VELLORE INSTITUTE OF TECHNOLOGY,CHENNAI

DBMS PROJECT

BY-GUNEESH LAKHANPAL

(21BCE5920)

TOPIC:LIBRARY MANAGEMENT SYSTEM

AIM

- Create a mini project to handle the operations of Library Management System using SQL queries and PL/ SQL (use the concept of Cursor, Trigger, Procedure/ function, Package and Exception Handling)

EXPLANATION OF THE PROJECT

- I have created library management system. This system contains the information about members, books, transaction about books and transaction_history. I have used the concept like cursor, trigger, exception handling and procedure.

1. The library has 3 kinds of members:
   a. Monthly this member can borrow 4 books
   b. Yearly this member can borrow 2 books
   c. Lifetime this member can be borrow 6 books

2. The same kind of a book cannot be borrowed by a member at one instance.

3. The fine amount should be calculated basing on the issue date ,return date and due date.

4. The fine amount can be 5/- per day.

5. When a book is issued automatically it should reflect in the book table.

MODULES IN PROJECT

o Issue Book Section

- This module contains all the operation related to:

1. Book No. must be a valid book no. from the book table or handle exception.
2. Member no. Must be a valid Member no. From member table.
3. The same member no. Cannot borrow the same book without returning the book.
4. If the due date is crossing the expiry date of the member doesn't issue the book.
5. If the number of book is already borrowed by the member without returning the book exceeds the membership limit then handle error.
6. If the stock of the book is not available then trap the error.
7. If all validations are fulfilled, then enter into transaction table book no. Memember no. Issue will by sysdate and due]date is sysdate+7,return date is null and fine is null.
8. On Saturday or Sunday no issue of the books.

o Return Book Section

- This module contains all the operation related to:

1. Return of the book is possible only if the member has borrowed the book, check for existence of record in the transaction table.
2. Update return date with the current date and calculate the fine amount by finding the difference between due date and return date.
3. Update the total fine of that member by add this fine amount with the existing total fine in the member table.
4. On Saturday or Sunday no return of book.
5. Upon returning the book delete the information from the transaction table and move the data to transaction history.
6. Create transaction_history as that of transaction table to record old data.

o Write a trigger to automatically increment and decrement the no_of books from the and member table upon issue and return.

o Write a trigger to move the data from the transaction to transaction_history table upon deletion.

EXPLANATION OF TABLES

1. MEMBER TABLE

COLUMN NAME	DATA TYPE	DESCRIPTION
M_NO	VARCHAR2(20)	PRIMARY KEY
M_NAME	VARCHAR2(20)	NOT NULL
M_TYPE	VARCHAR2(20)	(M,Y,L)
NO_OF_BOOKS	NUMBER(4)	
TOT_FINE	NUMBER(4)	

1. BOOK TABLE

COLUMN NAME	DATA TYPE	DESCRIPTION
B_NO	VARCHAR2(20)	PRIMARY KEY
B_NAME	VARCHAR2(20)	NOT NULL
AUTHOR	VARCHAR2(20)	
PRICE	VARCHAR2(20)	
NO_OF_BOOKS	NUMBER(4)	

1. TRANSACTION TABLE

COLUMN NAME	DATA TYPE	DESCRIPTION
B_NO	VARCHAR2(20)	FOREIGN KEY OF BOOKS
M_NO	VARCHAR2(20)	FOREIGN KEY OF MEMBER
ISSUE_DATE	DATE	SYSDATE
DUE_DATE	DATE	SYSDATE+7
RETURN_DATE	DATE	

MAIN SOURCE CODE

- **TABLE CREATION**

/*---------- MEMBER TABLE ----------*/

```
CREATE TABLE MEMBER (M_NO VARCHAR2(20) PRIMARY KEY, M_NAME VARCHAR2(20) NOT NULL, M_TYPE VARCHAR2(20), NO_OF_BOOKS NUMBER(4), TOT_FINE NUMBER(4));
```

/*---------- BOOK TABLE ---------- */

```
CREATE TABLE BOOK (B_NO VARCHAR2(20) PRIMARY KEY, B_NAME VARCHAR2(20) NOT NULL, AUTHOR VARCHAR2(20), PRICE VARCHAR2 (20), NO_OF_BOOKS NUMBER(4));
```

/*---------- TRANSACTION TABLE ---------- */

```
CREATE TABLE TRANSACTION (B_NO VARCHAR2(20), M_NO VARCHAR2(20), ISSUE_DATE DATE, DUE_DATE DATE, RETURN_DATE DATE, CONSTRAINT BID_FKEY FOREIGN KEY (B_NO) REFERENCES BOOK(B_NO), CONSTRAINT MID_FKEY FOREIGN KEY (M_NO) REFERENCES MEMBER(M_NO));
```

/*---------- TRANSACTION HISTORYTABLE ---------- */

```
CREATE TABLE TRANSACTION_HISTORY (B_NO VARCHAR2(20), M_NO VARCHAR2(20), ISSUE_DATE DATE, DUE_DATE DATE, RETURN_DATE DATE, CONSTRAINT BID_FKEY1 FOREIGN KEY (B_NO) REFERENCES BOOK(B_NO), CONSTRAINT MID_FKEY1 FOREIGN KEY (M_NO) REFERENCES MEMBER(M_NO));
```

