Online Library Management System

Mini Project Report -Database Lab (DSE 2260)

Department of Data Science & Computer Applications

B. Tech Data Science

4th Semester – Batch: B1/B2/B3/B4

Submitted By

Name of Team member	***Registration number***
Vishnu Jilla	200968104
Siddharth Singh	200968106
Atyam V R Manoj	200968108
Madamanchi Chandra Vardhan	200968112

Mentored By

Vinayak M
Assistant Professor-Senior
DSCA, MIT

Archana H/ Shameem
Assistant Professor-Senior
DSCA, MIT

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Date: 23-05-2022

CERTIFICATE

This is to certify that the Vishnu Jilla (200968104) Siddharth Singh (200968106), Atyam V V R Manoj (200968108), Madamanchi Chandra Vardhan (200968112) have successfully executed a mini project titled “Library Management System” rightly brining for the competencies and skill sets they have gained during the course- Database Lab (DSE 2262 & DSE), thereby resulting in the culmination of this project.

Vinayak M
Assistant Professor-Senior
DSCA, MIT

Archana H / Shameem
Assistant Professor-Senior
DSCA, MIT

ABSTRACT

Manual process of keeping student records, book records, account details, and managing employees is very difficult. There are various problems faced by the student in the library such as finding a particular book, information whether a book is available or not, when would a book be available, searching for books using ISBN number etc. To eliminate such problems, a library management system has been developed. It will handle all such issues faced by the students and by its admin personnel.

Library management system is a project which aims in developing a computerized system to maintain all the daily work of the library. It maintains the information about the books present in the library, their authors, the members of library to whom books are issued, library staff etc. To store all the information in the database from where the user will place their query and get the results on the basis of their query. Only valid users will be able to access this Library Management System.

The purpose of a library management system is to operate a library with efficiency and at reduced costs. Through this, it will be easy to manage accounts and various details of particular students and employees working in the library along with the records of books. It has a facility of admin login through which the admin can monitor the whole system. It has a facility where students after logging in their accounts can see the list of books issued and its issue date and return date.

Hence, this system is being developed to help the staff of the library to maintain the library in the best way possible and also reduce human efforts.

Contents

Chapter 1

Introduction

The Library Management System is a project that manages and stores books and their information electronically according to student's needs. The system helps both students and the library to manage and keep a constant track of all the books available in the library. It allows both the admin and the student to search for the desired book. It becomes necessary for colleges to keep a continuous check on the books issued and returned. This task if carried out manually will be tedious and includes chances of mistakes. These errors are avoided by allowing the system to keep track of information such as issue date, last date to return the book and even fine information and thus there is no need to keep manual track of this information which thereby avoids chances of mistakes. Thus, this system reduces manual work to a great extent allowing smooth flow of library activities by removing chances of errors in the details.

This Library Management System will have a login page from where its user can access. This page will provide login for admin, working staff members and the students. Staff members' accounts will be managed by the admin. To access the library resources students have to register by using their user id, name, phone number, Address. After successful registration they will be provided the login facility. Students can search books by using Book ID or by author name or by title of book along with author name. After completion of this process students will be provided with book details such as where it is located by using location number and by their row and column number.

Even if a student loses a book, he can inform the library working staff members where they can make changes to their account and take appropriate actions such as fine. Admin will be able to add staff, delete staff, add students, delete students, add books, manage account details, schedule working time tables etc.

The default user for our project is admin. Our project provides the facility for the admin to login and then review all the books and members in the library database. Admin also has an option of adding another person as admin to the system.

ENTITIES

1) MEMBER
2) AUTHOR
3) BOOK
4) BOOK RECORDS
5) LIBRARIAN
6) RECOMMENDATIONS

Detailed description of them:

MEMBER (This is master table of all users having access to library)

- MemberID (Primary Key) - This is the unique ID to identify each and every user such as a student's registration number.
- Name - Provides the name of the user
- Gender
- Address
- Mobile Number

BOOK (This is the master table for all the books available in the library)

- Book ID (Primary key) - Unique ID given to each book as there could be multiple books with the same title. It will help us to distinguish them.
- Title – Provides the name of the book.
- Publisher – Provides the name of its publisher.
- Year – Provides the year the book was published.
- Availability – Let us know whether the book is available or not.
- Copies – Indicates the no. of total copies in the library.

