DBMS PROJECT-LIBRARY MANAGEMENT SYSTEM

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Submitted to – Ma'am Simran

BY

Yashika Gupta (102103828)
Tanisha Maheshwary (102153037)
Kashvi Jain (102153038)
Yuvraj Puri (102153041)

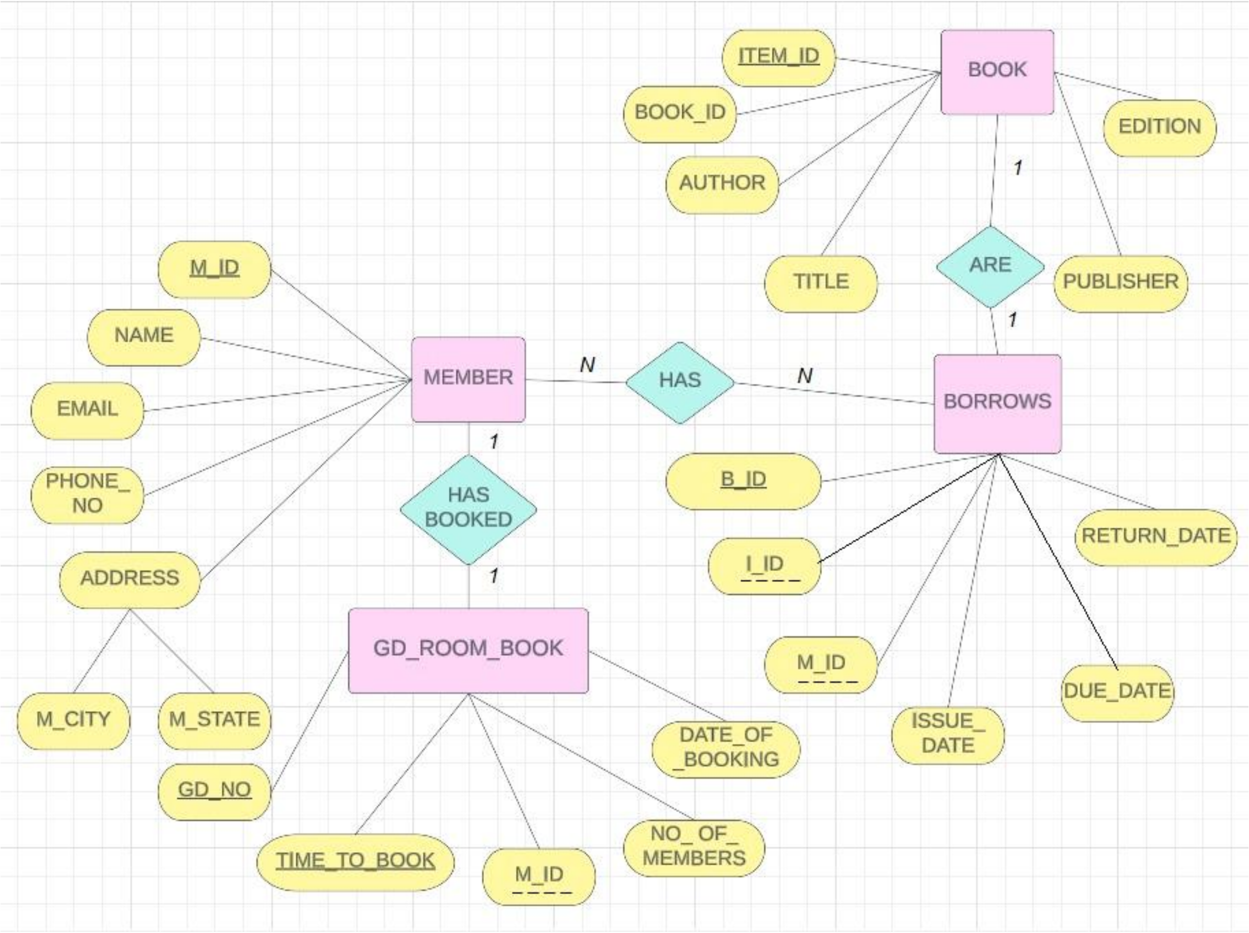
COE28

Index

Problem Statement

Libraries usually house vast quantities of high-quality materials such as books, journals, and e-books, requiring efficient management of members and record-keeping for issue and due dates, as well as fines. Manual management of such tasks can be burdensome. Therefore, we propose a solution to this problem in the form of an SQL-based library management system. This database can significantly reduce the challenges and difficulties associated with manual management of library materials, including tracking due dates and managing member accounts. The proposed solution has the potential to enhance the efficiency of library management, improve service delivery, and provide a better experience for library users.

ER Diagram


ITEM_ID
BOOK
BOOK_ID
EDITION
AUTHOR
1
TITLE
ARE
PUBLISHER
1
M_ID
NAME
MEMBER
N
HAS
N
BORROWS
EMAIL
1
PHONE_NO
HAS BOOKED
B_ID
RETURN_DATE
I_ID
ADDRESS
1
GD_ROOM_BOOK
M_ID
DUE_DATE
M_CITY
M_STATE
DATE_OF_BOOKING
ISSUE_DATE
GD_NO
TIME_TO_BOOK
M_ID
NO_OF_MEMBERS

ER To Table

- Member (m_id, name, email, phone_no, m_city, m_state)
- Book (item_id, book_id, title, edition, author, publisher)
- Borrows (b_id, issue_date, due_date, return_date, m_id, i_id)
- GD_Room_Book (GD_no, Time_to_book, No_of_members, Date_of_booking, m_id)

(Straight line represents primary key and dotted line represents foreign key)

TABLES AT A GLANCE

MEMBER TABLE

m_id	name	email	phone_no	m_city	m_state

Primary Key: m_id

BOOK TABLE

item_id	book_id	title	edition	author	publisher

Primary Key: item_id

BORROWS TABLE

b_id	issue_date	due_date	return_date	m_id	i_id

Primary Key: b_id

Foreign Key: m_id, i_id

GD_ROOM_BOOK TABLE

GD_no	Time_to_book	no_of_members	date_of_booking	m_id

Primary Key: gd_no, time_to_book

Foreign Key: m_id

Normalized Table

- Member1 (m_id, name, email, m_state)
- Member2(m_id, phone_no)
- Member3(m_id, m_city)
- Book (item_id, book_id, title, edition, author, publisher)
- Borrows1(b_id, issue_date, due_date, return_date)
- Borrows2(b_id, m_id, i_id)
- GD_Room_Book (GD_no, Time_to_book, No_of_members, Date_of_booking, m_id)

(Straight line represents primary key and dotted line represents foreign key)