- **INSERTING SOME DATA AT THE STARTING**

/*---------- VALUES IN MEMBER TABLE ---------- */

```
INSERT INTO MEMBER VALUES('1','DEEPESH','M',2,NULL);
INSERT INTO MEMBER VALUES('2','PRIYANSH','L',0,NULL);
INSERT INTO MEMBER VALUES('3','AKASH','Y',2,NULL);
INSERT INTO MEMBER VALUES('4','SWATI','M',4,NULL);
INSERT INTO MEMBER VALUES('5','BOSS','L',1,NULL);
INSERT INTO MEMBER VALUES('6','PRATIKHYA','Y',1,NULL);
INSERT INTO MEMBER VALUES('7','DHRUTI','L',2,NULL)
```

/*---------- VALUES IN BOOK TABLE---------- */

```
INSERT INTO BOOK VALUES('B1','YES YOU CAN WIN!','GAREY V','200',2);
INSERT INTO BOOK VALUES('B2','HALF GIRLFRIEND','CHETAN BHAGAT','100',3);
INSERT INTO BOOK VALUES('B3','HOW I MET UR MOTHER?','BARNEY
SINSTON','500',5);
INSERT INTO BOOK VALUES('B4','CORPORATE CHANKYA','MIRAL','170',5);
INSERT INTO BOOK VALUES('B5','LIFE AT EDGE','TDP','650',0);
INSERT INTO BOOK VALUES('B6','VIVEK GEETA!','KABIR','180',2);
```

/*---------- VALUES IN TRANS TABLE---------- */

```
INSERT INTO TRANSACTION VALUES('B4','7','01-MAY-20','05-MAY-20','07-MAY-20');
INSERT INTO TRANSACTION VALUES('B3','5','01-MAY-20','05-MAY-20',NULL);
INSERT INTO TRANSACTION VALUES('B3','2','07-MAY-20','14-MAY-20',NULL);
INSERT INTO TRANSACTION VALUES('B6','3','07-MAY-20','14-MAY-20',NULL);
INSERT INTO TRANSACTION VALUES('B1','1','07-MAY-20','14-MAY-20',NULL);
```

- **PL/SQL BLOCK FOR ISSUING BOOK**

/*---------- PROCEDURE TO ISSUE BOOK ----------P_NAME:ISSUE;PARAMETES:B_NO,M_NO)*/

```
CREATE OR REPLACE PROCEDURE INSERT1(BOOK_ID VARCHAR2,MEM_ID NUMBER)
IS
A BOOLEAN DEFAULT FALSE;
B BOOLEAN DEFAULT FALSE;
C BOOLEAN DEFAULT FALSE;
D BOOLEAN DEFAULT FALSE;
MEP NUMBER(4);
TEP NUMBER(4);
SEP NUMBER(4);
BEP NUMBER(4);
MB NUMBER(4);
DAT VARCHAR2(10);
TYP VARCHAR2(10);
EXPIRY_DATE DATE;
DDATE DATE;

BEGIN
```

/*(A)BOOK NO SHOULD BE VALID FROM BOOK TABLE OR HANDLE EXCEPTION*/

```
SELECT COUNT(*) INTO TEP FROM BOOK WHERE B_NO=BOOK_ID;
IF TEP=1
THEN
dbms_output.put_line('THIS BOOK '||BOOK_ID||' EXSIST IN LIBRARY.');
ELSE
```

```
dbms_output.put_line('THIS BOOK '||BOOK_ID||' DOSE NOT EXSIST IN LIBRARY.');
END IF;
```

/*(B)MEMBER NO SHOULD BE VALID FROM MEMBER TABLE OR HANDLE EXCEPTION*/

```
SELECT COUNT(*) INTO MEP FROM MEMBER WHERE M_NO=MEM_ID;
IF MEP=1
THEN
dbms_output.put_line('THE USER '||MEM_ID||' IS FROM THE CLUB.');
ELSE
dbms_output.put_line('THE USER '||MEM_ID||' IS NOT FROM THE CLUB.');
END IF;
```

/*(C)THE SAME MEMBER CAN'T BORROW THE SAME WITHOUT RETURING IT.*/

```
SELECT COUNT(*) INTO SEP FROM TRANSACTION WHERE B_NO=BOOK_ID AND
M_NO=MEM_ID AND RETURN_DATE IS NULL;
IF SEP=1
THEN
dbms_output.put_line('ISSUING BOOK TO MEMBER ‘||MEM_ID||’.');

ELSE
dbms_output.put_line('THE USER ALREADY HAVE THIS BOOK.');
END IF;
```

/*(D)IF THE DUE DATE CROSSING THE EXPIRY DATE OF THE MEMBER THEN DON'T ISSUE THE BOOK.*/

```
SELECT M_TYPE INTO TYP FROM MEMBER WHERE M_NO=MEM_ID;
EXPIRY_DATE:=ADD_MONTHS(ROUND(SYSDATE,'MONTH'),1);
DDATE:=ROUND(SYSDATE,'YEAR');
IF TYP='M'
THEN
   IF EXPIRY_DATE<SYSDATE+7
   THEN
   A:=TRUE;
   dbms_output.put_line('YOUR MEMBERSHIP EXPIRY DATE '||EXPIRY_DATE||' IS BEFORE
DUE DATE '||SYSDATE+7||'.');
   END IF;
ELSIF TYP='Y'
THEN
   IF EXPIRY_DATE<SYSDATE+7
   THEN
   A:=TRUE;
   dbms_output.put_line('YOUR MEMBERSHIP EXPIRY DATE '||EXPIRY_DATE||' IS BEFORE
DUE DATE '||SYSDATE+7||'.');
   END IF;
ELSIF TYP='L'
THEN
   dbms_output.put_line('YOU HAVE LIFETIME MEMBERSHIP.');
END IF;
```

