LIBRARY MANAGEMENT SYSTEM

A MINI PROJECT REPORT

18CSC303J-Database Management Systems

Submitted by

SETTEM SAISANDEEP [RA2111003010007]

Under the Guidance of

Dr.M.Senthil Raja

Assistant Professor, Department of Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELORS OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

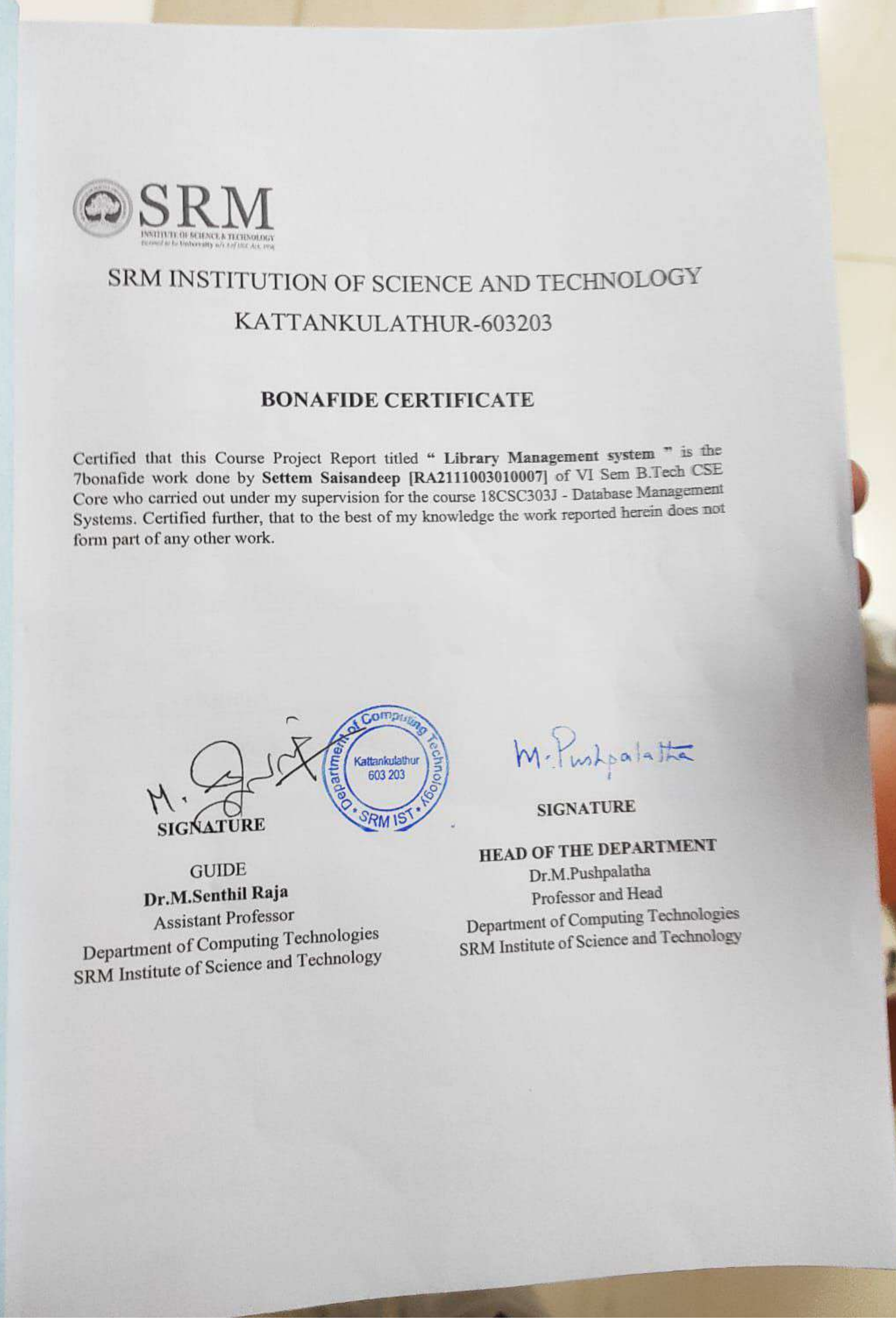
SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

MAY 2024



SRM INSTITUTE OF SCIENCE & TECHNOLOGY

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Course Project Report titled " **Library Management system** " is the 7bonafide work done by **Settem Saisandeep [RA2111003010007]** of VI Sem B.Tech CSE Core who carried out under my supervision for the course 18CSC303J - Database Management Systems. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

[illegible]

Department of Computing Technology
Kattankulathur
603 203
SRM IST

M.Pushpalatha

SIGNATURE

GUIDE
Dr.M.Senthil Raja
Assistant Professor
Department of Computing Technologies
SRM Institute of Science and Technology

SIGNATURE

HEAD OF THE DEPARTMENT
Dr.M.Pushpalatha
Professor and Head
Department of Computing Technologies
SRM Institute of Science and Technology

Content Page

ABSTRACT

Library management system is a project which aims in developing a computerized system to maintain all the daily work of library .This project has many features which are generally not available in normal library management systems like facility of user login and a facility of teachers login .It also has a facility of admin login through which the admin can monitor the whole system .It also has facility of an online notice board where teachers can student can put up information about workshops or seminars being held in our colleges or nearby colleges and librarian after proper verification from the concerned institution organizing the seminar can add it to the notice board . It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date and also the students can request the librarian to add new books by filling the book request form. The librarian after logging into his account i.e., admin account can generate various reports such as student report, issue report, teacher report and book report. Overall, this project of ours is being developed to help the students as well as staff of library to maintain the library in the best way possible and also reduce the human efforts. A library management system is an essential tool for any library to manage its operations efficiently and effectively. The system automates various processes such as book management, borrower management, and loan management, thereby reducing the workload of librarians and improving the overall efficiency of the library. The requirement for a library management system arises due to the increasing number of books, borrowers, and loan transactions in a library. Managing this large volume of data manually can be time-consuming and prone to errors. A library management system provides a centralized platform for storing and managing this data, thereby enabling librarians to access the information easily and quickly.

In addition to providing efficient management of library operations, a library management system also offers various benefits to the library patrons. The system enables patrons to search for books, place requests for books, and track the status of their loan transactions. This results in a more satisfying user experience for library patrons.
Furthermore, a library management system also facilitates effective communication between the library staff and patrons. The system enables librarians to send reminders for overdue books, issue notices for reserved books, and communicate with patrons regarding the status of their requests. The system will include a database to store information about the books, borrowers, and loan transactions.

The database will be designed to enable efficient and easy retrieval of data. The librarian will be able to perform various operations on the books, such as adding a new book, updating the book details, and deleting a book. The system will also enable the librarian to add, edit, and delete borrowers, and assign books to borrowers.

Introduction

A library management system is a software application that is used to manage various operations of a library such as book management, borrower management, loan management, and report generation. The aim of this project is to design and implement a library management system using MySQL.

The system will include a database to store information about the books, borrowers, and loan transactions. The database will be designed to enable efficient and easy retrieval of data. The librarian will be able to perform various operations on the books, such as adding a new book, updating the book details, and deleting a book. The system will also enable the librarian to add, edit, and delete borrowers, and assign books to borrowers. The loan management module will enable the librarian to track the loan status of each book, including the date of issue, the due date, and the return date. The system will automatically calculate the late fees for books that are returned after the due date. The librarian will be able to generate reports on overdue books, books borrowed, and books returned.

Overall, the library management system will simplify and streamline the management of the library's operations. By automating the management of books, borrowers, and loans, the system will enable the librarian to focus on providing better services to the library's patrons. The project will be an essential tool for managing a library effectively and efficiently.

It guides the students to promote their views differently. This knowledge optimizes the student to achieve a better result in academic as well as personal skill development. Improvisation in technology causes the demand for developing a way to enhance the traditional library set up to digital one. Numerous tedious processes reduce the efficiency of the library. For example, it always needs manual support to do any activities in the traditional library. The count and details of books are scribbled in the paper for reference. Each data is fetched in the notebook for future citations. To examine any data then they have to refer the notebooks. At the same time while distributing the books to the students they have to enter into the notebook where they need to represent the book id, distribution and renewal date, and student id. The librarians/staff have to assign a tag for each book and provide an id for it. They have to align and arrange the books on the shelves and marked it. Missing or theft of the book builds a serious issue and confusion to the librarians. While collecting the book from the students they have to verify the penalties of the books. Therefore, it causes a monotonous among the staff.

Consequently, it builds an uninteresting among the student due to the slow progress of the staff. To evoke the library into the technological era, we presented a system called the Library Management system (LMS). It is an automatic system that reduces the work burden of the staff/librarians through a single click. It will manage, organize and oriented the library task. The LMS supports the librarian to add/view/delete/update details from the library stock. Here we integrate all the library data into the SQL server.

ER – Diagram

Description: We will be making an ER diagram for Library Management System.