BOOK RECORDS

- User ID
- Book ID
- Date of Issue – Provides the date when a book was issued to a user.
- Due date – Provides the date by when the book is to be returned.
- Date of return – Provides the date when the book was returned.
- Dues – Indicates if the user is to pay a due amount.
- Renewals left – Indicates the no. of books that the user can still be issued.

LIBRARIAN

- Admin ID (Primary key) – Unique ID given to each staff member in the library.
- Name – Provides the name of the admin/staff member.
- Address
- Mobile number

AUTHOR

- Book ID (Primary key)
- Author – Provides the name of the author of the particular book.

RECOMMENDATIONS

- R ID – Provides unique ID to each recommended book.
- Book ID – Indicates the ID of that book.
- Book Name – Provides the name of the book.
- Description
- User ID

Chapter 2

Synopsis

2.1 Proposed System

The library database system's main aim is to prevail over delaying and procrastinating unnecessarily while searching for the books, authors etc. by providing all the information like the availability, publishers, last issued date, the dues pending on the book, and other essentials like id's, title etc. These are readily accessible to the user within a database. Consequently, this system helps in narrowing down the options and reduces human efforts leading to easy accessing via database. It also consists of all the managing information regarding the library. Some of them are Admin names, admin id's, all the necessary mobile numbers of the staff, addresses.

This helps the librarian or the staff to retrieve and manage all the data effortlessly through the library database system.

2.2 Objectives

The Main Objectives of this project are:

- To provide a friendly environment to maintain the details of books and library members.
- To maintain an easy circulation system using computers and to provide different reports.
- The Library System is a package to be used by Libraries to improve the efficiency of Librarians, Library employees and Users.
- The system provides books catalogue and information to members and helps them decide on the books to borrow from the library.
- The Librarian can keep the books catalogue updated all the time so that the members (students and the professors) get the updated information all the time.

Chapter 3

Functional Requirements

Login Form: A Valid Username and Password
Creation and Deletion of Account: Identification number of student and Student's Name.
Creation and Deletion of Account: Identification number of student and Student's Name.
View/Issue/Return: Identification number of students.
Updating stock details**:** It must enter details regarding new book, circulars, magazines etc. The stock details are updated each and every time a new stock arrives or when an item is taken out.
Providing Identification number**:** The system provides unique identification number to each item separately. It should also provide separate identification number to its members
Maintaining Records**:** It must maintain library assets, members and staffs and keep each and every detail that are relevant for running the library smoothly.

3.1 User Registering/Login module

Two lines about module briefly and it supports functionalities- New user registration, Login, Forgot password

3.1.1 New User Registration

The user must be able to create user id and password by supplying appropriate details.

INPUT	New username, Password, phone
Processing	The system must check availability of entered user name. Password must follow criteria- minimum 9 chars, at least one capital, one number and one special character. Check for validity of phone number by prompting to enter OTP
OUTPUT	User created Successfully message / highlight the information entered which is wrong and allow to renter.

3.1.2 Login

The existing user must be able to login upon entering proper user name and password.

INPUT	username, Password
Processing	Check the user's name and password against information stored in data storage
OUTPUT	If user entered correct user name & Password Login successful and open main application menu Else Display Login not successful, retry logging in