/*(E)IF THE NUMBER OF BOOK IS ALREDAY BORROWED BY THE MEMBER WITHOUT RETURING THE BOOK EXCEEDS THE MEMBERSHIP LIMIT THEN HANDLE ERROR. */

```
SELECT NO_OF_BOOKS INTO MB FROM MEMBER WHERE M_NO=MEM_ID;
IF TYP='M'
THEN
   IF MB>=4
   THEN
   B:=TRUE;
   dbms_output.put_line('YOU HAVE REACHED MONTHLY BORROW LIMIT OF 4
BOOKS.');
   END IF;
ELSIF TYP='Y'
THEN
   IF MB>=2
   THEN
   B:=TRUE;
   dbms_output.put_line('YOU HAVE REACHED YEARLY BORROW LIMIT OF 2 BOOKS.');
   END IF;
ELSIF TYP='L'
THEN
   IF MB>=6
   THEN
   B:=TRUE;
   dbms_output.put_line('YOU HAVE REACHED LIFETIME BORROW LIMIT OF 6 BOOKS.');
   END IF;
END IF;
```

/*(F)IF THE STOCK OF THE BOOK IS NOT AVAILABLE THEN TRAP THE ERROR.*/

```
SELECT NO_OF_BOOKS INTO BEP FROM BOOK WHERE B_NO=BOOK_ID;
IF BEP>=1
THEN
D:=TRUE;
   dbms_output.put_line('THE BOOK IS AVAILABLE IN THE LIBRARY.');
END IF;
```

/*(G)IF ALL VALIDATIONS ARE FULFILLED, THEN ENTER INTO TRANSACTION TABLE BOOKNO.,MEMNO. ISSUE WILL BY SYSDATE AND DUE_DATE IS SYSDATE+7 ,RETURE DATE IS NULL & FINE IS NULL.*/

```
IF(TEP IS NOT NULL AND MEP IS NOT NULL AND B IS NOT NULL AND A IS NOT NULL
AND  D IS NOT NULL AND C IS NOT NULL)
THEN
INSERT INTO TRANSACTION
VALUES(BOOK_ID,MEM_ID,SYSDATE,SYSDATE+7,NULL);
dbms_output.put_line('ITS WORKING');
END IF;
```

/*(H) ON SATURDAY OR SUNDAY NO ISSUE OF THE BOOKS.*/

```
SELECT TO_CHAR(SYSDATE,'DY') INTO DAT FROM DUAL;
IF DAT='SUN'
THEN
dbms_output.put_line('IT IS '||TO_CHAR(SYSDATE,'DAY')||'SO CANNOT ISSUE THE
BOOK.');
ELSIF
DAT = 'SAT'
THEN
dbms_output.put_line('IT IS '||TO_CHAR(SYSDATE,'DAY')||'SO CANNOT ISSUE THE
BOOK.');
ELSE
C:=TRUE;
dbms_output.put_line('IT IS '||TO_CHAR(SYSDATE,'DAY')||'SO CAN ISSUE BOOK.');
END IF;

END;
/
```

- **PL/SQL BLOCK FOR RETURING BOOK**

/*---PROCEDURE TO RETURN BOOK --- PROC_NAMEE:RETURNBOOK;PARAMETES:B_NO,M_NO)*/

```
CREATE OR REPLACE PROCEDURE RETURNBOOK(BOOK_ID VARCHAR2,MEM_ID
NUMBER)
IS
FINE NUMBER(20);
MEMID NUMBER(20);
RETRN_DATE DATE NOT NULL:='07-MAY-20';
DAT VARCHAR2(5);
DD DATE;

BEGIN
```

/*(A)RETURN THE BOOK IF THE MEMBER HAS BOWRROWED THE BOOK, CHECK IN THE EXSISTENCE TRANSACTION TABLE.*/

```
SELECT M_NO INTO MEMID FROM TRANSACTION WHERE M_NO=MEM_ID AND
B_NO=BOOK_ID;
```