We have four entities:

Member: Id, Name, Country

Publisher: P_Name, P_ID, Branch, Contact_no

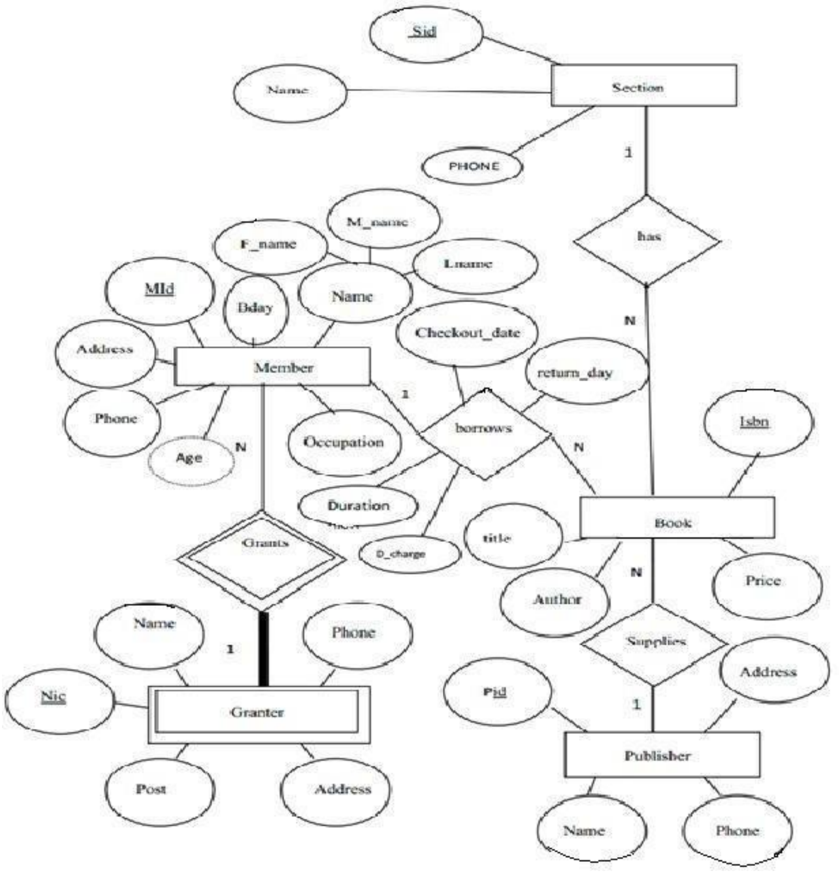
Books: Id, Title, Date_issued, Date_returned, Fine

Section: Sid, Name,Phone

Granter: Post, Names, Address, Phone

Diagram:

3.1Entity Relationship Diagram For library Management System



DDL AND DML – Commands

Here is a small list of DDL and DML commands in MySQL: DDL (Data Definition

Language) Commands:

- CREATE: Used to create a new database, table, index, or view. - ALTER: Used to modify the structure of an existing database, table, or view.
- DROP: Used to delete an existing database, table, index, or view. - TRUNCATE: Used to remove all data from a table without deleting the table itself.
- RENAME: Used to rename an existing database, table, or column. DML (Data Manipulation

  Language) Commands:

- SELECT: Used to retrieve data from one or more tables in a database.
- INSERT: Used to insert new rows of data into a table.
- UPDATE: Used to modify existing rows of data in a table.
- DELETE: Used to delete one or more rows of data from a table.
- MERGE: Used to combine the data from two tables into a single table. - CALL: Used to execute a stored procedure or function.

It is important to note that DDL commands are used to define the database structure, while DML commands are used to manipulate data within the database.

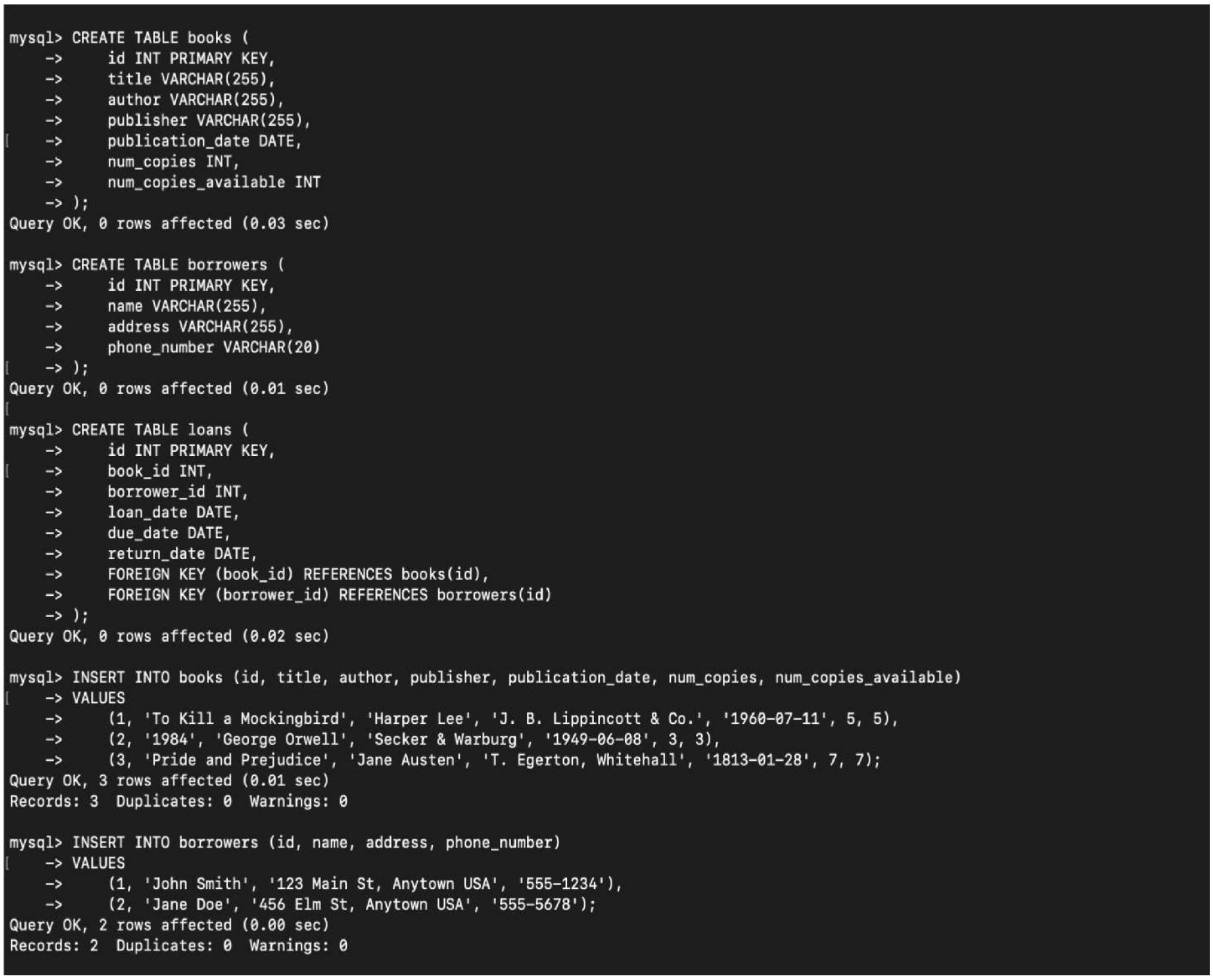
Commands used for the purpose of this project:

```
CREATE TABLE books (    id INT
PRIMARY    KEY,                   title
VARCHAR(255),                   author
VARCHAR(255),               publisher
VARCHAR(255),
publication_date          DATE,                 num_copies
           INT,
num_copies_available INT );
CREATE TABLE borrowers (    id
INT PRIMARY KEY,    name
VARCHAR(255),    address
VARCHAR(255),     phone_number
VARCHAR(20)
);

CREATE TABLE loans (    id INT
PRIMARY KEY,     book_id INT,
borrower_id INT,     loan_date
DATE,     due_date DATE,
    return_date DATE,
    FOREIGN KEY (book_id) REFERENCES books(id),
    FOREIGN KEY (borrower_id) REFERENCES borrowers(id)
);


INSERT         INTO  books  (id,       title,     author,           publisher,        publication_date,
           num_copies, num_copies_available)
VALUES
    (1, 'To Kill a Mockingbird', 'Harper Lee', 'J. B. Lippincott & Co.', '196007-11', 5, 5),
      (2, '1984', 'George Orwell', 'Secker & Warburg', '1949-06-08', 3, 3),
(3, 'Pride and Prejudice', 'Jane Austen', 'T. Egerton, Whitehall', '1813-01-
28', 7, 7);

INSERT INTO borrowers (id, name, address, phone_number) VALUES
    (1, 'John Smith', '123 Main St, Anytown USA', '555-1234'),
    (2, 'Jane Doe', '456 Elm St, Anytown USA', '555-5678');

INSERT INTO loans (id, book_id, borrower_id, loan_date, due_date, return_date)  VALUES
```

(1, 1, 1, '2023-05-01', '2023-05-15', NULL),
(2, 2, 2, '2023-05-05', '2023-05-19', NULL);