3.1.3 Forgot password

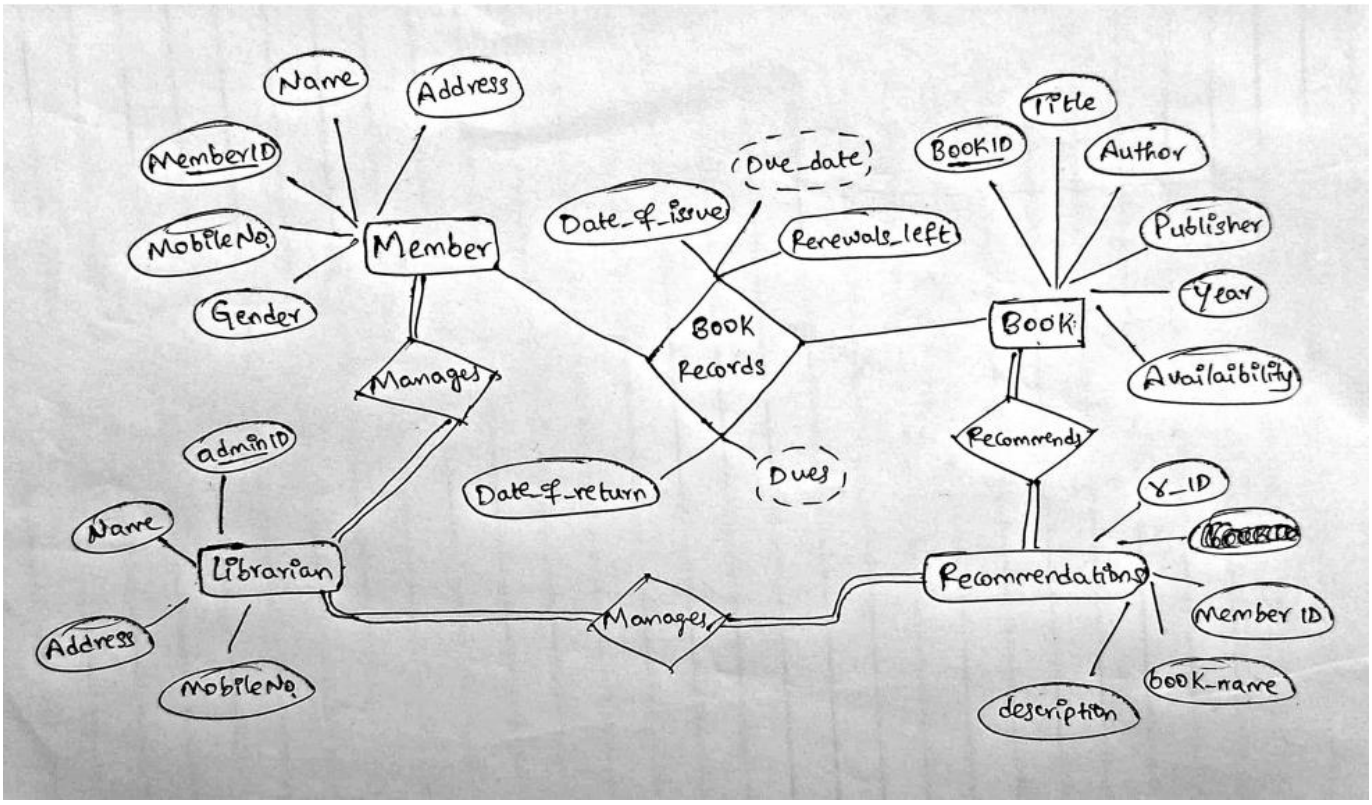
If existing user name is not bale to login, forgot password can be used to reset password.

INPUT	Prompt user to enter username, Phone
Processing	If username and corresponding phone exist in the data storage Send OTP to Phone. Prompt the user to enter OTP If OTP matching Prompt user to change password according to criteria. Else OTP not matching. Else User name and corresponding Phone not existing in the storage
OUTPUT	Password successfully changed / User name, phone not matching

Chapter 4

Detailed Design

4.1 ER Diagram



4.2 Schema Diagram

Member(MemberID, name, gender, address, mobile_no)

Librarian(adminID, name, address, mobile_no)

Author(bookID, author)