/*(B)UPDATE RETURN DATE WITH CURRENT DATE & CALCUATE THE AMOUNT OF FINE.*/

```
UPDATE TRANSACTION SET RETURN_DATE='07-MAY-20' WHERE B_NO=BOOK_ID
AND M_NO=MEM_ID;
SELECT DUE_DATE INTO DD FROM TRANSACTION WHERE B_NO=BOOK_ID AND
M_NO=MEM_ID;
FINE:=(DD-RETRN_DATE)*5;
```

```
/*(C)UPDATE THE TOT_FINE IN THE MEMBER TABLE.*/

UPDATE MEMBER SET TOT_FINE=FINE WHERE M_NO=MEM_ID;

/*(D)NO RETURN ON SATURDAY & SUNDAY.*/

SELECT TO_CHAR(SYSDATE,'DY') INTO DAT FROM DUAL;
IF DAT='SUN'
THEN
dbms_output.put_line('IT IS '||to_char(SYSDATE,'DAY')||' SO YOU CANNOT RETURN
BOOK.');
END IF;
IF DAT='SAT'
THEN
dbms_output.put_line('IT IS '||to_char(SYSDATE,'DAY')||' SO YOU CANNOT RETURN
BOOK.');
END IF;

/*(E)UPON RETURING THE BOOK DELETE THE INFORMATION FROM TRANSACTION & MOVE TO
TRANSACTION_HISTORY TABLE.*/

--USED USING TRIGGER

/*(F)CREATE TRANSACTION_HISTORY AS THAT OF TRANSACTION TABLE TO RECORD OLD DATA.*/

EXCEPTION
WHEN NO_DATA_FOUND THEN
dbms_output.put_line('THERE IS NO BOOK ISSUED TO THIS MEMBER');
END;
/
```

- **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 ON TRANSACTION
FOR EACH ROW
BEGIN
   IF INSERTING THEN
        UPDATE BOOK
        SET NO_OF_BOOKS=NO_OF_BOOKS-1
```

```
UPDATE MEMBER
SETNO_OF_BOOKS=G THEN UPDATE
BOOK
SET NO_OF_BOOKS=NO_OF_BOOKS+1
WHERE B_N0=:OLD.B_NO;
UPDATE MEMBER
SET NO_OF_BOOKS=NO_OF_BOOKS-1
WHERE M_NO=:OLD.M_NO;
```

- **TRIGGER FOR DELETING DATA FROM TANSACTION & MOVE IT TO TRANSACTION HISTORY TABLE**

```
/* ----------TRIGGER TO MOVE TRANSACTION DATA INTO TRANSCTION_HISTORY TABLE

UPON DELETION ----------*/

CREATE OR REPLACE TRIGGER MOVE_TRIGGER
BEFORE DELETE ON TRANSACTION
FOR EACH ROW
BEGIN INSERT INTO TRANSACTION_HISTORY
VALUES(:OLD.B_NO,:OLD.M_NO,:OLD.ISSUE_DATE,:OLD.DUE_DATE,:OLD.RETURN_D
ATE);
END;
/
```

RESULT

TABLE CREATION

SQL Worksheet

Clear | Find | Actions | Save | Run

```
1  /*-------------------------------- LIBRARY MANAGEMENT SYSTEM ----------------------------------*/
2
3  /*---------- MEMBER TABLE ----------*/
4  CREATE TABLE MEMBER (M_NO VARCHAR2(20) PRIMARY KEY, M_NAME VARCHAR2(20) NOT NULL, M_TYPE VARCHAR2(20), NO_OF_BOOKS NUMBER(4), TOT_FINE NUMBER(4));
5  /*---------- BOOK TABLE ----------*/
6  CREATE TABLE BOOK (B_NO VARCHAR2(20) PRIMARY KEY, B_NAME VARCHAR2(20) NOT NULL, AUTHOR VARCHAR2(20), PRICE VARCHAR2 (20), NO_OF_BOOKS NUMBER(4));
7  /*---------- TRANSACTION TABLE ----------*/
8  CREATE TABLE TRANSACTION (B_NO VARCHAR2(20), M_NO VARCHAR2(20), ISSUE_DATE DATE, DUE_DATE DATE, RETURN_DATE DATE, CONSTRAINT BID_FKEY FOREIGN KEY (B_NO)
9  REFERENCES BOOK(B_NO), CONSTRAINT MID_FKEY  FOREIGN KEY (M_NO) REFERENCES MEMBER(M_NO));
10 /*---------- TRANSACTION HISTORY TABLE ----------*/
11 CREATE TABLE TRANSACTION_HISTORY (B_NO VARCHAR2(20), M_NO VARCHAR2(20), ISSUE_DATE DATE, DUE_DATE DATE, RETURN_DATE DATE, CONSTRAINT BID_FKEY1 FOREIGN KEY (B_NO)
12 REFERENCES BOOK(B_NO), CONSTRAINT MID_FKEY1 FOREIGN KEY (M_NO) REFERENCES MEMBER(M_NO));
13
```

Table created.

```
18 /*------------- VALUES IN MEMBER TABLE -------------*/
19 INSERT INTO MEMBER VALUES('1','DEEPESH','M',2,NULL);
20 INSERT INTO MEMBER VALUES('2','PRIYANSH','L',0,NULL);
21 INSERT INTO MEMBER VALUES('3','AKASH','Y',2,NULL);
22 INSERT INTO MEMBER VALUES('4','SWATI','M',4,NULL);
23 INSERT INTO MEMBER VALUES('5','BOSS','L',1,NULL);
24 INSERT INTO MEMBER VALUES('6','PRATIKHYA','Y',1,NULL);
25 INSERT INTO MEMBER VALUES('7','DHRUTI','L',2,NULL);
26
27 SELECT * FROM MEMBER;
28
```

M_NO	M_NAME	M_TYPE	NO_OF_BOOKS	TOT_FINE
1	DEEPESH	M	2	–
2	PRIYANSH	L	0	–
3	AKASH	Y	2	–
4	SWATI	M	4	–
5	BOSS	L	1	–
6	PRATIKHYA	Y	1	–
7	DHRUTI	L	2	–

Download CSV

7 rows selected.