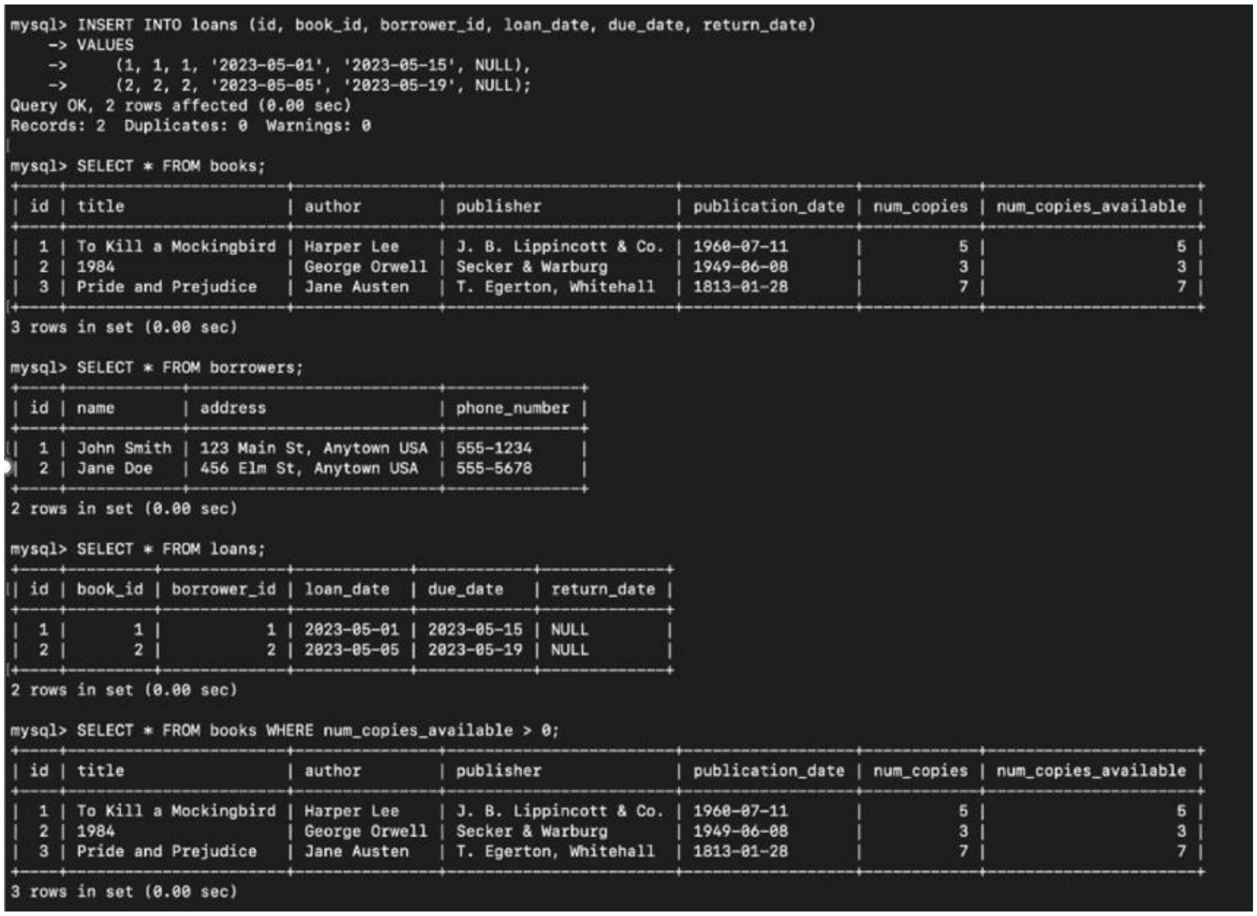
UPDATE books SET num_copies_available = 4 WHERE id = 1;

Output:

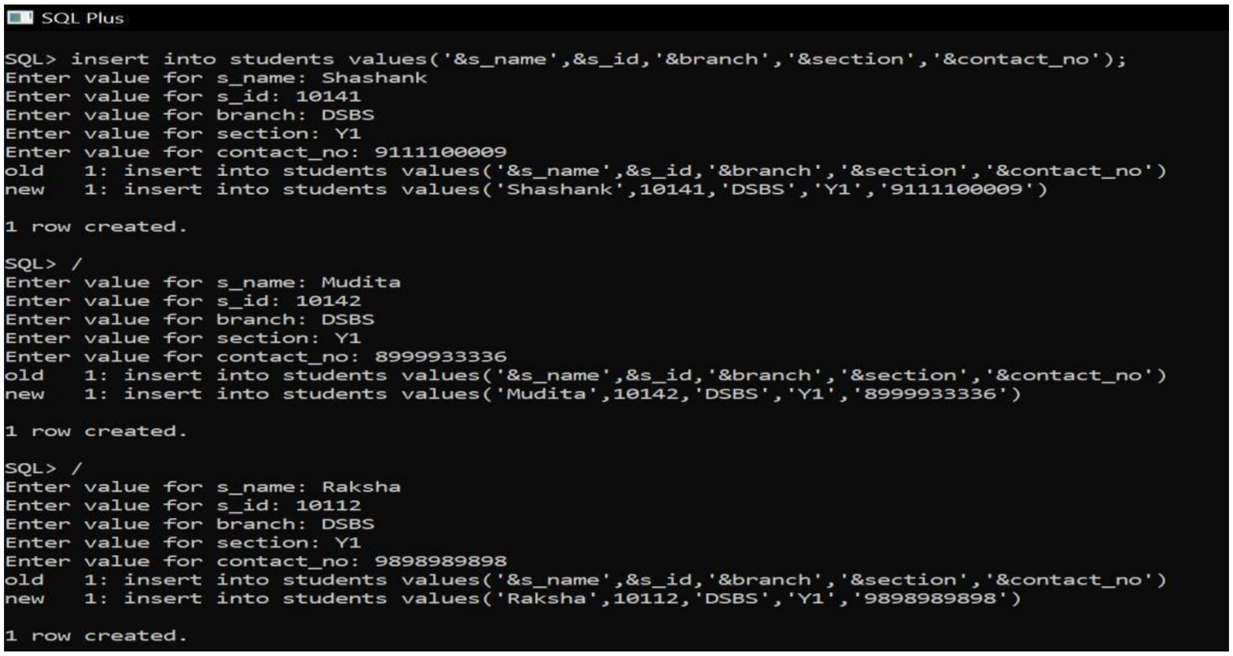


```
mysql> CREATE TABLE books (
    ->     id INT PRIMARY KEY,
    ->     title VARCHAR(255),
    ->     author VARCHAR(255),
    ->     publisher VARCHAR(255),
    ->     publication_date DATE,
    ->     num_copies INT,
    ->     num_copies_available INT
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> CREATE TABLE borrowers (
    ->     id INT PRIMARY KEY,
    ->     name VARCHAR(255),
    ->     address VARCHAR(255),
    ->     phone_number VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE loans (
    ->     id INT PRIMARY KEY,
    ->     book_id INT,
    ->     borrower_id INT,
    ->     loan_date DATE,
    ->     due_date DATE,
    ->     return_date DATE,
    ->     FOREIGN KEY (book_id) REFERENCES books(id),
    ->     FOREIGN KEY (borrower_id) REFERENCES borrowers(id)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO books (id, title, author, publisher, publication_date, num_copies, num_copies_available)
    -> VALUES
    ->     (1, 'To Kill a Mockingbird', 'Harper Lee', 'J. B. Lippincott & Co.', '1960-07-11', 5, 5),
    ->     (2, '1984', 'George Orwell', 'Secker & Warburg', '1949-06-08', 3, 3),
    ->     (3, 'Pride and Prejudice', 'Jane Austen', 'T. Egerton, Whitehall', '1813-01-28', 7, 7);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> INSERT INTO borrowers (id, name, address, phone_number)
    -> VALUES
    ->     (1, 'John Smith', '123 Main St, Anytown USA', '555-1234'),
    ->     (2, 'Jane Doe', '456 Elm St, Anytown USA', '555-5678');
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

UPDATE loans SET return_date = '2023-05-10' WHERE id = 1;

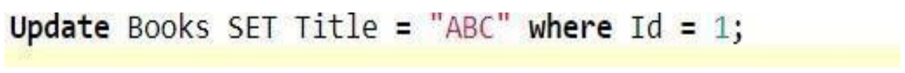