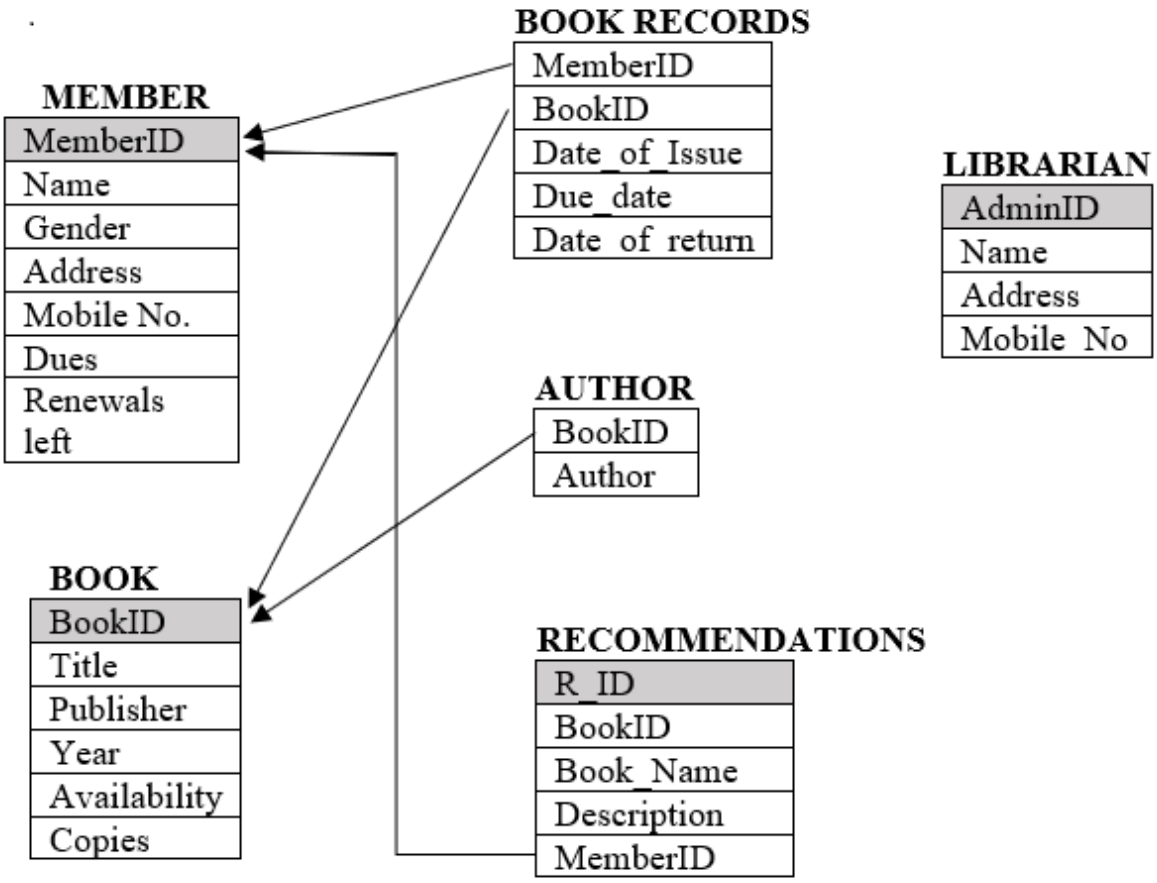
bookID is a foreign Key from Book

Book(book_id, title, publisher, year, availability, copies)

Book_Records(MemberID, bookID, date_of_issue, due_date, date_of_return, dues, renewals_left)

userID is a Foreign Key from User and bookID is Foreign Key from Book

Recommendations(r_id, bookID,book_name, description, MemberID)



4.3 Data Dictionary

MEMBER

Field	Data type	Default	Key
MemberID	VARCHAR(20)	Not NULL	Primary key
Name	VARCHAR(50)	NULL	
Gender	CHAR(1)	NULL	
Address	VARCHAR (100)	NULL	
Mobile_no	CHAR(10)	NULL	

LIBRARIAN

Field	Data type	Default	Key
AdminID	VARCHAR(20)	Not NULL	Primary Key
Name	VARCHAR(50)	NULL	
Address	VARCHAR(100)	NULL	
Mobile_no	CHAR(10)	NULL	