```
/*------------ VALUES IN BOOK TABLE ------------*/
INSERT INTO BOOK VALUES('B1','YES YOU CAN WIN!','GAREY V','200',2);
INSERT INTO BOOK VALUES('B2','HALF GIRLFRIEND','CHETAN BHAGAT','100',3);
INSERT INTO BOOK VALUES('B3','HOW I MET UR MOTHER?','BARNEY SINSTON','500',5);
INSERT INTO BOOK VALUES('B4','CORPORATE CHANKYA','MIRAL','170',5);
INSERT INTO BOOK VALUES('B5','LIFE AT EDGE','TDP','650',0);
INSERT INTO BOOK VALUES('B6','VIVEK GEETA!','KABIR','180',2);

SELECT * FROM BOOK;
```

B_NO	B_NAME	AUTHOR	PRICE	NO_OF_BOOKS
B1	YES YOU CAN WIN!	GAREY V	200	2
B2	HALF GIRLFRIEND	CHETAN BHAGAT	100	3
B3	HOW I MET UR MOTHER?	BARNEY SINSTON	500	5
B4	CORPORATE CHANKYA	MIRAL	170	5
B5	LIFE AT EDGE	TDP	650	0
B6	VIVEK GEETA!	KABIR	180	2

Download CSV

6 rows selected.

```
/*------------ VALUES IN TRANS TABLE ------------*/
INSERT INTO TRANSACTION VALUES('B4','7','01-MAY-20','05-MAY-20','07-MAY-20');
INSERT INTO TRANSACTION VALUES('B3','5','01-MAY-20','05-MAY-20',NULL);
INSERT INTO TRANSACTION VALUES('B3','2','07-MAY-20','14-MAY-20',NULL);
INSERT INTO TRANSACTION VALUES('B6','3','07-MAY-20','14-MAY-20',NULL);
INSERT INTO TRANSACTION VALUES('B1','1','07-MAY-20','14-MAY-20',NULL);

SELECT * FROM TRANSACTION;
```

B_NO	M_NO	ISSUE_DATE	DUE_DATE	RETURN_DATE
B4	7	01-MAY-20	05-MAY-20	07-MAY-20
B3	5	01-MAY-20	05-MAY-20	-
B3	2	07-MAY-20	14-MAY-20	-
B6	3	07-MAY-20	14-MAY-20	-
B1	1	07-MAY-20	14-MAY-20	-

Download CSV

5 rows selected.

```
/*------------- PROCEDURE TO ISSUE BOOK -------------P_NAME:ISSUE;PARAMETES:B_NO,M_NO)*/
CREATE OR REPLACE PROCEDURE INSERT1(BOOK_ID VARCHAR2,MEM_ID NUMBER)
IS
A BOOLEAN DEFAULT FALSE;
B BOOLEAN DEFAULT FALSE;
C BOOLEAN DEFAULT FALSE;
D BOOLEAN DEFAULT FALSE;
MEP NUMBER(4);
TEP NUMBER(4);
SEP NUMBER(4);
BEP NUMBER(4);
MB NUMBER(4);
DAT VARCHAR2(10);
TYP VARCHAR2(10);
EXPIRY_DATE DATE;
DDATE DATE;

BEGIN

/*(A)BOOK NO SHOULD BE VALID FROM BOOK TABLE OR HANDLE EXCEPTION*/
SELECT COUNT(*) INTO TEP FROM BOOK WHERE B_NO=BOOK_ID;
IF TEP=1
THEN
dbms_output.put_line('THIS BOOK '||BOOK_ID||' EXSIST IN LIBRARY.');
ELSE
dbms_output.put_line('THIS BOOK '||BOOK_ID||' DOSE NOT EXSIST IN LIBRARY.');
END IF;

/*(B)MEMBER NO SHOULD BE VALID FROM MEMBER TABLE OR HANDLE EXCEPTION*/
SELECT COUNT(*) INTO MEP FROM MEMBER WHERE M_NO=MEM_ID;
IF MEP=1
THEN
dbms_output.put_line('THE USER '||MEM_ID||' IS FROM THE CLUB.');
ELSE
dbms_output.put_line('THE USER '||MEM_ID||' IS NOT FROM THE CLUB.');
```

```
/*(C)THE SAME MEMBER CAN'T BORROW THE SAME WITHOUT RETURING IT.*/
SELECT COUNT(*) INTO SEP FROM TRANSACTION WHERE B_NO=BOOK_ID AND M_NO=MEM_ID AND RETURN_DATE IS NULL;
IF SEP=1
THEN
dbms_output.put_line('THE USER ALREADY HAVE THIS BOOK.');
END IF;