PL/SQL Codes

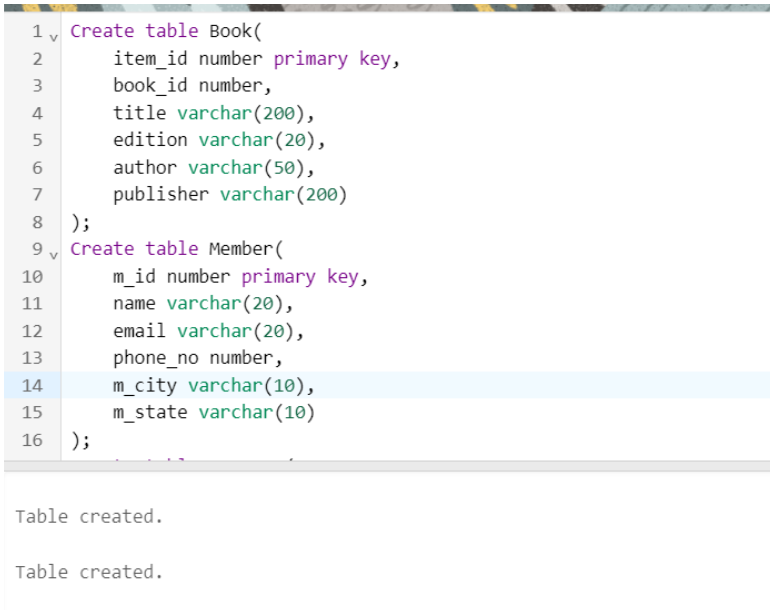
TABLE CREATION:

Table Creation of Book

```
Create table Book(
   item_id number primary key,
   book_id number,
   title varchar(200),
   edition varchar(20),
   author varchar(50),
   publisher varchar(200)
);
```

Table Creation of Member

```
Create table Member(
   m_id number primary key,
   name varchar(20),
   email varchar(20),
   phone_no number,
   m_city varchar(10),
   m_state varchar(10)
);
```



```
Create table Book(
    item_id number primary key,
    book_id number,
    title varchar(200),
    edition varchar(20),
    author varchar(50),
    publisher varchar(200)
);
Create table Member(
    m_id number primary key,
    name varchar(20),
    email varchar(20),
    phone_no number,
    m_city varchar(10),
    m_state varchar(10)
);
```

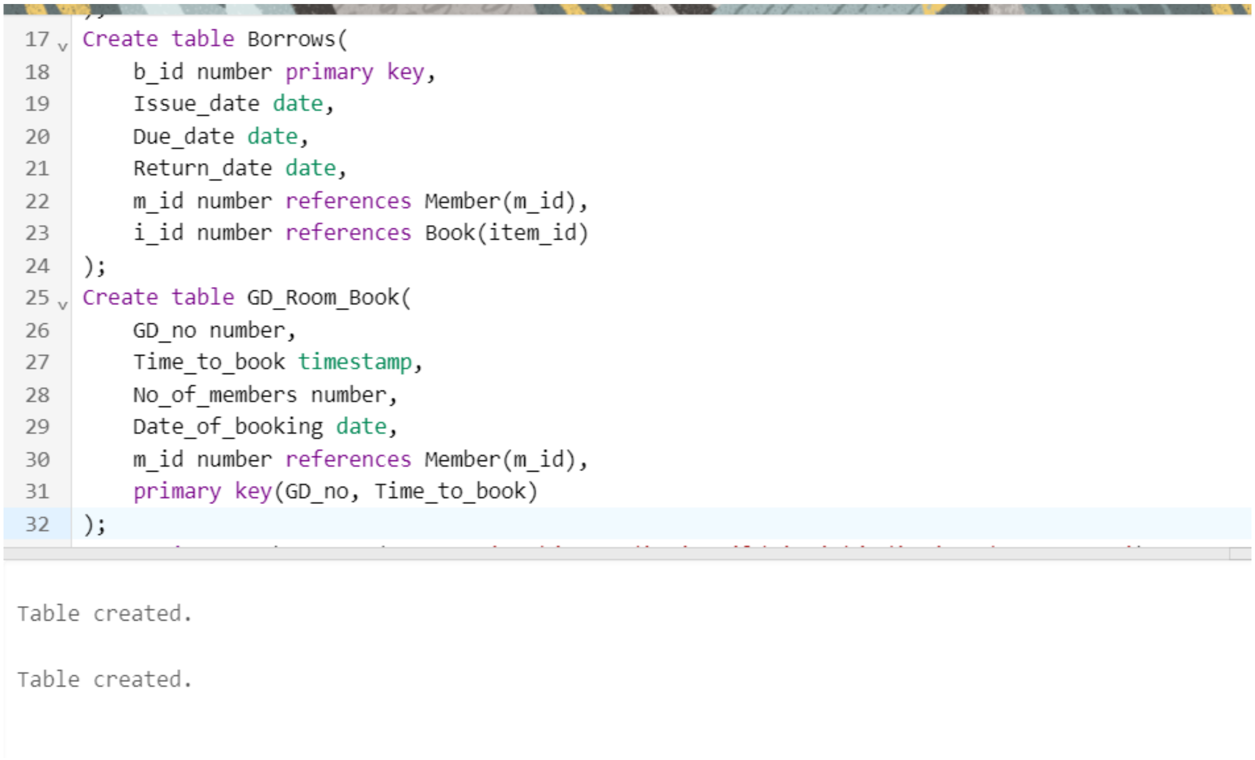
```
Table created.

Table created.
```

Table Creation of Borrows

```
Create table Borrows(
   b_id number primary key,
   Issue_date date,
   Due_date date,
   Return_date date,
   m_id number references Member(m_id),
   i_id number references Book(item_id)
);
```

Table Creation of GD_Room_Book

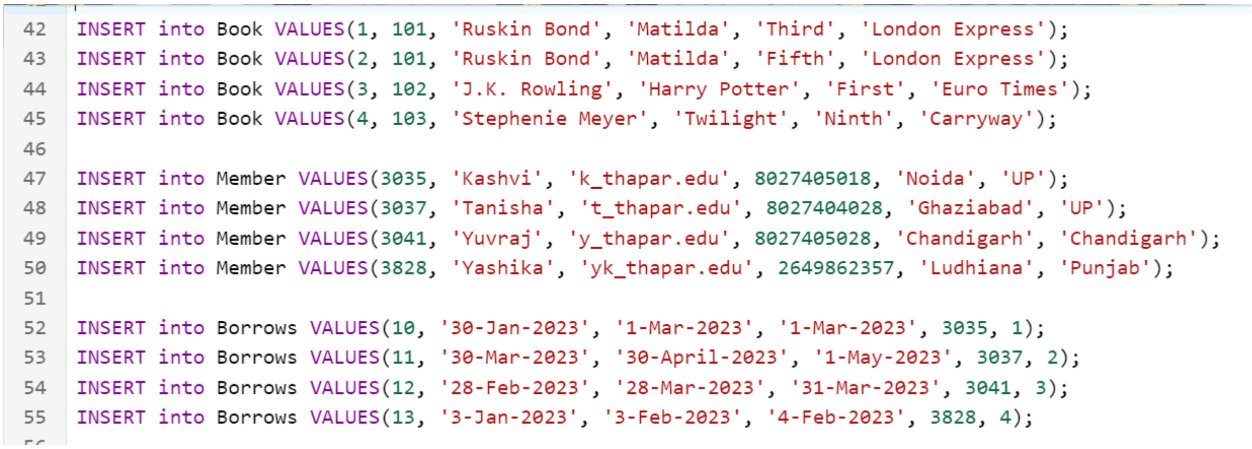
```
Create table GD_Room_Book(
   GD_no number,
   Time_to_book timestamp,
   No_of_members number,
   Date_of_booking date,
   m_id number references Member(m_id),
   primary key(GD_no, Time_to_book)
);
```