AUTHOR

Field	Data type	Default	Key
BookID	VARCHAR(20)	Not NULL	Foreign key
Author	VARCHAR(50)	Not NULL	

BOOK

Field	Data type	Default	Key
BookID	VARCHAR(20)	Not NULL	On delete Cascade, Primary key
Title	VARCHAR(50)	NULL	
Publisher	VARCHAR(50)	NULL	
Year	YEAR(4)		
Availability	VARCHAR(5)	Yes or No	
Copies	NUMBER(3)		

RECOMMENDATION

Field	Data type	Default	Key
R_ID	VARCHAR(20)	Not NULL	Primary Key
Book_Name	VARCHAR(50)	NULL	
Description	VARCHAR(100)	NULL	
MemberID	VARCHAR(20)	Not NULL	Foreign Key

BOOK RECORDS

Field	Data type	Default	Key
MemberID	VARCHAR(20)	Not NULL	Foreign Key
BookID	VARCHAR(20)	Not NULL	Foreign Key
Date_Of_Issue	DATE		
Due_Date	DATE	>Date_Of_Issue	
Date_Of_Return	DATE	>=Due_Date	
Dues	NUMBER(10)	0	
Renewals_Left	NUMBER(3)	4	

4.4 Relational Model Implementation

create table member(

memberID varchar(20) constraint MemberID_is_primary primary key,

Name varchar(50),

Gender char(1) check(Gender in ('M','F')),

Address varchar(100),

Mobile_no number(10),

dues number(10) default 0,

renewals_left number(3) default 4);

create table book(

bookID varchar(20) primary key,

title varchar(50),

publisher varchar(50),

availability varchar(5) constraint availability_yes_or_no check(availability in ('Yes','No')),

copies number(3));

create table librarian(

adminID varchar(20) constraint adminID_is_primary primary key,

name varchar(50),

Address varchar(50),

mobile_no char(10));

create table author(

bookID varchar(20) references book(bookID),

author varchar(50));

create table book_records(

memberID varchar(20) references member(memberID),

bookid varchar(20) references book(bookID),

date_of_issue date,

due_date date,

date_of_return date,

check((due_date > date_of_issue) and (date_of_return >= due_date)));

create table **recommendation**(

r_id varchar2 (20) primary key,

memberid varchar(20) not null references member(memberID),

book_name varchar(50),

description varchar(100));

4.5 Queries

INSERTING DATA INTO TABLES

- **MEMBER TABLE**

```
INSERT INTO MEMBER
VALUES('1','DEEPESH','M',NULL,9496875601,default,default);

INSERT INTO MEMBER
VALUES('2','PRIYANSH','M',NULL,9496875602,default,default);

INSERT INTO MEMBER
VALUES('3','AKASH',’M’,NULL,9496875603,default,default);

INSERT INTO MEMBER
VALUES('4','SWATI','F',NULL,9496875604,default,default);

INSERT INTO MEMBER
VALUES('5','BOSE','M',NULL,9496875605,default,default);

INSERT INTO MEMBER
VALUES('6','PRATIKHYA','F',NULL,9496875606,default,default);

INSERT INTO MEMBER
VALUES('7','DHRUTI','F',NULL,9496875607,default,default);
```

- **BOOK TABLE**

```
INSERT INTO BOOK VALUES('100', 'ARTIFICIAL INTELLIGENCE',
'PEARSON', 'YES', 3);

INSERT INTO BOOK VALUES('101', 'MATHEMATICS FOR DATA
SCIENCE', ' SAINZ', 'NO', 9);

INSERT INTO BOOK VALUES('104', 'MACHINE LEARNING ',
'HARPERCOLLINS','YES', 6);

INSERT INTO BOOK VALUES('106', 'BIOLOGY ', 'HAMILTON', 'YES',8);

INSERT INTO BOOK VALUES('116', 'AUTOMOTIVE ENGINES ',
'WILEY', 'YES', 8);

INSERT INTO BOOK VALUES('112', 'THERMODYNAMICS ', 'VETTEL',
'YES', 12);
```