```
mysql> INSERT INTO loans (id, book_id, borrower_id, loan_date, due_date, return_date)
    -> VALUES
    ->     (1, 1, 1, '2023-05-01', '2023-05-15', NULL),
    ->     (2, 2, 2, '2023-05-05', '2023-05-19', NULL);
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM books;
+----+-----------------------+---------------+----------------------+------------------+------------+----------------------+
| id | title                 | author        | publisher            | publication_date | num_copies | num_copies_available |
+----+-----------------------+---------------+----------------------+------------------+------------+----------------------+
|  1 | To Kill a Mockingbird | Harper Lee    | J. B. Lippincott & Co. | 1960-07-11     |          5 |                    5 |
|  2 | 1984                  | George Orwell | Secker & Warburg     | 1949-06-08       |          3 |                    3 |
|  3 | Pride and Prejudice   | Jane Austen   | T. Egerton, Whitehall | 1813-01-28      |          7 |                    7 |
+----+-----------------------+---------------+----------------------+------------------+------------+----------------------+
3 rows in set (0.00 sec)

mysql> SELECT * FROM borrowers;
+----+------------+------------------------+--------------+
| id | name       | address                | phone_number |
+----+------------+------------------------+--------------+
|  1 | John Smith | 123 Main St, Anytown USA | 555-1234   |
|  2 | Jane Doe   | 456 Elm St, Anytown USA  | 555-5678   |
+----+------------+------------------------+--------------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM loans;
+----+---------+-------------+------------+------------+-------------+
| id | book_id | borrower_id | loan_date  | due_date   | return_date |
+----+---------+-------------+------------+------------+-------------+
|  1 |       1 |           1 | 2023-05-01 | 2023-05-15 | NULL        |
|  2 |       2 |           2 | 2023-05-05 | 2023-05-19 | NULL        |
+----+---------+-------------+------------+------------+-------------+
2 rows in set (0.00 sec)

mysql> SELECT * FROM books WHERE num_copies_available > 0;
+----+-----------------------+---------------+----------------------+------------------+------------+----------------------+
| id | title                 | author        | publisher            | publication_date | num_copies | num_copies_available |
+----+-----------------------+---------------+----------------------+------------------+------------+----------------------+
|  1 | To Kill a Mockingbird | Harper Lee    | J. B. Lippincott & Co. | 1960-07-11     |          5 |                    5 |
|  2 | 1984                  | George Orwell | Secker & Warburg     | 1949-06-08       |          3 |                    3 |
|  3 | Pride and Prejudice   | Jane Austen   | T. Egerton, Whitehall | 1813-01-28      |          7 |                    7 |
+----+-----------------------+---------------+----------------------+------------------+------------+----------------------+
3 rows in set (0.00 sec)
```

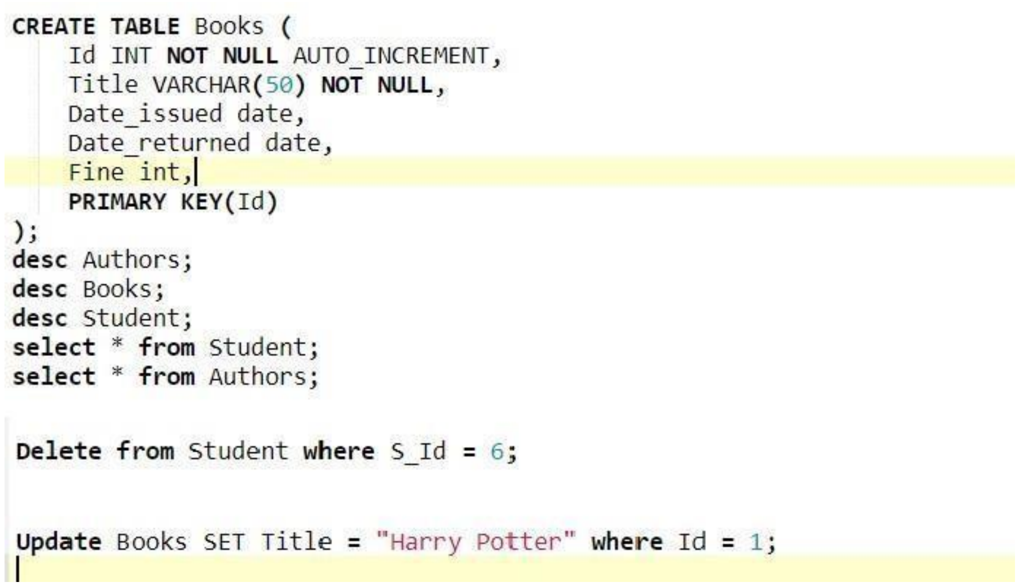


SQL Plus

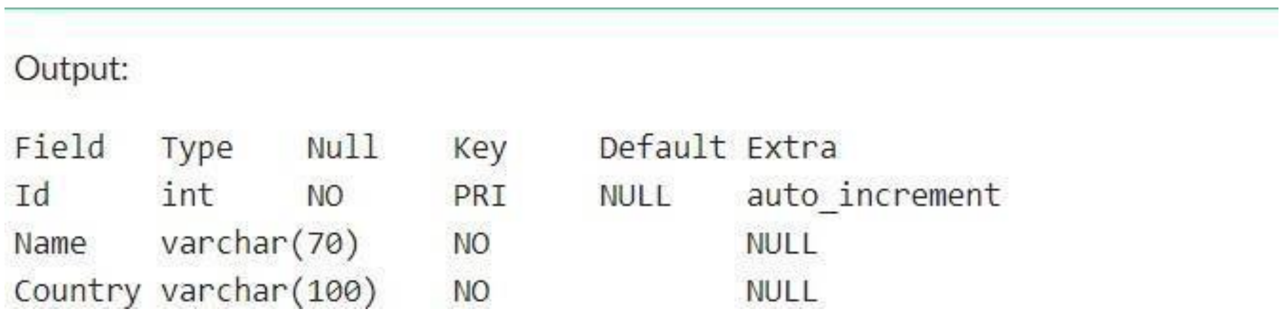
```
SQL> insert into students values('&s_name',&s_id,'&branch','&section','&contact_no');
Enter value for s_name: Shashank
Enter value for s_id: 10141
Enter value for branch: DSBS
Enter value for section: Y1
Enter value for contact_no: 9111100009
old   1: insert into students values('&s_name',&s_id,'&branch','&section','&contact_no')
new   1: insert into students values('Shashank',10141,'DSBS','Y1','9111100009')

1 row created.

SQL> /
Enter value for s_name: Mudita
Enter value for s_id: 10142
Enter value for branch: DSBS
Enter value for section: Y1
Enter value for contact_no: 8999933336
old   1: insert into students values('&s_name',&s_id,'&branch','&section','&contact_no')
new   1: insert into students values('Mudita',10142,'DSBS','Y1','8999933336')

1 row created.

SQL> /
Enter value for s_name: Raksha
Enter value for s_id: 10112
Enter value for branch: DSBS
Enter value for section: Y1
Enter value for contact_no: 9898989898
old   1: insert into students values('&s_name',&s_id,'&branch','&section','&contact_no')
new   1: insert into students values('Raksha',10112,'DSBS','Y1','9898989898')

1 row created.
```



```
Update Books SET Title = "ABC" where Id = 1;
```

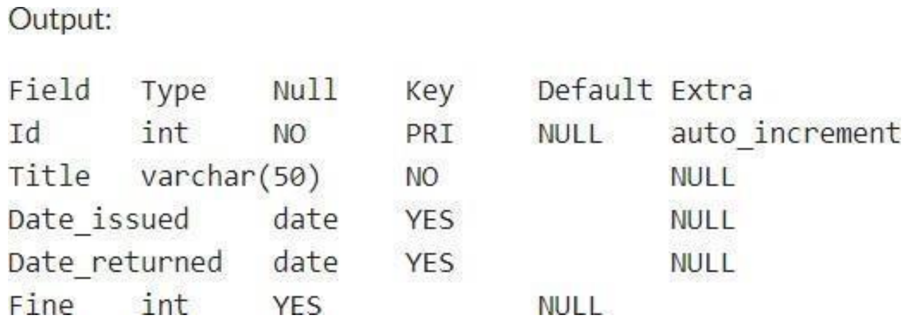


```
CREATE TABLE Books (
    Id INT NOT NULL AUTO_INCREMENT,
    Title VARCHAR(50) NOT NULL,
    Date_issued date,
    Date_returned date,
    Fine int,
    PRIMARY KEY(Id)
);
desc Authors;
desc Books;
desc Student;
select * from Student;
select * from Authors;

Delete from Student where S_Id = 6;


Update Books SET Title = "Harry Potter" where Id = 1;
```



Output:

```
Field   Type         Null    Key     Default Extra
Id      int          NO      PRI     NULL    auto_increment
Name    varchar(70)          NO              NULL
Country varchar(100)         NO              NULL
```

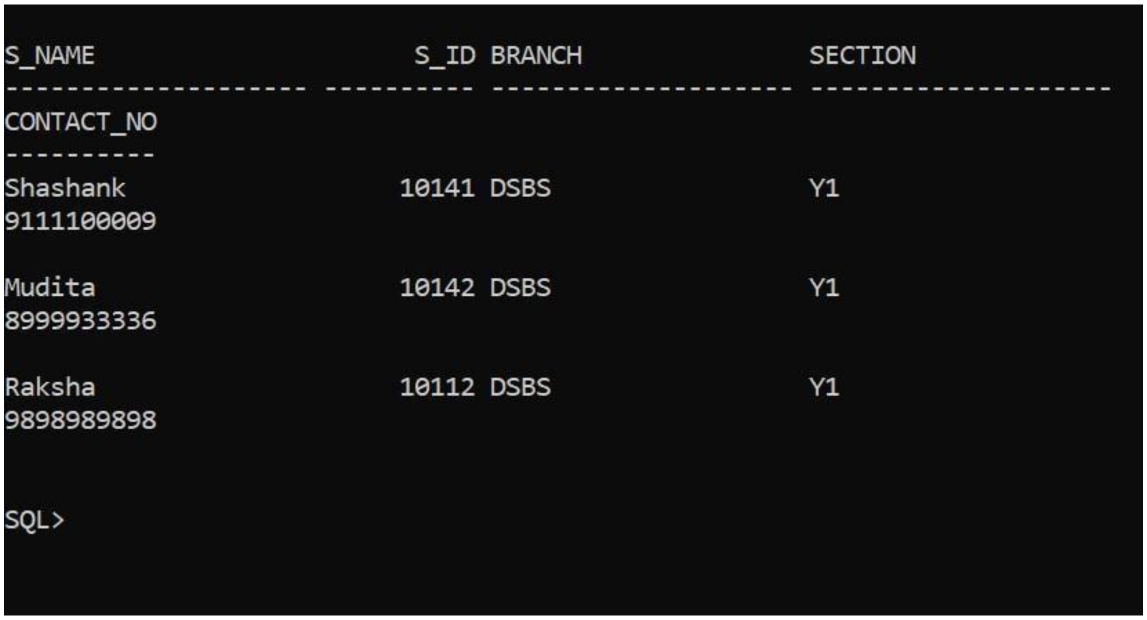


Output:

```
Field    Type        Null    Key     Default Extra
Id       int         NO      PRI     NULL    auto_increment
Title    varchar(50)         NO              NULL
Date_issued          date    YES             NULL
Date_returned        date    YES             NULL
Fine     int         YES             NULL
```

Output:

```
Field   Type     Null    Key     Default Extra
S_Name  varchar(25)      NO              NULL
S_ID    int      NO      PRI     NULL
Branch  varchar(20)      YES             NULL
Section varchar(3)       YES             NULL
Contact_no       bigint  YES             NULL
```



```
S_NAME                         S_ID BRANCH               SECTION
------------------------- ---------- -------------------- --------------------
CONTACT_NO
----------
Shashank                      10141 DSBS                 Y1
9111100009

Mudita                        10142 DSBS                 Y1
8999933336

Raksha                        10112 DSBS                 Y1
9898989898


SQL>
```

```
Id      Name     Country
1       J.D. Salinger   USA
2       F. Scott. Fitzgerald    USA
3       Jane Austen     UK
4       Scott Hanselman USA
5       Jason N. Gaylord        USA
```

TCL – Commands

Description:

TCL (Transaction Control Language)

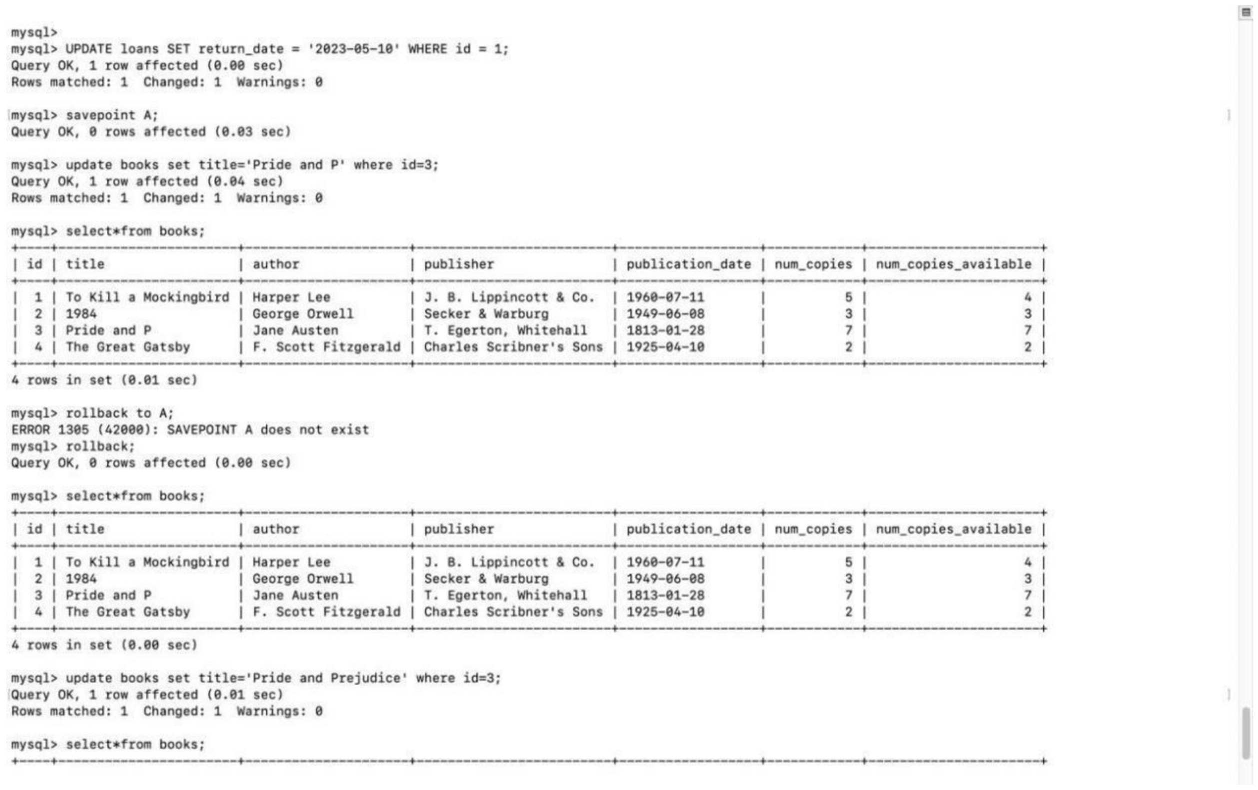
A single unit of work in a database is formed after the consecutive execution of commands is known as a transaction. There are certain commands present in SQL known as TCL commands that help the user manage the transactions that take place in a database.

TCL commands are:

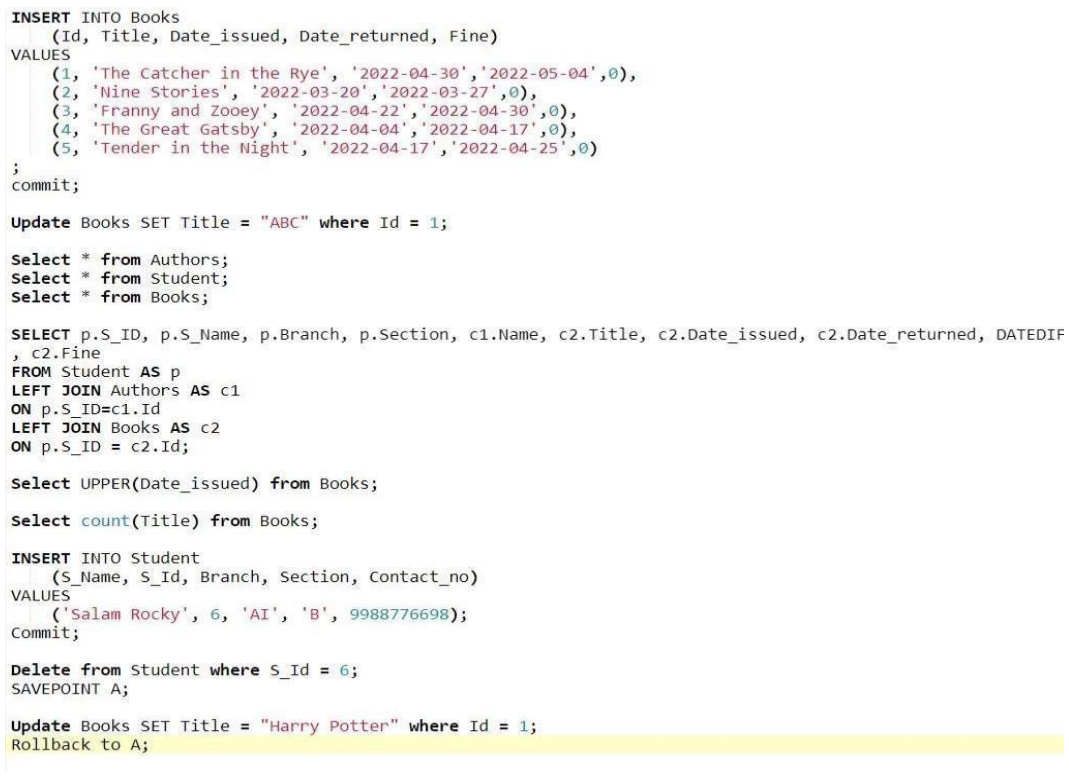
COMMIT - used to save all the transactions to the database.

ROLLBACK - used to undo transactions that have not already been saved.

SAVEPOINT - used to roll the transaction back to a certain point. **Code and Output:**

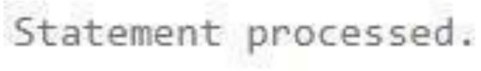