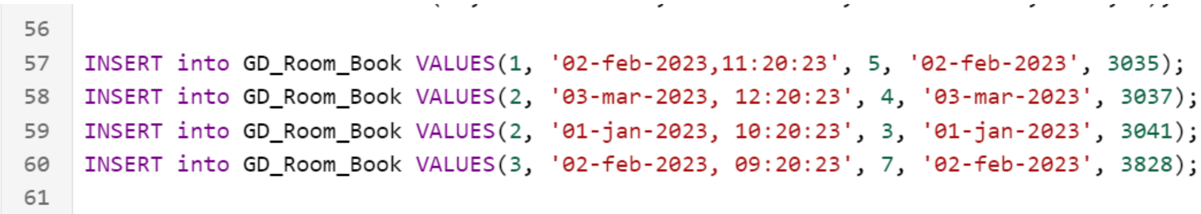
```
Create table Borrows(
    b_id number primary key,
    Issue_date date,
    Due_date date,
    Return_date date,
    m_id number references Member(m_id),
    i_id number references Book(item_id)
);
Create table GD_Room_Book(
    GD_no number,
    Time_to_book timestamp,
    No_of_members number,
    Date_of_booking date,
    m_id number references Member(m_id),
    primary key(GD_no, Time_to_book)
);
```

```
Table created.

Table created.
```

INSERTION OF VALUES INTO THE TABLES:



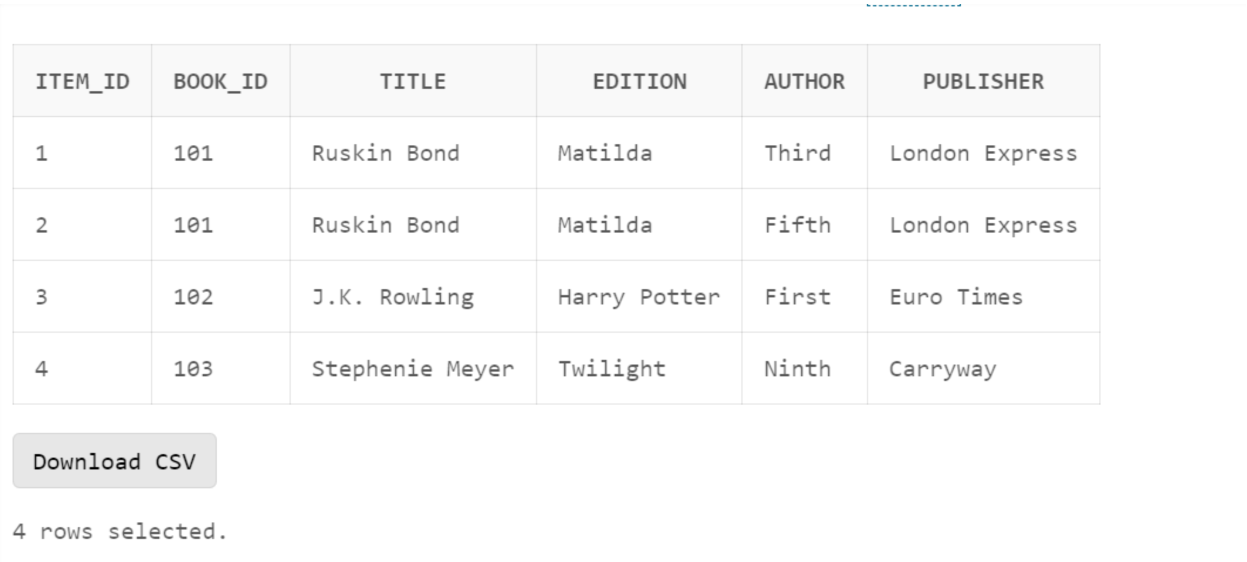
```sql
INSERT into Book VALUES(1, 101, 'Ruskin Bond', 'Matilda', 'Third', 'London Express');
INSERT into Book VALUES(2, 101, 'Ruskin Bond', 'Matilda', 'Fifth', 'London Express');
INSERT into Book VALUES(3, 102, 'J.K. Rowling', 'Harry Potter', 'First', 'Euro Times');
INSERT into Book VALUES(4, 103, 'Stephenie Meyer', 'Twilight', 'Ninth', 'Carryway');

INSERT into Member VALUES(3035, 'Kashvi', 'k_thapar.edu', 8027405018, 'Noida', 'UP');
INSERT into Member VALUES(3037, 'Tanisha', 't_thapar.edu', 8027404028, 'Ghaziabad', 'UP');
INSERT into Member VALUES(3041, 'Yuvraj', 'y_thapar.edu', 8027405028, 'Chandigarh', 'Chandigarh');
INSERT into Member VALUES(3828, 'Yashika', 'yk_thapar.edu', 2649862357, 'Ludhiana', 'Punjab');

INSERT into Borrows VALUES(10, '30-Jan-2023', '1-Mar-2023', '1-Mar-2023', 3035, 1);
INSERT into Borrows VALUES(11, '30-Mar-2023', '30-April-2023', '1-May-2023', 3037, 2);
INSERT into Borrows VALUES(12, '28-Feb-2023', '28-Mar-2023', '31-Mar-2023', 3041, 3);
INSERT into Borrows VALUES(13, '3-Jan-2023', '3-Feb-2023', '4-Feb-2023', 3828, 4);
```



```sql

INSERT into GD_Room_Book VALUES(1, '02-feb-2023,11:20:23', 5, '02-feb-2023', 3035);
INSERT into GD_Room_Book VALUES(2, '03-mar-2023, 12:20:23', 4, '03-mar-2023', 3037);
INSERT into GD_Room_Book VALUES(2, '01-jan-2023, 10:20:23', 3, '01-jan-2023', 3041);
INSERT into GD_Room_Book VALUES(3, '02-feb-2023, 09:20:23', 7, '02-feb-2023', 3828);