- **LIBRARIAN TABLE**

INSERT INTO LIBRARIAN VALUES('126789','ROSS', 'ABCD', '6542387188');

INSERT INTO LIBRARIAN VALUES('126749','RACHEL', 'CCCD', '7542386188');

INSERT INTO LIBRARIAN VALUES('126759','CHANDLER', 'AFCD', '4542386188');

INSERT INTO LIBRARIAN VALUES('126739','JOEY', 'MMAT', '8972386188');

INSERT INTO LIBRARIAN VALUES('126888','MONICA', 'LAPD', '9992386188');

INSERT INTO LIBRARIAN VALUES('126889','AMBER HEARD', 'LAPD', '9992386188');

- **AUTHOR TABLE**

INSERT INTO AUTHOR VALUES('100','KANYE');

INSERT INTO AUTHOR VALUES('101','ELON ');

INSERT INTO AUTHOR VALUES('101','DEPP');

INSERT INTO AUTHOR VALUES('104','DAKOTA JHONSON');

INSERT INTO AUTHOR VALUES('116','VPJ');

INSERT INTO AUTHOR VALUES('112','FRANK OCEAN');

- **BOOK RECORDS TABLE**

INSERT INTO BOOK_RECORDS VALUES('7','100','01-MAY-22','05-MAY-22','07-MAY-22');

INSERT INTO BOOK_RECORDS VALUES('5','101','01-MAY-22','05-MAY-22',NULL);

INSERT INTO BOOK_RECORDS VALUES('2','104','07-MAY-22','14-MAY-22','18-MAY-22');

INSERT INTO BOOK_RECORDS VALUES('3','106','07-MAY-22','14-MAY-22',NULL);

INSERT INTO BOOK_RECORDS VALUES('1','116','07-MAY-22','14-MAY-22',NULL);

INSERT INTO BOOK_RECORDS VALUES('4','112','02-MAY-22','11-MAY-22',NULL);

- **RECOMMENDATION TABLE**

Insert Into RECOMMENDATION values ('1','100', 'ARTIFICIAL INTELLIGENCE', NULL);

Insert Into RECOMMENDATION values ('2','104', 'MACHINE LEARNING', NULL);

Insert Into RECOMMENDATION values ('3','106','BIOLOGY', NULL);

Insert Into RECOMMENDATION values ('4','116','AUTOMOTIVE ENGINES', NULL);

Retrieving the data

1. Retrieve the name, gender and address of the member whose ID is 1.

➢ Select Name, Gender, Address from Member where MemberID=1;

2. Display name and publisher name of the book with ID 112.

➢ Select Title , Publisher from Book where Bookid=112;

3. Display name, and ID of the member who took the book with ID 104.

➢ Select Name, MemberID from Member, Book Records where Member.MemberID =Book Records.MemberID and bookid=104;

4. Display the name and author name of the book with ID 100.

➢ Select Author,Title from Author , Book where Book.bookid=Author.bookid and bookid=100;

5. Display ID's of book and member to whom it was issued on 2nd May.

➢ Select Bookid,memberid from Book Records where Date_of_Issue='02-MAY-22';

4.6 Triggers

TRIGGER FOR UPDATING BOOKS ON ISSUE & RETURN

/* TRIGGER TO AUTOMATICALLY INCREMENT & DECREMENT THE NO_OF_BOOKS FROM MEMBER & BOOK TABLE UPON ISSUE & RETURN */

```
CREATE OR REPLACE TRIGGER INCR_TRIGGER
AFTER INSERT OR UPDATE OF MEMBERID ON BOOK_RECORDS
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
BEGIN
 IF INSERTING THEN
 UPDATE BOOK
 SET copies=copies-1
 WHERE BookID=:NEW.BookID;
 UPDATE member
 SET renewals_left=renewals_left-1
 WHERE memberID=:NEW.memberID;
 ELSIF UPDATING THEN
 UPDATE BOOK
 SET copies=copies+1
 WHERE BookID=:NEW.BookID;
 UPDATE member
 SET renewals_left=renewals_left+1
```

```
 WHERE memberID=:NEW.memberID;
 END IF;
END;
/
```

