do You wish to continue?
1 for Yes, 0 for No:
```

After:

```
mysql> select * from books;
+----------+----------------------+----------------+-----------------------+---------+--------+
| ISBN     | bname                | author         | publisher             | edition | Status |
+----------+----------------------+----------------+-----------------------+---------+--------+
| 28491246 | Equations on Paper   | Verditri Mathur | HGKJ Publishers       |       1 | f      |
| 35768590 | Concept of Physics   | Kunandar Thakur | RSK Publishers        |       4 | f      |
| 42781941 | Wizard of Oz         | Hilli Perk      | Royal Publishing House |       3 | f      |
| 51895189 | The Question of Life | Niladri James   | Goyal Publishing House |       2 | f      |
| 53785795 | Tom Cruise           | Tommy Hilfiger  | JK Publishers         |       1 | f      |
| 89471049 | Tales of Nirvana     | Philip Jones    | RSK Publishers        |       1 | f      |
+----------+----------------------+----------------+-----------------------+---------+--------+
6 rows in set (0.00 sec)
```

3.Update fine (Fine Payment)

Update fine set fine_amt=fine_amt-"+str(paying)+" where m_id="+str(mid) ;

Before:

```
mysql> select * from fine;
+--------+----------+
| m_id   | fine_amt |
+--------+----------+
| 674537 |        0 |
| 849681 |       90 |
+--------+----------+
2 rows in set (0.00 sec)
```

Program:

```
Enter choice (1-9): 3
Membership Identification Number(m_id): 849681
Total fine to pay: 90
Fine amount you will be paying: 90
Card Number: 482947291758
Name on Card: Jyotshna Jha
CVV: 782
MESSAGE: Pin Generated: 70430
Enter Pin sent in the message in registered number: 70430
Transaction Successfull
Do You wish to continue?
1 for Yes, 0 for No: 1
```

After:

```
mysql> select * from fine;
+--------+----------+
| m_id   | fine_amt |
+--------+----------+
| 674537 |        0 |
| 849681 |        0 |
+--------+----------+
2 rows in set (0.00 sec)
```

4.View info about a person

Select * from Members where m_id=%s;

Before:

```
mysql> select * from members;
+--------+----------+--------+------------+---------+------------+
| m_id   | fname    | lname  | DOB        | mgender | mphone     |
+--------+----------+--------+------------+---------+------------+
| 435981 | Meena    | Kumari | 2004-05-17 | Female  | 8403365937 |
| 649289 | Johanna  | Varkey | 2003-04-28 | Female  | 9357821225 |
| 674537 | Sumit    | Kumar  | 2000-11-22 | Male    | 9256792444 |
| 739758 | Tarun    | Mittal | 2002-12-26 | Male    | 8392658296 |
| 849681 | Jyotshna | Jha    | 2003-08-17 | Female  | 7303354767 |
| 894918 | Mahinat  | Khan   | 1999-08-12 | Female  | 8294728955 |
+--------+----------+--------+------------+---------+------------+
6 rows in set (0.00 sec)
```

Output:

```
Enter choice (1-9): 4
Membership Identification Number(m_id): 649289
Member Identification Number: 649289
Name: Johanna Varkey
Date of Birth: 2003-04-28
Gender: Female
Phone Number: 9357821225
Do You wish to continue?
1 for Yes, 0 for No: 1
```

5.Change the name of the member

Update MEMBERS set fname=%s, lname=%s where m_id=%s;

Before:

```
mysql> select * from members;
+--------+----------+--------+------------+---------+------------+
| m_id   | fname    | lname  | DOB        | mgender | mphone     |
+--------+----------+--------+------------+---------+------------+
| 435981 | Meena    | Kumari | 2004-05-17 | Female  | 8403365937 |
| 649289 | Johanna  | Varkey | 2003-04-28 | Female  | 9357821225 |
| 674537 | Sumit    | Kumar  | 2000-11-22 | Male    | 9256792444 |
| 739758 | Tarun    | Mittal | 2002-12-26 | Male    | 8392658296 |
| 849681 | Jyotshna | Jha    | 2003-08-17 | Female  | 7303354767 |
| 894918 | Mahinat  | Khan   | 1999-08-12 | Female  | 8294728955 |
+--------+----------+--------+------------+---------+------------+
6 rows in set (0.00 sec)
```

Program:

```
Enter choice (1-9): 5
Membership Identification Number(m_id): 435981
First Name: Meena
Last Name: Sahay
Name Updated Successfully
Do You wish to continue?
1 for Yes, 0 for No:
```

After:

```
mysql> select * from members;
+--------+----------+--------+------------+---------+------------+
| m_id   | fname    | lname  | DOB        | mgender | mphone     |
+--------+----------+--------+------------+---------+------------+
| 435981 | Meena    | Sahay  | 2004-05-17 | Female  | 8403365937 |
| 649289 | Johanna  | Varkey | 2003-04-28 | Female  | 9357821225 |
| 674537 | Sumit    | Kumar  | 2000-11-22 | Male    | 9256792444 |
| 739758 | Tarun    | Mittal | 2002-12-26 | Male    | 8392658296 |
| 849681 | Jyotshna | Jha    | 2003-08-17 | Female  | 7303354767 |
| 894918 | Mahinat  | Khan   | 1999-08-12 | Female  | 8294728955 |
+--------+----------+--------+------------+---------+------------+
6 rows in set (0.00 sec)
```

6.Change the phnumber of the member

Update MEMBERS set mphone=%s where m_id=%s;

Before:

```
mysql> select * from members;
+--------+----------+--------+------------+---------+------------+
| m_id   | fname    | lname  | DOB        | mgender | mphone     |
+--------+----------+--------+------------+---------+------------+
| 435981 | Meena    | Sahay  | 2004-05-17 | Female  | 8403365937 |
| 649289 | Johanna  | Varkey | 2003-04-28 | Female  | 9357821225 |
| 674537 | Sumit    | Kumar  | 2000-11-22 | Male    | 9256792444 |
| 739758 | Tarun    | Mittal | 2002-12-26 | Male    | 8392658296 |
| 849681 | Jyotshna | Jha    | 2003-08-17 | Female  | 7303354767 |
| 894918 | Mahinat  | Khan   | 1999-08-12 | Female  | 8294728955 |
+--------+----------+--------+------------+---------+------------+
6 rows in set (0.00 sec)
```

Program:

```
Enter choice (1-9): 6
Membership Identification Number(m_id): 739758
New Phone Number: 9827745812
Phone Number Updated Successfully
Do You wish to continue?
1 for Yes, 0 for No:
```

After:

```
mysql> select * from members;
+--------+---------+--------+------------+---------+------------+
| m_id   | fname   | lname  | DOB        | mgender | mphone     |
+--------+---------+--------+------------+---------+------------+
| 435981 | Meena   | Sahay  | 2004-05-17 | Female  | 8403365937 |
| 649289 | Johanna | Varkey | 2003-04-28 | Female  | 9357821225 |
| 674537 | Sumit   | Kumar  | 2000-11-22 | Male    | 9256792444 |
| 739758 | Tarun   | Mittal | 2002-12-26 | Male    | 9827745812 |
| 849681 | Jyotshna| Jha    | 2003-08-17 | Female  | 7303354767 |
| 894918 | Mahinat | Khan   | 1999-08-12 | Female  | 8294728955 |
+--------+---------+--------+------------+---------+------------+
6 rows in set (0.00 sec)
```

7.Issue a book :->

i)Count the number of books issued on each member

Insert into issue(m_id, ISBN, issue_date, return_date_e) values (%s, %s, %s, %s);

ii)Check the status of the book to be issued

Select isbn, status from books where isbn="+str(isbn)

iii) If status available, issue book and set the return date 21 days from now <used python timdelta function>

Before:

```
mysql> select * from issue;
+--------+----------+------------+--------------+--------------+-------------+
| m_id   | ISBN     | issue_date | return_date_e | return_date_a | return_stat |
+--------+----------+------------+--------------+--------------+-------------+
| 435981 | 35768590 | 2020-04-23 | 2020-05-14    | 2020-05-11    | R           |
| 674537 | 28491246 | 2020-09-30 | 2020-10-21    | 2020-10-13    | R           |
| 674537 | 35768590 | 2021-05-27 | 2021-06-17    | 2021-06-21    | R           |
| 674537 | 51895189 | 2019-11-01 | 2019-11-22    | 2019-11-13    | R           |
| 674537 | 53785795 | 2021-01-23 | 2021-02-13    | 2021-02-11    | R           |
| 849681 | 89471049 | 2022-03-21 | 2022-04-11    | 0000-00-00    | NR          |
+--------+----------+------------+--------------+--------------+-------------+
```

Program:

```
Enter choice (1-9): 7
Membership Identification Number(m_id): 894918
ISBN Number: 51895189
Date of Issue (yyyy-mm-dd): 2020-04-12
Book Issued Successfully
Do You wish to continue?
1 for Yes, 0 for No: 1
```

After:

```
mysql> select * from issue;
+--------+----------+------------+--------------+--------------+-------------+
| m_id   | ISBN     | issue_date | return_date_e | return_date_a | return_stat |
+--------+----------+------------+--------------+--------------+-------------+
| 435981 | 35768590 | 2020-04-23 | 2020-05-14    | 2020-05-11    | R           |
| 674537 | 28491246 | 2020-09-30 | 2020-10-21    | 2020-10-13    | R           |
| 674537 | 35768590 | 2021-05-27 | 2021-06-17    | 2021-06-21    | R           |
| 674537 | 51895189 | 2019-11-01 | 2019-11-22    | 2019-11-13    | R           |
| 674537 | 53785795 | 2021-01-23 | 2021-02-13    | 2021-02-11    | R           |
| 849681 | 89471049 | 2022-03-21 | 2022-04-11    | 0000-00-00    | NR          |
| 894918 | 51895189 | 2020-04-12 | 2020-05-03    | 0000-00-00    | NR          |
+--------+----------+------------+--------------+--------------+-------------+
7 rows in set (0.00 sec)
```

8.Return of a book :->

i)select all books which are not yet returned by a particular member
Select ISBN,bname,issue_date, return_date_e from issue i natural join books b
where i.return_stat='NR' and b.status='t'and i.m_id="+str(mid))

ii)In return process reset the attributes
Update issue set return_date_a=%s, return_stat='R' where m_id=%s and isbn=%s and issue_date=%s"

iii)Impose fine on late return
*Calculate fine first<python logic>
 delta=(retdate-retdate_e).days
 fine=30*delta
*note(it will first try to insert the fine if m_id is new to fine table)
  Insert into Fine values ("+str(mid)+", "+str(fine)+")
if fails in doing so (not unique value of m_id)
  Update issue set return_date_a=%s, return_stat='R' where m_id=%s and isbn=%s and issue_date=%s"

Before:

```
mysql> select * from issue;
+--------+----------+------------+--------------+--------------+-------------+
| m_id   | ISBN     | issue_date | return_date_e | return_date_a | return_stat |
+--------+----------+------------+--------------+--------------+-------------+
| 435981 | 35768590 | 2020-04-23 | 2020-05-14    | 2020-05-11    | R           |
| 674537 | 28491246 | 2020-09-30 | 2020-10-21    | 2020-10-13    | R           |
| 674537 | 35768590 | 2021-05-27 | 2021-06-17    | 2021-06-21    | R           |
| 674537 | 51895189 | 2019-11-01 | 2019-11-22    | 2019-11-13    | R           |
| 674537 | 53785795 | 2021-01-23 | 2021-02-13    | 2021-02-11    | R           |
| 849681 | 89471049 | 2022-03-21 | 2022-04-11    | 0000-00-00    | NR          |
| 894918 | 51895189 | 2020-04-12 | 2020-05-03    | 0000-00-00    | NR          |
+--------+----------+------------+--------------+--------------+-------------+
7 rows in set (0.00 sec)
```

Program:

```
Enter choice (1-9): 8
Membership Identification Number(m_id): 849681
Books Currently Issued:
ISBN             Book Name
89471049         Tales of Nirvana
ISBN of book to return: 89471049
Issue Date:  2022-03-21
Return Date: 2022-04-14
Book Returned Successfully
Fine is Imposed for Late Return
Fine Value:  90
Do You wish to continue?
1 for Yes, 0 for No:
```

After:

```
mysql> select * from issue;
+--------+----------+------------+--------------+--------------+-------------+
| m_id   | ISBN     | issue_date | return_date_e | return_date_a | return_stat |
+--------+----------+------------+--------------+--------------+-------------+
| 435981 | 35768590 | 2020-04-23 | 2020-05-14    | 2020-05-11    | R           |
| 674537 | 28491246 | 2020-09-30 | 2020-10-21    | 2020-10-13    | R           |
| 674537 | 35768590 | 2021-05-27 | 2021-06-17    | 2021-06-21    | R           |
| 674537 | 51895189 | 2019-11-01 | 2019-11-22    | 2019-11-13    | R           |
| 674537 | 53785795 | 2021-01-23 | 2021-02-13    | 2021-02-11    | R           |
| 849681 | 89471049 | 2022-03-21 | 2022-04-11    | 2022-04-14    | R           |
| 894918 | 51895189 | 2020-04-12 | 2020-05-03    | 0000-00-00    | NR          |
+--------+----------+------------+--------------+--------------+-------------+
7 rows in set (0.00 sec)
```

```
mysql> select * from fine;
+--------+----------+
| m_id   | fine_amt |
+--------+----------+
| 674537 |        0 |
| 849681 |       90 |
+--------+----------+
2 rows in set (0.00 sec)
```

9.To check the status of the book

Select status from books where isbn="+str(isbn);

Before:

```
mysql> select * from books;
+----------+----------------------+----------------+-----------------------+---------+--------+
| ISBN     | bname                | author         | publisher             | edition | Status |
+----------+----------------------+----------------+-----------------------+---------+--------+
| 28491246 | Equations on Paper   | Verditri Mathur | HGKJ Publishers       |       1 | f      |
| 35768590 | Concept of Physics   | Kunandar Thakur | RSK Publishers        |       4 | f      |
| 42781941 | Wizard of Oz         | Hilli Perk      | Royal Publishing House |       3 | f      |
| 51895189 | The Question of Life | Niladri James   | Goyal Publishing House |       2 | t      |
| 53785795 | Tom Cruise           | Tommy Hilfiger  | JK Publishers         |       1 | f      |
| 89471049 | Tales of Nirvana     | Philip Jones    | RSK Publishers        |       1 | f      |
+----------+----------------------+----------------+-----------------------+---------+--------+
6 rows in set (0.00 sec)
```

```
Enter choice (1-9): 9
Books Available:
ISBN            Book Name
28491246        Equations on Paper
35768590        Concept of Physics
42781941        Wizard of Oz
51895189        The Question of Life
53785795        Tom Cruise
89471049        Tales of Nirvana
ISBN of book: 42781941
Book is Available
Do You wish to continue?
1 for Yes, 0 for No:
```

```
Enter choice (1-9): 9
Books Available:
ISBN            Book Name
28491246        Equations on Paper
35768590        Concept of Physics
42781941        Wizard of Oz
51895189        The Question of Life
53785795        Tom Cruise
89471049        Tales of Nirvana
ISBN of book: 51895189
Book is Taken
Do You wish to continue?
1 for Yes, 0 for No:
```