```

VIEWING DATASET IN TABLES

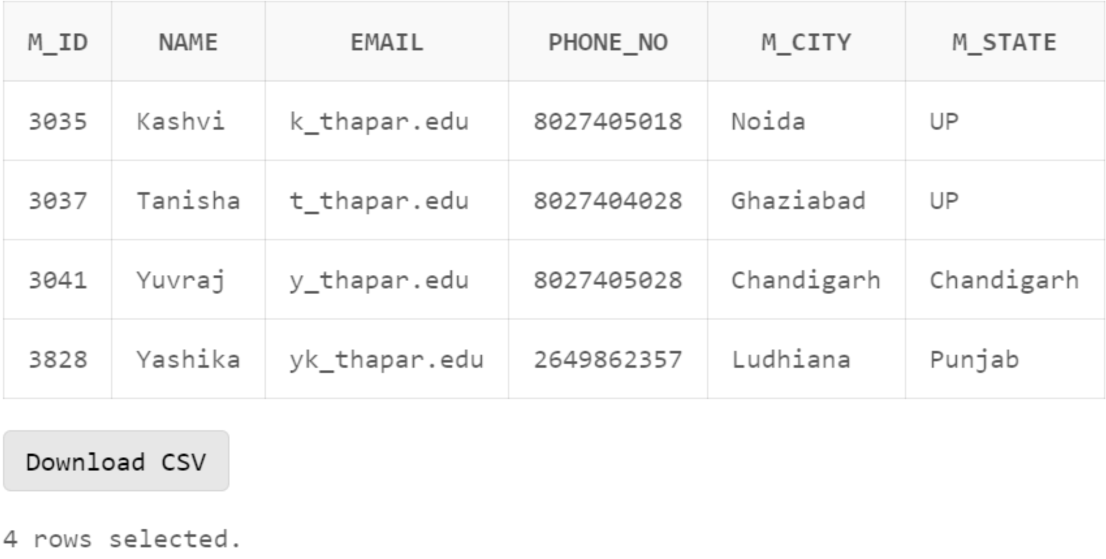
Book Table:



ITEM_ID	BOOK_ID	TITLE	EDITION	AUTHOR	PUBLISHER
1	101	Ruskin Bond	Matilda	Third	London Express
2	101	Ruskin Bond	Matilda	Fifth	London Express
3	102	J.K. Rowling	Harry Potter	First	Euro Times
4	103	Stephenie Meyer	Twilight	Ninth	Carryway

4 rows selected.

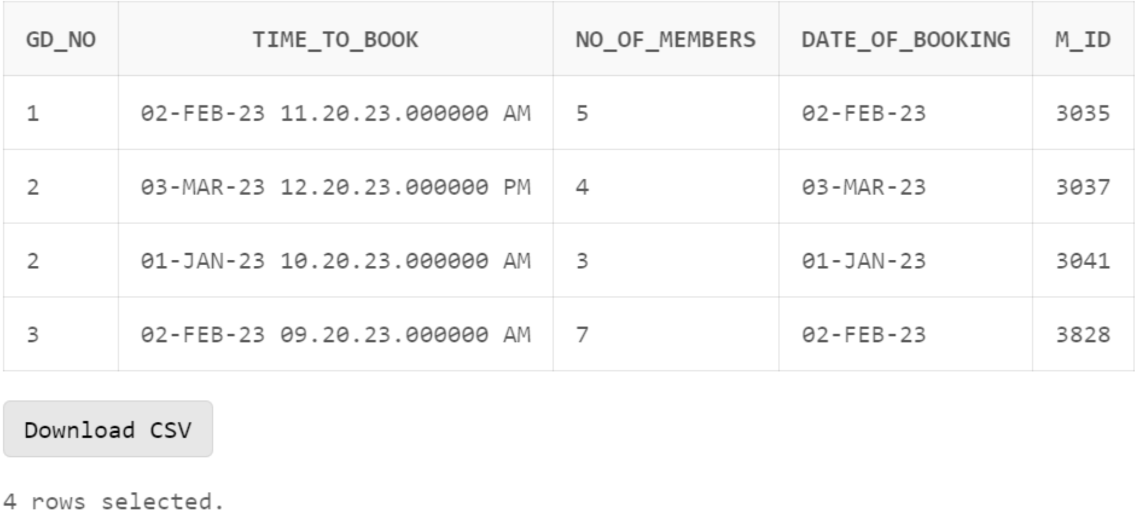
Member Table:



M_ID	NAME	EMAIL	PHONE_NO	M_CITY	M_STATE
3035	Kashvi	k_thapar.edu	8027405018	Noida	UP
3037	Tanisha	t_thapar.edu	8027404028	Ghaziabad	UP
3041	Yuvraj	y_thapar.edu	8027405028	Chandigarh	Chandigarh
3828	Yashika	yk_thapar.edu	2649862357	Ludhiana	Punjab

4 rows selected.

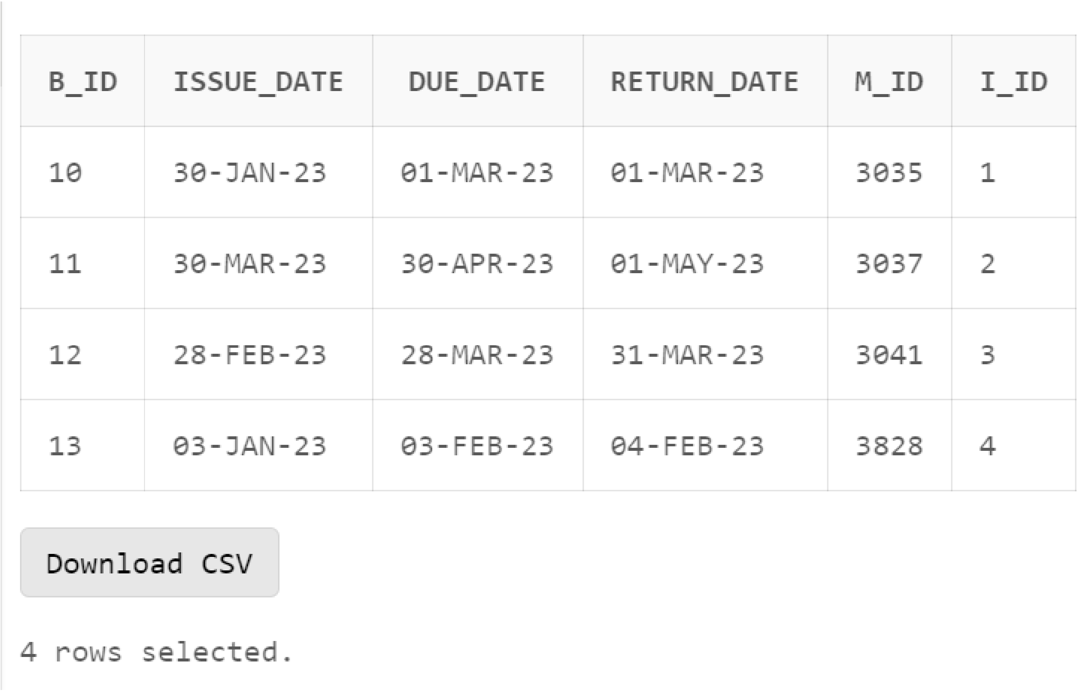
GD_Room_Book Table:



GD_NO	TIME_TO_BOOK	NO_OF_MEMBERS	DATE_OF_BOOKING	M_ID
1	02-FEB-23 11.20.23.000000 AM	5	02-FEB-23	3035
2	03-MAR-23 12.20.23.000000 PM	4	03-MAR-23	3037
2	01-JAN-23 10.20.23.000000 AM	3	01-JAN-23	3041
3	02-FEB-23 09.20.23.000000 AM	7	02-FEB-23	3828

```
4 rows selected.
```

Borrows Table:



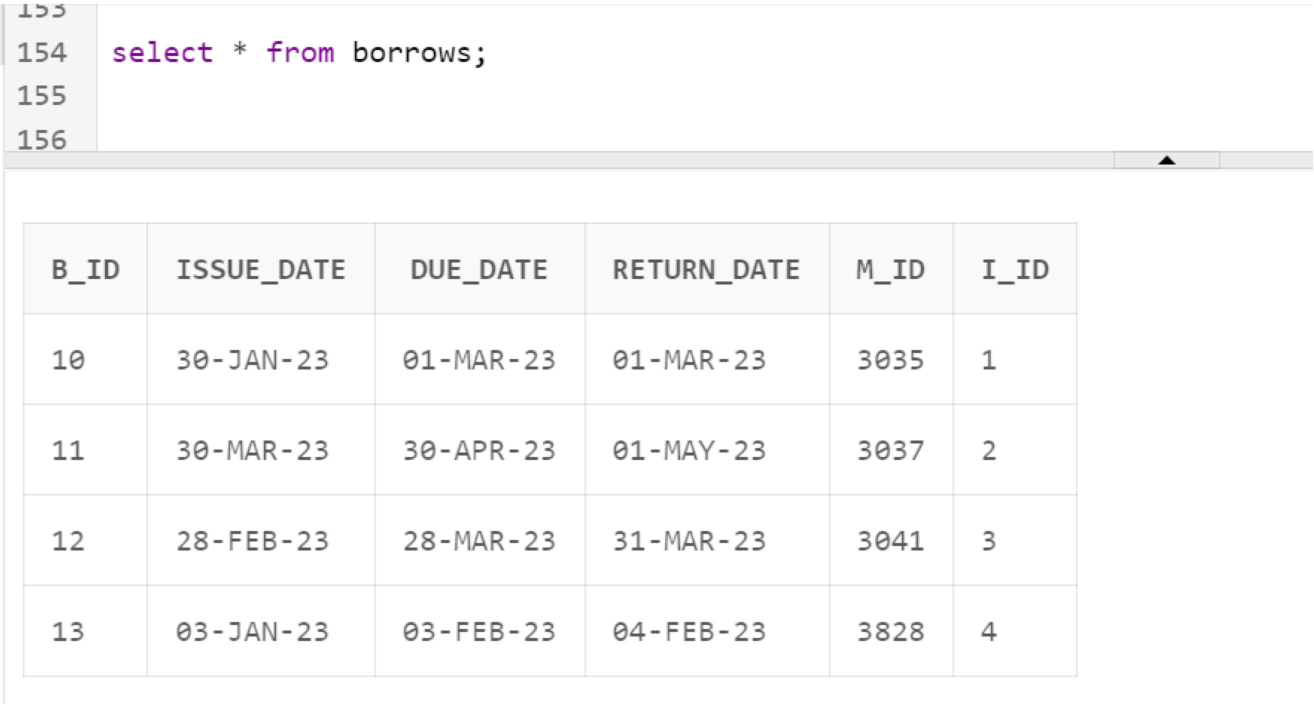
B_ID	ISSUE_DATE	DUE_DATE	RETURN_DATE	M_ID	I_ID
10	30-JAN-23	01-MAR-23	01-MAR-23	3035	1
11	30-MAR-23	30-APR-23	01-MAY-23	3037	2
12	28-FEB-23	28-MAR-23	31-MAR-23	3041	3
13	03-JAN-23	03-FEB-23	04-FEB-23	3828	4

```
4 rows selected.
```

CALCULATING THE FINE (USING FUNCTION):

```
create function calculate_fine (d_issuedate IN date,cal_date IN date,
n_fineperday IN number)
    RETURN NUMBER IS
    n_fine number := 0;
    n_days number:= 0;
BEGIN
    n_days := cal_date-d_issuedate;
    if(n_days>0)
    then
    n_fine := n_days*n_fineperday;
    return n_fine;
    end if;
    return 0;
END;
```

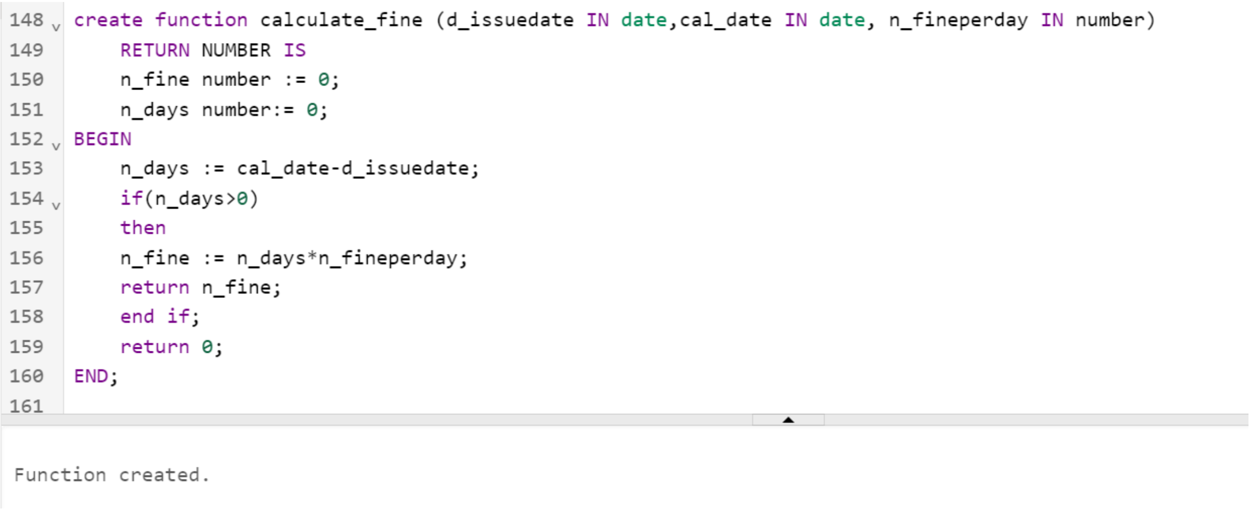
Original contents of Borrows table:



```
select * from borrows;
```