```
mysql>
mysql> UPDATE loans SET return_date = '2023-05-10' WHERE id = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> savepoint A;
Query OK, 0 rows affected (0.03 sec)

mysql> update books set title='Pride and P' where id=3;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select*from books;
+----+-----------------------+--------------------+-------------------------+------------------+------------+----------------------+
| id | title                 | author             | publisher               | publication_date | num_copies | num_copies_available |
+----+-----------------------+--------------------+-------------------------+------------------+------------+----------------------+
|  1 | To Kill a Mockingbird | Harper Lee         | J. B. Lippincott & Co.  | 1960-07-11       |          5 |                    4 |
|  2 | 1984                  | George Orwell      | Secker & Warburg        | 1949-06-08       |          3 |                    3 |
|  3 | Pride and P           | Jane Austen        | T. Egerton, Whitehall   | 1813-01-28       |          7 |                    7 |
|  4 | The Great Gatsby      | F. Scott Fitzgerald | Charles Scribner's Sons | 1925-04-10       |          2 |                    2 |
+----+-----------------------+--------------------+-------------------------+------------------+------------+----------------------+
4 rows in set (0.01 sec)

mysql> rollback to A;
ERROR 1305 (42000): SAVEPOINT A does not exist
mysql> rollback;
Query OK, 0 rows affected (0.00 sec)

mysql> select*from books;
+----+-----------------------+--------------------+-------------------------+------------------+------------+----------------------+
| id | title                 | author             | publisher               | publication_date | num_copies | num_copies_available |
+----+-----------------------+--------------------+-------------------------+------------------+------------+----------------------+
|  1 | To Kill a Mockingbird | Harper Lee         | J. B. Lippincott & Co.  | 1960-07-11       |          5 |                    4 |
|  2 | 1984                  | George Orwell      | Secker & Warburg        | 1949-06-08       |          3 |                    3 |
|  3 | Pride and P           | Jane Austen        | T. Egerton, Whitehall   | 1813-01-28       |          7 |                    7 |
|  4 | The Great Gatsby      | F. Scott Fitzgerald | Charles Scribner's Sons | 1925-04-10       |          2 |                    2 |
+----+-----------------------+--------------------+-------------------------+------------------+------------+----------------------+
4 rows in set (0.00 sec)

mysql> update books set title='Pride and Prejudice' where id=3;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select*from books;
+----+-----------------------+--------------------+-------------------------+------------------+------------+----------------------+
```



```
INSERT INTO Books
    (Id, Title, Date_issued, Date_returned, Fine)
VALUES
    (1, 'The Catcher in the Rye', '2022-04-30','2022-05-04',0),
    (2, 'Nine Stories', '2022-03-20','2022-03-27',0),
    (3, 'Franny and Zooey', '2022-04-22','2022-04-30',0),
    (4, 'The Great Gatsby', '2022-04-04','2022-04-17',0),
    (5, 'Tender in the Night', '2022-04-17','2022-04-25',0)
;
commit;

Update Books SET Title = "ABC" where Id = 1;

Select * from Authors;
Select * from Student;
Select * from Books;

SELECT p.S_ID, p.S_Name, p.Branch, p.Section, c1.Name, c2.Title, c2.Date_issued, c2.Date_returned, DATEDIF
, c2.Fine
FROM Student AS p
LEFT JOIN Authors AS c1
ON p.S_ID=c1.Id
LEFT JOIN Books AS c2
ON p.S_ID = c2.Id;

Select UPPER(Date_issued) from Books;

Select count(Title) from Books;

INSERT INTO Student
    (S_Name, S_Id, Branch, Section, Contact_no)
VALUES
    ('Salam Rocky', 6, 'AI', 'B', 9988776698);
Commit;

Delete from Student where S_Id = 6;
SAVEPOINT A;

Update Books SET Title = "Harry Potter" where Id = 1;
Rollback to A;
```

OUTPUT:

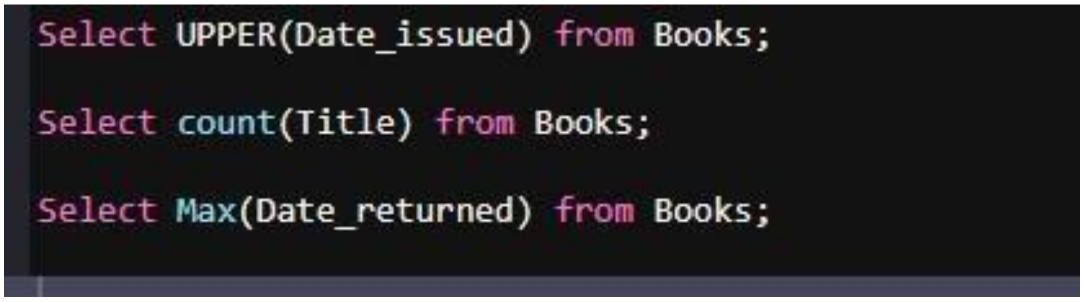


```
Statement processed.
```

INBUILT - Commands

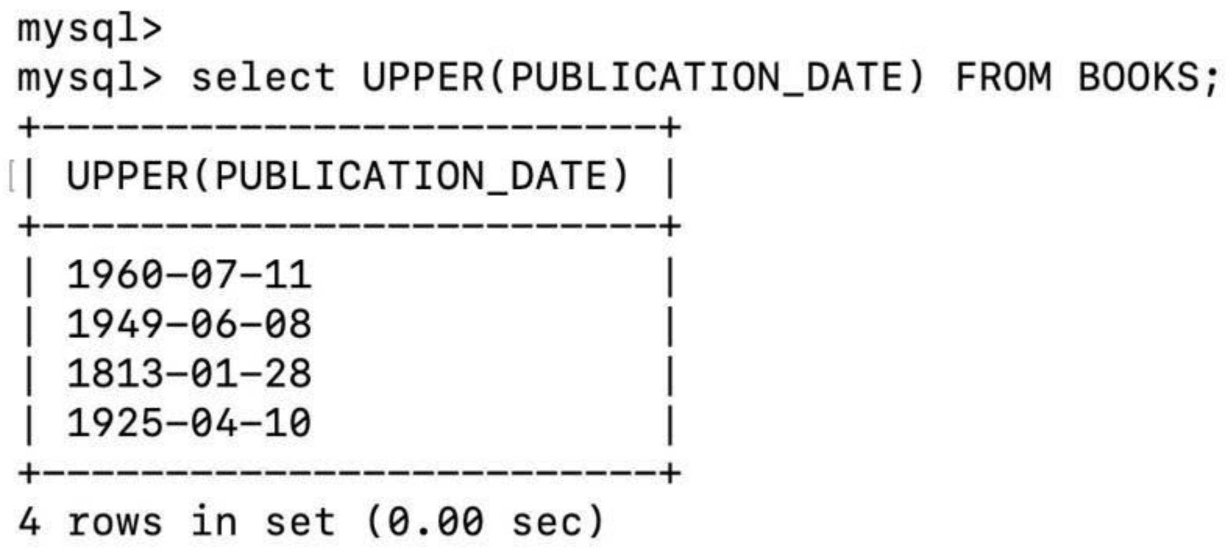
Description: Inbuild functions take an entire column or an entire row and output a single value or null value. Some examples of aggregate functions are sum(), max(), min(), avg() etc.

Code:

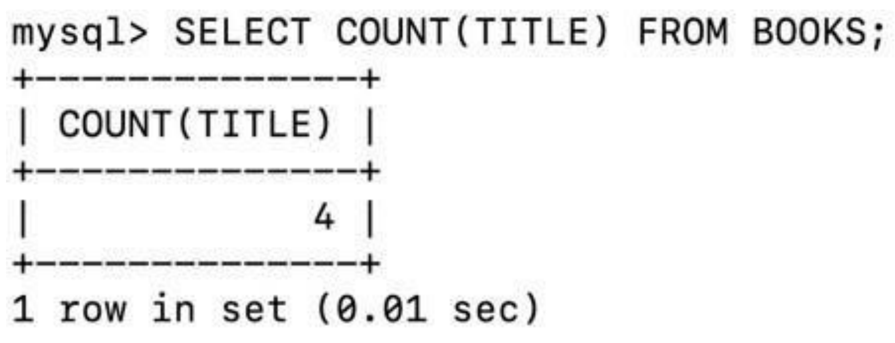


```
Select UPPER(Date_issued) from Books;

Select count(Title) from Books;

Select Max(Date_returned) from Books;
```

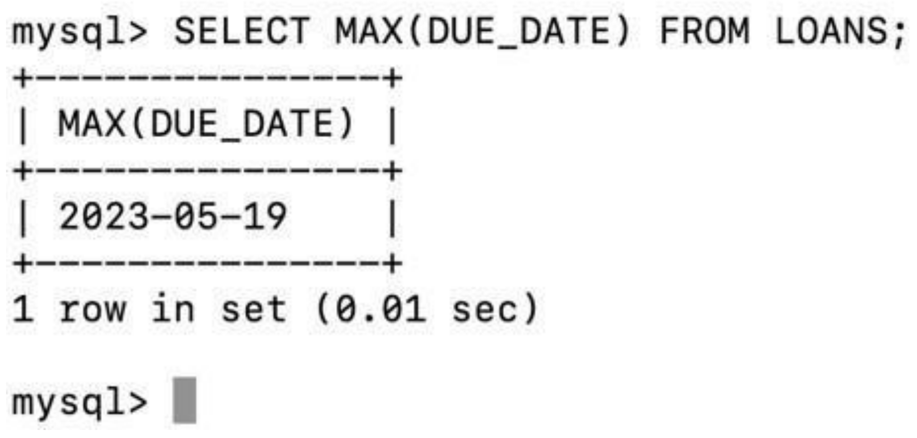
OUTPUT:



```
mysql>
mysql> select UPPER(PUBLICATION_DATE) FROM BOOKS;
+-------------------------+
| UPPER(PUBLICATION_DATE) |
+-------------------------+
| 1960-07-11              |
| 1949-06-08              |
| 1813-01-28              |
| 1925-04-10              |
+-------------------------+
4 rows in set (0.00 sec)
```



```
mysql> SELECT COUNT(TITLE) FROM BOOKS;
+--------------+
| COUNT(TITLE) |
+--------------+
|            4 |
+--------------+
1 row in set (0.01 sec)
```

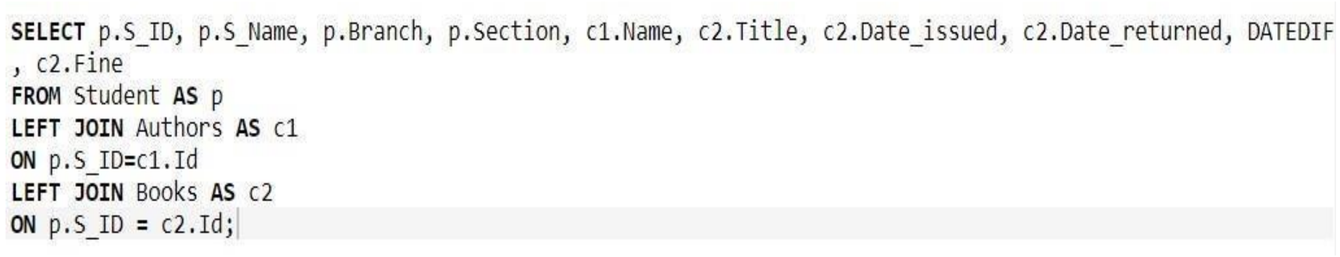


```
mysql> SELECT MAX(DUE_DATE) FROM LOANS;
+---------------+
| MAX(DUE_DATE) |
+---------------+
| 2023-05-19    |
+---------------+
1 row in set (0.01 sec)

mysql> 
```

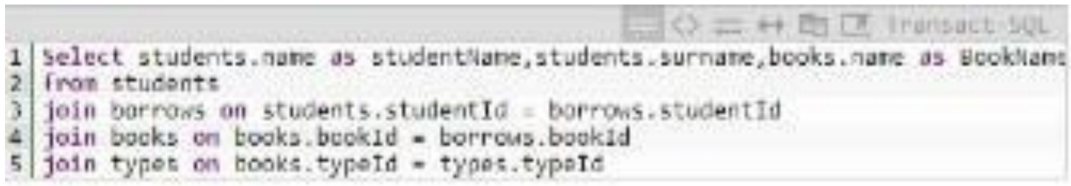
JOINS

Description: There are 4 Types of joins: Inner, left, right, full join.

Code and Output:

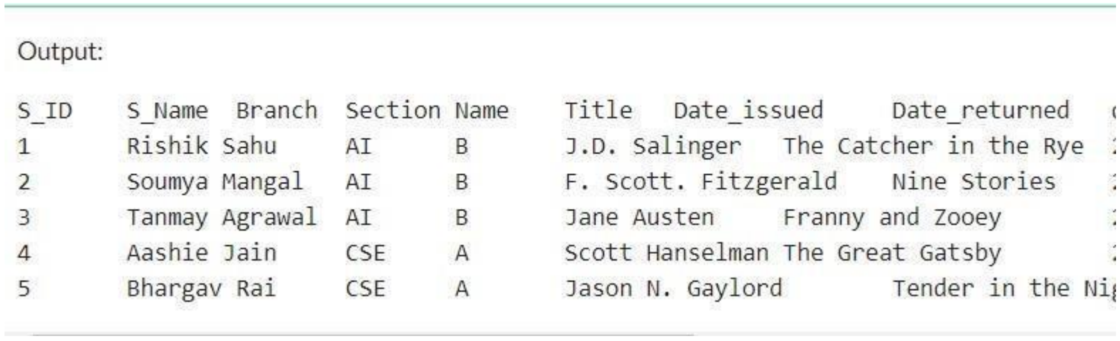


```
SELECT p.S_ID, p.S_Name, p.Branch, p.Section, c1.Name, c2.Title, c2.Date_issued, c2.Date_returned, DATEDIF
, c2.Fine
FROM Student AS p
LEFT JOIN Authors AS c1
ON p.S_ID=c1.Id
LEFT JOIN Books AS c2
ON p.S_ID = c2.Id;
```

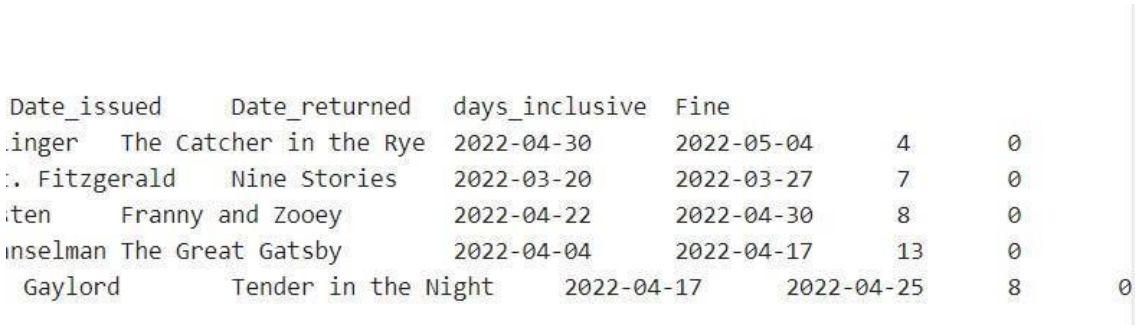


```
Select students.name as studentName,students.surname,books.name as BookName
from students
join borrows on students.studentId = borrows.studentId
join books on books.bookId = borrows.bookId
join types on books.typeId = types.typeId
```

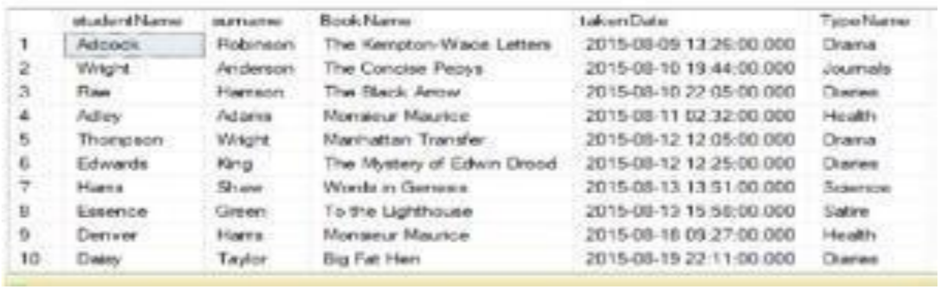
Output:



```
Output:

S_ID    S_Name   Branch   Section Name    Title    Date_issued     Date_returned   (
1       Rishik Sahu     AI       B       J.D. Salinger   The Catcher in the Rye  :
2       Soumya Mangal   AI       B       F. Scott. Fitzgerald    Nine Stories    :
3       Tanmay Agrawal  AI       B       Jane Austen     Franny and Zooey        :
4       Aashie Jain     CSE      A       Scott Hanselman The Great Gatsby        :
5       Bhargav Rai     CSE      A       Jason N. Gaylord        Tender in the Ni
```



```
Date_issued     Date_returned   days_inclusive  Fine
.inger   The Catcher in the Rye  2022-04-30      2022-05-04      4       0
:. Fitzgerald    Nine Stories    2022-03-20      2022-03-27      7       0
;ten     Franny and Zooey        2022-04-22      2022-04-30      8       0
nselman The Great Gatsby        2022-04-04      2022-04-17      13      0
 Gaylord        Tender in the Night     2022-04-17      2022-04-25      8       0
```



	studentName	surname	BookName	takenDate	TypeName
1	Adcock	Robinson	The Kempton-Wace Letters	2015-08-09 13:26:00.000	Drama
2	Wright	Anderson	The Concise Pepys	2015-08-10 19:44:00.000	Journals
3	Raw	Harrison	The Black Arrow	2015-08-10 22:05:00.000	Diaries
4	Adley	Adams	Monsieur Maurice	2015-08-11 02:32:00.000	Health
5	Thompson	Wright	Manhattan Transfer	2015-08-12 12:05:00.000	Drama
6	Edwards	King	The Mystery of Edwin Drood	2015-08-12 12:25:00.000	Diaries
7	Harris	Shaw	Words in Genesis	2015-08-13 13:51:00.000	Science
8	Essence	Green	To the Lighthouse	2015-08-13 15:58:00.000	Satire
9	Denver	Harris	Monsieur Maurice	2015-08-18 09:27:00.000	Health
10	Daisy	Taylor	Big Fat Hen	2015-08-19 22:11:00.000	Diaries

NESTED QUERIES

Description:

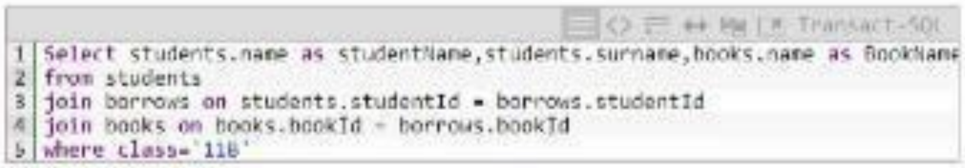
A Subquery or Inner query or a Nested query is a query within another SQL query and embedded within the WHERE clause.

A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

Subqueries can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN, etc.

Code and Output:

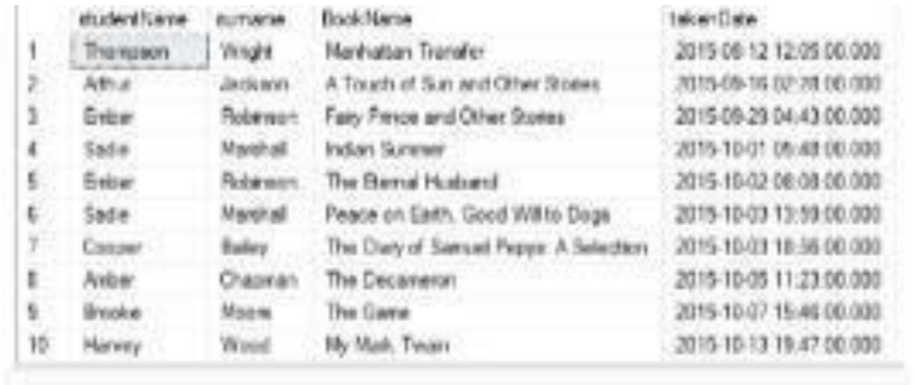
```
Select * from Student
where S_Id = (Select Id from Books where Date_returned > "2022-05-02");
```