/*(D)IF THE DUE DATE CROSSING THE EXPIRY DATE OF THE MEMBER THEN DON'T ISSUE THE BOOK.*/
SELECT M_TYPE INTO TYP FROM MEMBER WHERE M_NO=MEM_ID;
EXPIRY_DATE:=ADD_MONTHS(ROUND(SYSDATE,'MONTH'),1);
DDATE:=ROUND(SYSDATE,'YEAR');
IF TYP='M'
THEN
    IF EXPIRY_DATE<SYSDATE+7
    THEN
    A:=TRUE;
    dbms_output.put_line('YOUR MEMBERSHIP EXPIRY DATE '||EXPIRY_DATE||' IS BEFORE DUE DATE '||SYSDATE+7||'.');
    END IF;
ELSIF TYP='Y'
THEN
    IF DDATE<SYSDATE+7
    THEN
    A:=TRUE;
    dbms_output.put_line('YOUR MEMBERSHIP EXPIRY DATE '||EXPIRY_DATE||' IS BEFORE DUE DATE '||SYSDATE+7||'.');
    END IF;
ELSIF TYP='L'
THEN
    dbms_output.put_line('YOU HAVE LIFETIME MEMBERSHIP.');
END IF;

/*(E)IF THE NUMBER OF BOOK IS ALREDAY BORROWED BY THE MEMBER WITHOUT RETURING THE BOOK EXCEEDS THE MEMBERSHIP LIMIT THEN HANDLE ERROR. */
SELECT NO_OF_BOOKS INTO MB FROM MEMBER WHERE M_NO=MEM_ID;
IF TYP='M'
THEN
    IF MB>=4
```

```
    THEN
    B:=TRUE;
    dbms_output.put_line('YOU HAVE REACHED MONTHLY BORROW LIMIT OF 4 BOOKS.');
    END IF;
ELSIF TYP='Y'
THEN
    IF MB>=2
    THEN
    B:=TRUE;
    dbms_output.put_line('YOU HAVE REACHED YEARLY BORROW LIMIT OF 2 BOOKS.');
    END IF;
ELSIF TYP='L'
THEN
    IF MB>=6
    THEN
    B:=TRUE;
    dbms_output.put_line('YOU HAVE REACHED LIFETIME BORROW LIMIT OF 6 BOOKS.');
    END IF;
END IF;

/*(F)IF THE STOCK OF THE BOOK IS NOT AVAILABLE THEN TRAP THE ERROR.*/
SELECT NO_OF_BOOKS INTO BEP FROM BOOK WHERE B_NO=BOOK_ID;
IF BEP>=1
THEN
D:=TRUE;
    dbms_output.put_line('THE BOOK IS AVAILABLE IN THE LIBRARY.');
END IF;


/*(G)IF ALL VALIDATIONS ARE FULFILLED, THEN ENTER INTO TRANSACTION TABLE BOOKNO.,MEMNO. ISSUE WILL BY SYSDATE AND DUE_DATE IS SYSDATE+7 ,RETURE DATE IS NULL & FIN
IF(TEP IS NOT NULL AND MEP IS NOT NULL AND B IS NOT NULL AND A IS NOT NULL AND  D IS NOT NULL AND C IS NOT NULL)
THEN
INSERT INTO TRANSACTION VALUES(BOOK_ID,MEM_ID,SYSDATE,SYSDATE+7,NULL);
dbms_output.put_line('ITS WORKING');
END IF;
```

```
/*(H) ON SATURDAY OR SUNDAY NO ISSUE OF THE BOOKS.*/
SELECT TO_CHAR(SYSDATE,'DY') INTO DAT FROM DUAL;
IF DAT='SUN'
THEN
dbms_output.put_line('IT IS '||TO_CHAR(SYSDATE,'DAY')||'SO CANNOT ISSUE THE BOOK.');
ELSIF
DAT = 'SAT'
THEN
dbms_output.put_line('IT IS '||TO_CHAR(SYSDATE,'DAY')||'SO CANNOT ISSUE THE BOOK.');
ELSE
C:=TRUE;
dbms_output.put_line('IT IS '||TO_CHAR(SYSDATE,'DAY')||'SO CAN ISSUE BOOK.');
END IF;

END;
/
```

```
/*------------- PROCEDURE TO RETURN BOOK -------------PROC_NAMEE:RETURNBOOK;PARAMETES:B_NO,M_NO)*/
CREATE OR REPLACE PROCEDURE RETURNBOOK(BOOK_ID VARCHAR2,MEM_ID NUMBER)
IS
FINE NUMBER(20);
MEMID NUMBER(20);
RETRN_DATE DATE NOT NULL:='07-MAY-20';
DAT VARCHAR2(5);
DD DATE;

BEGIN

/*(A)RETURN THE BOOK IF THE MEMBER HAS BOWRROWED THE BOOK, CHECK IN THE EXSISTENCE TRANSACTION TABLE.*/
SELECT M_NO INTO MEMID FROM TRANSACTION WHERE M_NO=MEM_ID AND B_NO=BOOK_ID;

/*(B)UPDATE RETURN DATE WITH CURRENT DATE  & CALCUATE THE AMOUNT OF FINE.*/
UPDATE TRANSACTION SET RETURN_DATE='07-MAY-20' WHERE B_NO=BOOK_ID AND M_NO=MEM_ID;
SELECT DUE_DATE INTO DD FROM TRANSACTION WHERE B_NO=BOOK_ID AND M_NO=MEM_ID;
FINE:=(DD-RETRN_DATE)*5;
dbms_output.put_line('FINE IS '||fine);

/*(C)UPDATE THE TOT_FINE IN THE MEMBER TABLE.*/
UPDATE MEMBER SET TOT_FINE=FINE WHERE M_NO=MEM_ID;