B_ID	ISSUE_DATE	DUE_DATE	RETURN_DATE	M_ID	I_ID
10	30-JAN-23	01-MAR-23	01-MAR-23	3035	1
11	30-MAR-23	30-APR-23	01-MAY-23	3037	2
12	28-FEB-23	28-MAR-23	31-MAR-23	3041	3
13	03-JAN-23	03-FEB-23	04-FEB-23	3828	4

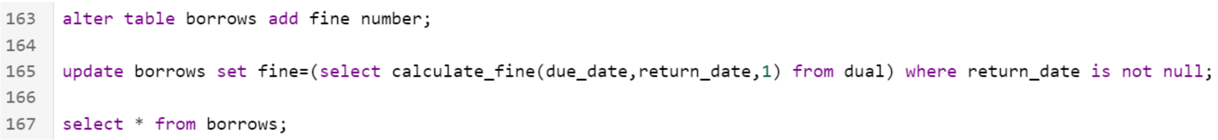
Function Application:



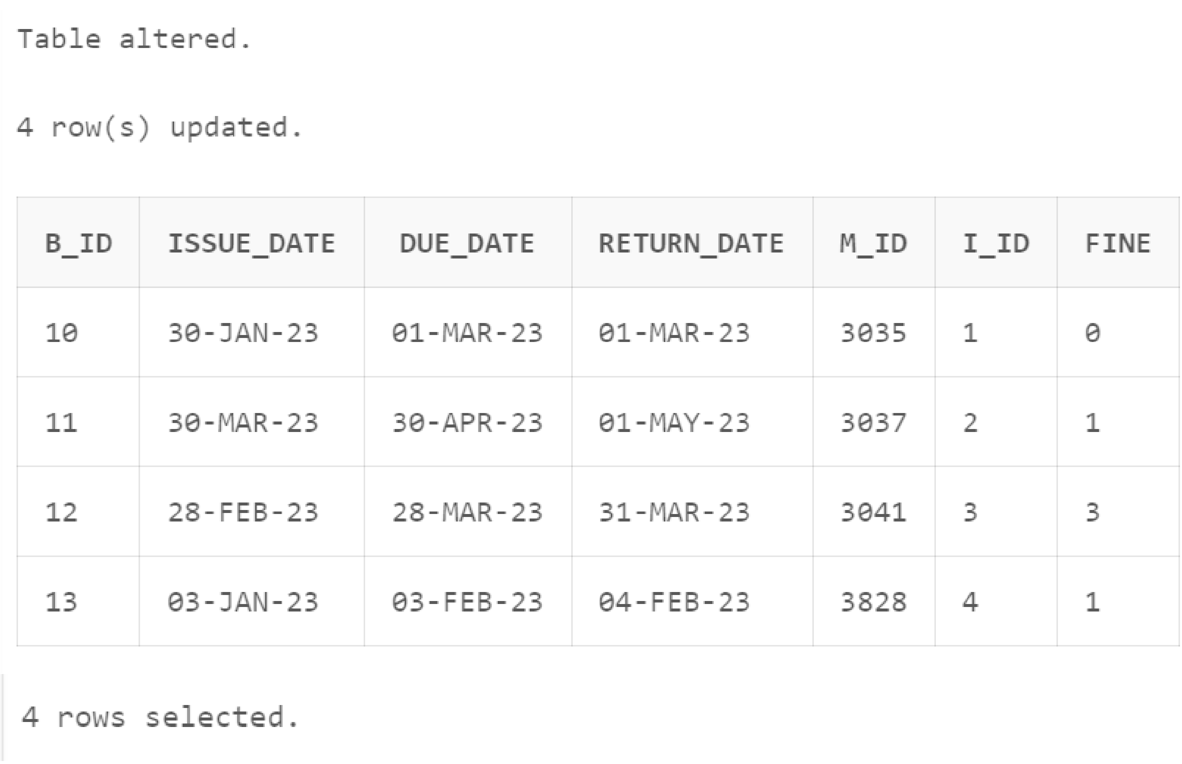
```
create function calculate_fine (d_issuedate IN date,cal_date IN date, n_fineperday IN number)
    RETURN NUMBER IS
    n_fine number := 0;
    n_days number:= 0;
BEGIN
    n_days := cal_date-d_issuedate;
    if(n_days>0)
    then
    n_fine := n_days*n_fineperday;
    return n_fine;
    end if;
    return 0;
END;
```

```
Function created.
```

Final contents of the Borrows table:



```
alter table borrows add fine number;

update borrows set fine=(select calculate_fine(due_date,return_date,1) from dual) where return_date is not null;

select * from borrows;
```



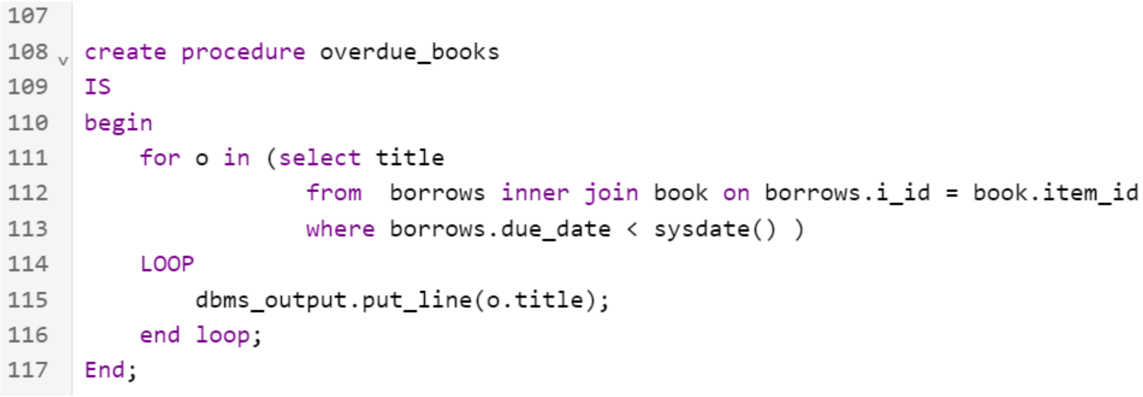
```
Table altered.

4 row(s) updated.
```

B_ID	ISSUE_DATE	DUE_DATE	RETURN_DATE	M_ID	I_ID	FINE
10	30-JAN-23	01-MAR-23	01-MAR-23	3035	1	0
11	30-MAR-23	30-APR-23	01-MAY-23	3037	2	1
12	28-FEB-23	28-MAR-23	31-MAR-23	3041	3	3
13	03-JAN-23	03-FEB-23	04-FEB-23	3828	4	1