4.7 Stored Procedures

1. **PROCEDURE FOR INSERTION OF NEW BOOKS**

```
CREATE OR REPLACE PROCEDURE INSERT1(BOOK_ID VARCHAR2,
USER_ID NUMBER) IS
MEP NUMBER(4);
TEP NUMBER(4);
SEP NUMBER(4);
BEGIN

/* TO CHECK IF BOOK EXISTS IN LIBRARY OR NOT */

SELECT COUNT(*) INTO TEP FROM BOOK WHERE BOOKID=BOOK_ID;
IF TEP>=1 THEN
DBMS_OUTPUT.PUT_LINE('THIS BOOK ' || BOOKID || ' EXISTS IN
LIBRARY');
ELSE
DBMS_OUTPUT.PUT_LINE('THIS BOOK ' || BOOKID || ' DOES NOT EXIST
IN LIBRARY');
END IF;

/* TO CHECK IF USER IS PART OF THE CLUB OR NOT */
SELECT COUNT(*) INTO MEP FROM MEMBER WHERE
MEMBERID=USER_ID;
IF MEP=1 THEN
```

```
DBMS_OUTPUT.PUT_LINE('THIS USER ' || USERID || ' IS FROM THE CLUB');
ELSE
DBMS_OUTPUT.PUT_LINE('THIS USER ' || USERID || ' IS NOT FROM THE CLUB');
END IF;

/* TO CHECK IF BOOK HAS BEEN ALREADY ISSUED OR NOT */
SELECT COUNT(*) INTO SEP FROM BOOK_RECORDS WHERE BOOKID=BOOK_ID AND USERID=USER_ID
AND DATE_OF_RETURN IS NULL;
IF SEP=1 THEN
DBMS_OUTPUT.PUT_LINE('ISSUING BOOK TO USER ' || USERID ||'.');
ELSE
DBMS_OUTPUT.PUT_LINE(THIS USER ALREADY HAS THE BOOK);
END;
/
```

2. PROCEDURE WHEN A BOOK IS BEING RETURNED.

```
CREATE OR REPLACE PROCEDURE RETURNBOOK(BOOK_ID VARCHAR2, USER_ID NUMBER) IS
FINE NUMBER(20);
USERID NUMBER(20);
RETURN_DATE DATE NOT NULL := '07-MAY-22';
DAT VARCHAR2(5);
DD DATE;
BEGIN
/* CALCULATING FINE FOR RETURNING THE BOOK LATE*/
```

```
SELECT M_NO INTO USER_ID FROM BOOK_RECORDS WHERE BOOKID=BOOK_ID AND USERID=USER_ID;

UPDATE BOOK_RECORDS SET RETURN_DATE='07-MAY-22' WHERE BOOKID=BOOK_ID AND USERID=USER_ID;
SELECT DUE_DATE INTO DD FROM BOOK_RECORDS WHERE BOOKID=BOOK_ID AND USERID=USER_ID;
FINE := (DD - DATE_OF_RETURN) * 5;
DBMS_OUTPUT.PUT_LINE('FINE IS' || FINE);

/* CANNOT RETURN THE BOOK ON SATURDAY OR SUNDAY*/
SELECT TO_CHAR(SYSDATE, 'DY') INTO DAT FROM DUAL;
IF DAT IN ('SUN', 'SAT') THEN
DBMS_OUTPUT.PUT_LINE('IT IS' || TO_CHAR(SYSDATE, 'DAY') || 'SO YOU CANNOT RETURN);
```

4.8 CONCLUSION

We have created a Library Management System using PL/SQL. We got to learn many different aspects of SQL and implement procedures, functions and triggers. This application can be used to easily replace the manual entering of data of books and their records in registers, henceforth providing easy access to anyone who wants check the records and information of books and members.