/*(D)NO RETURN ON SATURDAY & SUNDAY.*/
SELECT TO_CHAR(SYSDATE,'DY') INTO DAT FROM DUAL;
IF DAT='SUN'
THEN
dbms_output.put_line('IT IS '||to_char(SYSDATE,'DAY')||' SO YOU CANNOT RETURN BOOK.');
END IF;
IF DAT='SAT'
THEN
dbms_output.put_line('IT IS '||to_char(SYSDATE,'DAY')||' SO YOU CANNOT RETURN BOOK.');
END IF;
```

```
/*(E)UPON RETURING THE BOOK DELETE THE INFORMATION FROM TRANSACTION & MOVE TO TRANSACTION_HISTORY TABLE.*/
--USED USING TRIGGER

/*(F)CREATE TRANSACTION_HISTORY AS THAT OF TRANSACTION TABLE TO RECORD OLD DATA.*/
EXCEPTION
WHEN NO_DATA_FOUND THEN
dbms_output.put_line('THERE IS NO BOOK ISSUED TO THIS MEMBER');
END;
/

/*------------- 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 ON TRANSACTION
FOR EACH ROW
BEGIN
    IF INSERTING THEN
            UPDATE BOOK
            SET NO_OF_BOOKS=NO_OF_BOOKS-1
            WHERE B_NO=:NEW.B_NO;
            UPDATE MEMBER
            SET NO_OF_BOOKS=NO_OF_BOOKS+1
            WHERE M_NO=:NEW.M_NO;
    ELSIF UPDATING THEN
            UPDATE BOOK
            SET NO_OF_BOOKS=NO_OF_BOOKS+1
            WHERE B_NO=:OLD.B_NO;
            UPDATE MEMBER
            SET NO_OF_BOOKS=NO_OF_BOOKS-1
            WHERE M_NO=:OLD.M_NO;
    END IF;
END;
/
```

```
/*-----------------WRITE A TRIGGER TO MOVE DATE FROM TRANSACTION INTO TRANSACTION_HISTORY UPON DELETION--------------------*/
CREATE OR REPLACE TRIGGER MOVE_TRIGGER
BEFORE DELETE ON TRANSACTION
FOR EACH ROW
BEGIN
INSERT INTO TRANSACTION_HISTORY VALUES(:OLD.B_NO,:OLD.M_NO,:OLD.ISSUE_DATE,:OLD.DUE_DATE,:OLD.RETURN_DATE);
END;
/
```

```
Procedure created.

Procedure created.

Trigger created.

Trigger created.
```

ISSUING A BOOK ON SUNDAY

```
107  BEGIN INSERT1('B2',6);
108  EXCEPTION
109     WHEN OTHERS
110     THEN
111        DBMS_OUTPUT.put_line (DBMS_UTILITY.format_error_stack);
112  END;
113  /
```

```
Statement processed.
THIS BOOK B2 EXSIST IN LIBRARY.
THE USER 6 IS FROM THE CLUB.


ITS WORKING
IT IS SUNDAY   SO CANNOT ISSUE THE BOOK.
```

ISSUING A BOOK ON MONDAY

```
248  BEGIN INSERT1('B2',6);
249  EXCEPTION
250     WHEN OTHERS
251     THEN
252      dbms_output.put_line(DBMS_UTILITY.format_error_stack);
253  END;
254  /
```

```
Statement processed.
THIS BOOK B2 EXSIST IN LIBRARY.
THE USER 6 IS FROM THE CLUB.
ISSUING BOOK TO MEMBER 6.
THE BOOK IS AVAILABLE IN THE LIBRARY.
ITS WORKING
IT IS MONDAY   SO CAN ISSUE BOOK.
```

ISSUING THE SAME BOOK AGAIN

```
248  BEGIN INSERT1('B2',6);
249  EXCEPTION
250     WHEN OTHERS
251     THEN
252      dbms_output.put_line(DBMS_UTILITY.format_error_stack);
253  END;
254  /
```

```
Statement processed.
THIS BOOK B2 EXSIST IN LIBRARY.
THE USER 6 IS FROM THE CLUB.
THE USER ALREADY HAVE THIS BOOK.
THE BOOK IS AVAILABLE IN THE LIBRARY.
ITS WORKING
IT IS MONDAY   SO CAN ISSUE BOOK.
```

TRANSACTION TABLE AFTER ISSUING A BOOK (‘B2’,6)

B_NO	M_NO	ISSUE_DATE	DUE_DATE	RETURN_DATE
B4	7	01-MAY-20	05-MAY-20	07-MAY-20
B3	5	01-MAY-20	05-MAY-20	-
B3	2	07-MAY-20	14-MAY-20	-
B6	3	07-MAY-20	14-MAY-20	-
B1	1	07-MAY-20	14-MAY-20	-
B2	6	12-MAY-20	19-MAY-20	-

Download CSV
6 rows selected.

USER CAN BOOK MORE BOOK OR NOT

```
254  BEGIN INSERT1('B2',4);
255  EXCEPTION
256     WHEN OTHERS
257     THEN
258      dbms_output.put_line(DBMS_UTILITY.format_error_stack);
259  END;
260  /
```

```
Statement processed.
THIS BOOK B2 EXSIST IN LIBRARY.
THE USER 4 IS FROM THE CLUB.