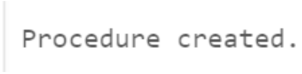
```
4 rows selected.
```

KNOWING OVERDUE BOOKS (USING PROCEDURE):

```
create procedure overdue_books
IS
begin
   for o in (select title
                  from  borrows inner join book on borrows.i_id = book.item_id
                  where borrows.due_date < sysdate() )
   LOOP
      dbms_output.put_line(o.title);
   end loop;
End;
```

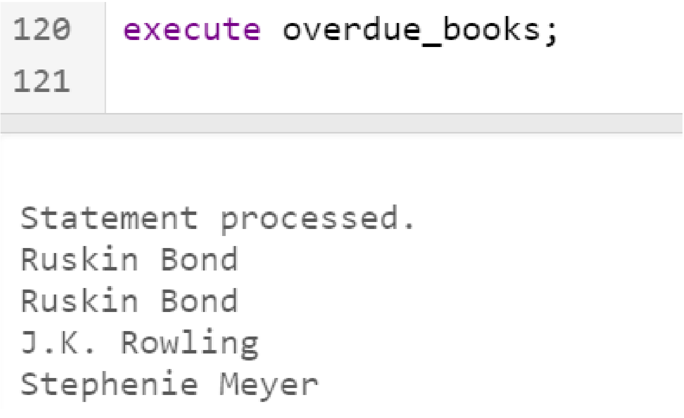


```
create procedure overdue_books
IS
begin
    for o in (select title
                from  borrows inner join book on borrows.i_id = book.item_id
                where borrows.due_date < sysdate() )
    LOOP
        dbms_output.put_line(o.title);
    end loop;
End;
```



```
Procedure created.
```

```
execute overdue_books;
```



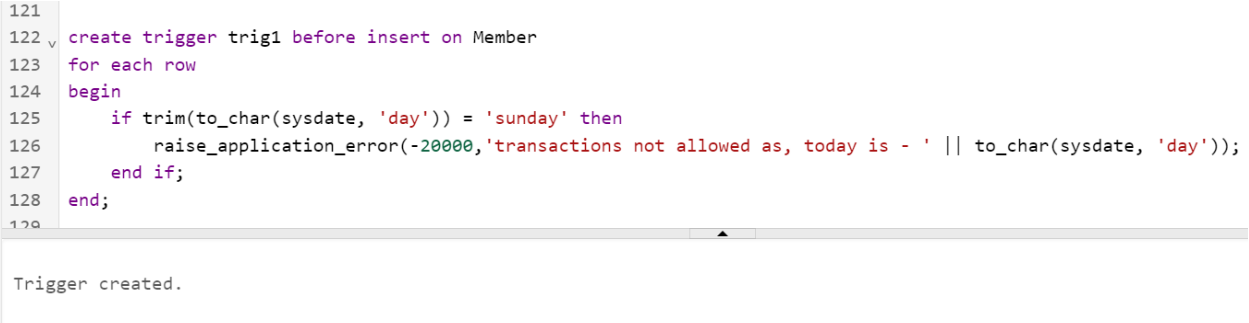
```
execute overdue_books;
```

```
Statement processed.
Ruskin Bond
Ruskin Bond
J.K. Rowling
Stephenie Meyer
```

NO WORK ON SUNDAYS (USING TRIGGERS):

TRIGGER 1:

```
create trigger trig1 before insert on member
for each row
begin
    if trim(to_char(sysdate, 'day')) = 'sunday' then
        raise_application_error(-20000,'transactions not allowed as, today is - '
|| to_char(sysdate, 'day'));
    end if;
end;
```

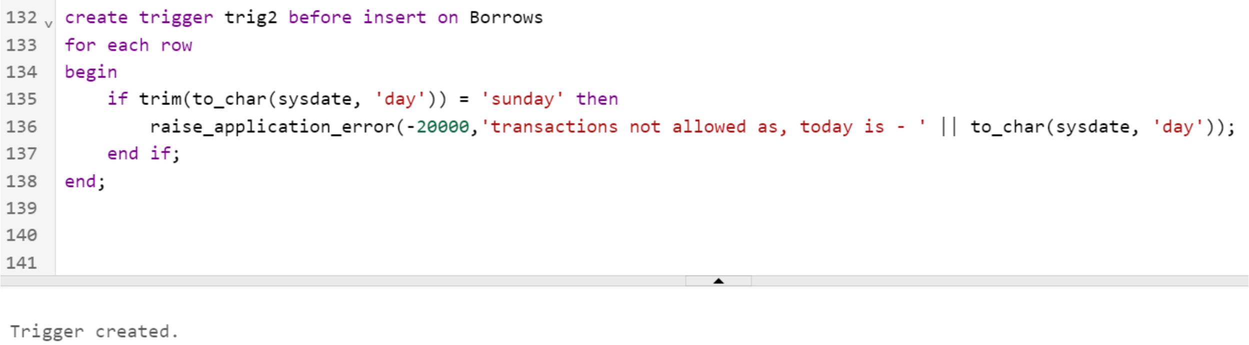


```
create trigger trig1 before insert on Member
for each row
begin
    if trim(to_char(sysdate, 'day')) = 'sunday' then
        raise_application_error(-20000,'transactions not allowed as, today is - ' || to_char(sysdate, 'day'));
    end if;
end;
```

```
Trigger created.
```

TRIGGER 2:

```
create trigger trig2 before insert on Borrows
for each row
begin
     if trim(to_char(sysdate, 'day')) = 'sunday' then
         raise_application_error(-20000,'transactions not allowed as, today is - '
|| to_char(sysdate, 'day'));
    end if;
 end;
```

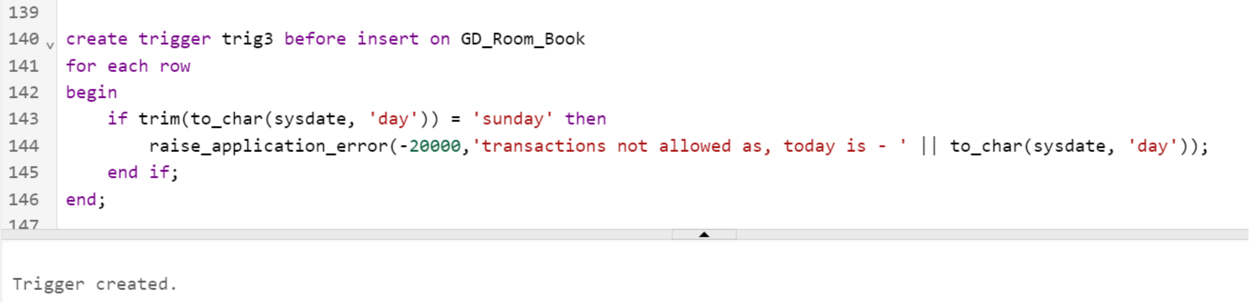


```
create trigger trig2 before insert on Borrows
for each row
begin
    if trim(to_char(sysdate, 'day')) = 'sunday' then
        raise_application_error(-20000,'transactions not allowed as, today is - ' || to_char(sysdate, 'day'));
    end if;
end;
```

```
Trigger created.
```

TRIGGER 3:

```
create trigger trig3 before insert on GD_Room_Book
for each row
begin
   if trim(to_char(sysdate, 'day')) = 'sunday' then
      raise_application_error(-20000,'transactions not allowed as, today is - ' || to_char(sysdate, 'day'));
   end if;
end;
```

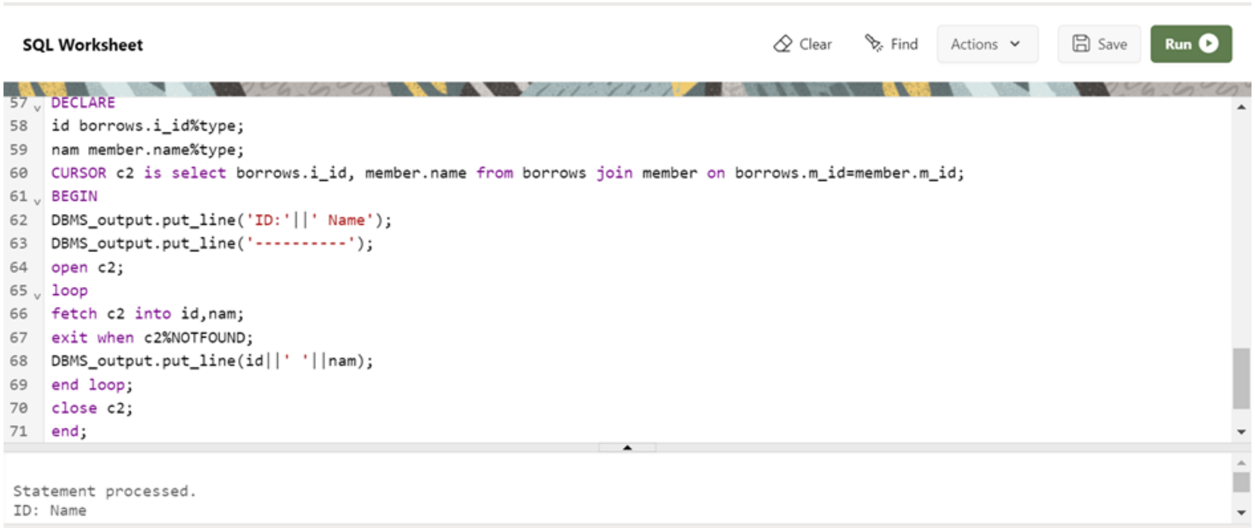


```
create trigger trig3 before insert on GD_Room_Book
for each row
begin
    if trim(to_char(sysdate, 'day')) = 'sunday' then
        raise_application_error(-20000,'transactions not allowed as, today is - ' || to_char(sysdate, 'day'));
    end if;
end;
```

```
Trigger created.
```

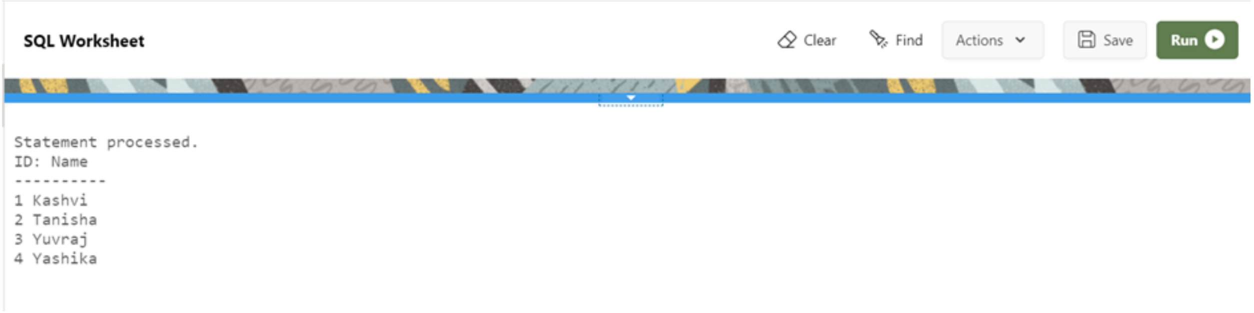
TO SEE WHICH ITEM IS BORROWED BY WHICH MEMBER (CURSOR):

```
DECLARE
        id borrows.i_id%type;
        nam member.name%type;
        CURSOR c2 is select borrows.i_id, member.name from borrows join
        member on borrows.m_id=member.m_id;
BEGIN
        DBMS_output.put_line('ID:'||' Name');
        DBMS_output.put_line('----------');
        open c2;
        loop
        fetch c2 into id,nam;
        exit when c2%NOTFOUND;
        DBMS_output.put_line(id||' '||nam);
        end loop;
        close c2;
end;
```



```
DECLARE
id borrows.i_id%type;
nam member.name%type;
CURSOR c2 is select borrows.i_id, member.name from borrows join member on borrows.m_id=member.m_id;
BEGIN
DBMS_output.put_line('ID:'||' Name');
DBMS_output.put_line('----------');
open c2;
loop
fetch c2 into id,nam;
exit when c2%NOTFOUND;
DBMS_output.put_line(id||' '||nam);
end loop;
close c2;
end;
```

```
Statement processed.
ID: Name
```



```
Statement processed.
ID: Name
----------
1 Kashvi
2 Tanisha
3 Yuvraj
4 Yashika
```

CONCLUSION

In conclusion, the development of a library management system using a database management system has proven to be an effective solution for managing library operations efficiently. The database provides a centralized and organized way to store, access and manage information related to books, library members, items, authentication.

The project was developed using a combination of different technologies, including SQL, PLSQL. Additionally, the system's security features ensure that only authorized users can access the database and its information.

Overall, the library management system developed in this project is a valuable tool for improving library operations and increasing the efficiency of librarians and library members. The system's ability to store, manage and retrieve information in an organized and efficient manner makes it a must-have for any modern library seeking to provide excellent services to its users.

REFERENCES

https://www.edrawsoft.com/article/er-diagrams-for-library-management-system.html

https://www.vidyalayaschoolsoftware.com/blog/2022/07/top-advantages-of-library-management-system/

https://www.academia.edu/37726542/Library_Management_System_Mini_Project_Report_On_LIBRARY_MANAGEMENT_SYSTEM

https://www.tutorialspoint.com/plsql/plsql_cursors.htm#:~:text=A%20cursor%20holds%20the%20rows,statement%2C%20one%20at%20a%20time.

https://www.analyticsvidhya.com/blog/2022/07/library-management-system-using-mysql/