YOU HAVE REACHED MONTHLY BORROW LIMIT OF 4 BOOKS.
```

CAN'T RETURN A BOOK ON SUNDAY OR SATURADY

```
30  BEGIN RETURNBOOK('B3',2);
31  EXCEPTION
32     WHEN OTHERS
33     THEN
34      dbms_output.put_line(DBMS_UTILITY.format_error_stack);
35  END;
36  /
37
```

```
Statement processed.
FINE IS 35
IT IS SUNDAY    SO YOU CANNOT RETURN BOOK.
```

RETURNING BOOK ON OTHER DAYS

```
261
262 BEGIN RETURNBOOK('B3',2);
263 EXCEPTION
264    WHEN OTHERS
265    THEN
266     dbms_output.put_line(DBMS_UTILITY.format_error_stack);
267 END;
268
```

```
Statement processed.
FINE IS 35
```

```
272 SELECT * FROM MEMBER;
```

M_NO	M_NAME	M_TYPE	NO_OF_BOOKS	TOT_FINE
1	DEEPESH	M	2	–
2	PRIYANSH	L	1	35
3	AKASH	Y	2	–
4	SWATI	M	5	–
5	BOSS	L	1	–
6	PRATIKHYA	Y	1	–
7	DHRUTI	L	2	–

Download CSV
7 rows selected.

TRIGGER ON INCREMENT & DECREMRNT NO. OF BOOKS ON ISSUE & RETURN BEFORE ISSUE OF BOOK

TO ('B4',2)

```
272  SELECT * FROM MEMBER;
```

M_NO	M_NAME	M_TYPE	NO_OF_BOOKS	TOT_FINE
1	DEEPESH	M	2	–
2	PRIYANSH	L	1	35
3	AKASH	Y	2	–
4	SWATI	M	5	–
5	BOSS	L	1	–
6	PRATIKHYA	Y	1	–
7	DHRUTI	L	2	–

Download CSV
7 rows selected.

```
273  SELECT * FROM BOOK;
```

B_NO	B_NAME	AUTHOR	PRICE	NO_OF_BOOKS
B1	YES YOU CAN WIN!	GAREY V	200	2
B2	HALF GIRLFRIEND	CHETAN BHAGAT	100	2
B3	HOW I MET UR MOTHER?	BARNEY SINSTON	500	6
B4	CORPORATE CHANKYA	MIRAL	170	5
B5	LIFE AT EDGE	TDP	650	0
B6	VIVEK GEETA!	KABIR	180	1

Download CSV
6 rows selected.

AFTER ISSUE OF BOOK TO ('B4',2)

M_NO	M_NAME	M_TYPE	NO_OF_BOOKS	TOT_FINE
1	DEEPESH	M	2	–
2	PRIYANSH	L	2	35
3	AKASH	Y	2	–
4	SWATI	M	5	–
5	BOSS	L	1	–
6	PRATIKHYA	Y	1	–
7	DHRUTI	L	2	–

Download CSV
7 rows selected.

B_NO	B_NAME	AUTHOR	PRICE	NO_OF_BOOKS
B1	YES YOU CAN WIN!	GAREY V	200	2
B2	HALF GIRLFRIEND	CHETAN BHAGAT	100	2
B3	HOW I MET UR MOTHER?	BARNEY SINSTON	500	6
B4	CORPORATE CHANKYA	MIRAL	170	4
B5	LIFE AT EDGE	TDP	650	0
B6	VIVEK GEETA!	KABIR	180	1

Download CSV

RETURNING BOOK ('B6',3)

```
283  BEGIN RETURNBOOK('B6',3);
284  EXCEPTION
285     WHEN OTHERS
286     THEN
287      dbms_output.put_line(DBMS_UTILITY.format_error_stack);
288  END;
289
```

M_NO	M_NAME	M_TYPE	NO_OF_BOOKS	TOT_FINE
1	DEEPESH	M	2	-
2	PRIYANSH	L	2	35
3	AKASH	Y	1	35
4	SWATI	M	5	-
5	BOSS	L	1	-
6	PRATIKHYA	Y	1	-
7	DHRUTI	L	2	-

Download CSV
7 rows selected.

B_NO	B_NAME	AUTHOR	PRICE	NO_OF_BOOKS
B1	YES YOU CAN WIN!	GAREY V	200	2
B2	HALF GIRLFRIEND	CHETAN BHAGAT	100	2
B3	HOW I MET UR MOTHER?	BARNEY SINSTON	500	6
B4	CORPORATE CHANKYA	MIRAL	170	4
B5	LIFE AT EDGE	TDP	650	0
B6	VIVEK GEETA!	KABIR	180	2

TRIGGER TO MOVE DATA FROM TRANSACTION TO TRANSACTION HISTORY ON DELETION

```
300  SELECT * FROM TRANSACTION_HISTORY;
```

no data found

```
300  SELECT * FROM TRANSACTION_HISTORY;
301  DELETE FROM TRANSACTION WHERE B_NO='B2' AND M_NO='4';
```

B_NO	M_NO	ISSUE_DATE	DUE_DATE	RETURN_DATE
B2	4	12-MAY-20	19-MAY-20	-

Download CSV

CONCLUSION :

I have created a library management system using PL/SQL. I have learnt to many things during this project creation. I have learnt the concept like cursor, trigger, function procedure, package, package body etc.