```
Select students.name as studentName,students.surname,books.name as BookName
from students
join borrows on students.studentId = borrows.studentId
join books on books.bookId = borrows.bookId
where class='11B'
```

Output:

```
S_Name   S_ID    Branch  Section Contact_no
Rishik Sahu      1       AI      B       9988776611
```



	studentName	surname	BookName	takenDate
1	Thompson	Wright	Manhattan Transfer	2015-08-12 12:05:00.000
2	Arthur	Jackson	A Touch of Sun and Other Stories	2015-09-16 02:28:00.000
3	Ember	Robinson	Fairy Prince and Other Stories	2015-09-29 04:43:00.000
4	Sadie	Marshall	Indian Summer	2015-10-01 06:48:00.000
5	Ember	Robinson	The Eternal Husband	2015-10-02 06:08:00.000
6	Sadie	Marshall	Peace on Earth, Good Will to Dogs	2015-10-03 13:59:00.000
7	Cooper	Bailey	The Diary of Samuel Pepys: A Selection	2015-10-03 18:56:00.000
8	Amber	Chapman	The Decameron	2015-10-05 11:23:00.000
9	Brooke	Moore	The Game	2015-10-07 15:46:00.000
10	Harvey	Wood	My Mark Twain	2015-10-13 19:47:00.000

Sub Query

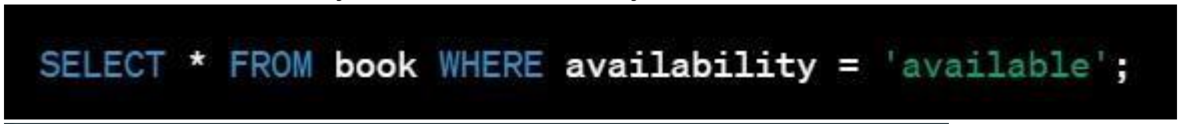
A subquery, also known as a nested query or inner query, is a query that is nested within another query. It allows you to combine multiple queries into a single query by embedding one query within another. The result of a subquery is used by the outer query to perform further operations or filtering.

Subqueries are typically enclosed in parentheses and placed within the WHERE clause, HAVING clause, or the FROM clause of the outer query. The subquery is evaluated first, and its result is then used by the outer query.

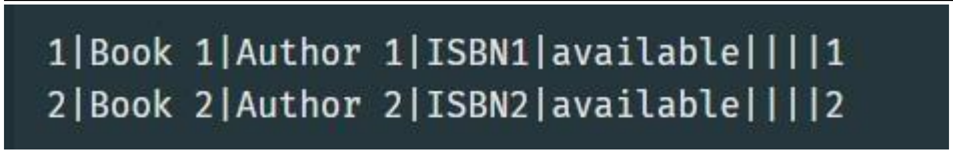
Subqueries offer great flexibility and power in SQL as they allow you to break down complex problems into smaller, more manageable parts. They can be used to retrieve data, filter records, perform calculations, or even update and delete data based on certain conditions. However, it's important to note that excessive or poorly optimized use of subqueries can impact performance, so it's recommended to use them judiciously and consider alternative approaches like joins or temporary tables where appropriate.

Code and Output:

1. Retrieve all books currently available in the library:

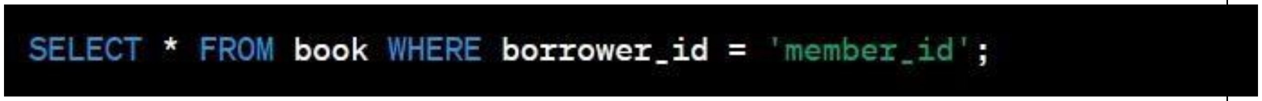


```
SELECT * FROM book WHERE availability = 'available';
```



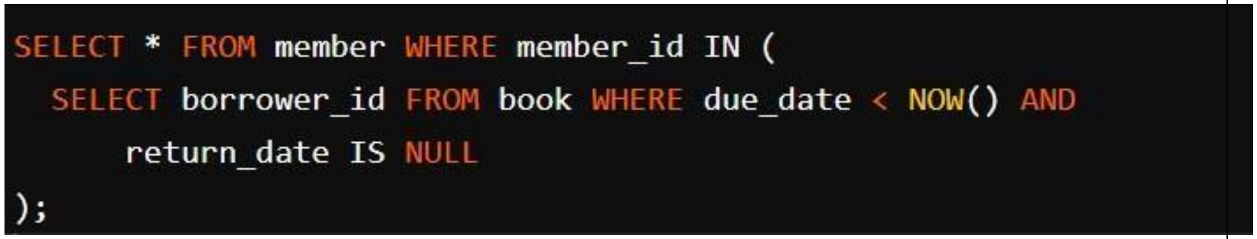
```
1|Book 1|Author 1|ISBN1|available|||1
2|Book 2|Author 2|ISBN2|available|||2
```

2. Retrieve all books borrowed by a specific member:



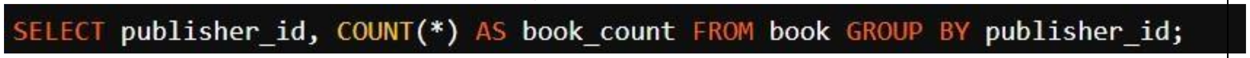
```sql
SELECT * FROM book WHERE borrower_id = 'member_id';
```

3. Retrieve all members who have overdue books:



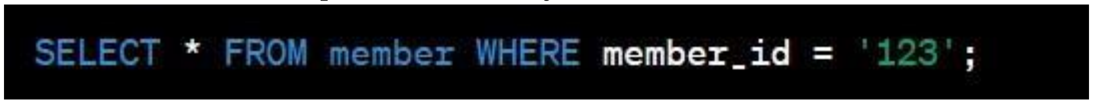
```sql
SELECT * FROM member WHERE member_id IN (
  SELECT borrower_id FROM book WHERE due_date < NOW() AND
     return_date IS NULL
);
```

4. Retrieve the number of books published by each publisher:



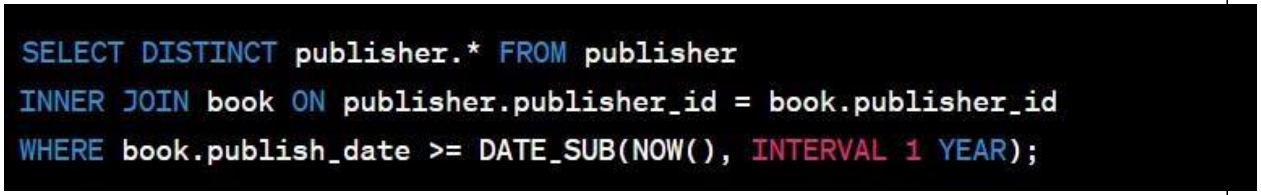
```sql
SELECT publisher_id, COUNT(*) AS book_count FROM book GROUP BY publisher_id;
```

5. Retrieve the details of a specific member by their ID:



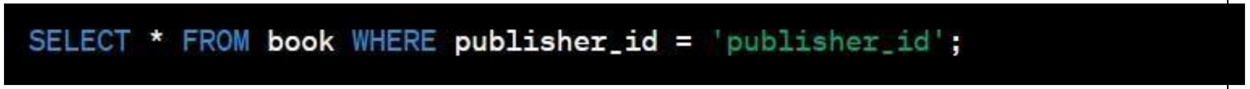
```sql
SELECT * FROM member WHERE member_id = '123';
```

6. Retrieve all publishers who have published books in the last year:



```sql
SELECT DISTINCT publisher.* FROM publisher
INNER JOIN book ON publisher.publisher_id = book.publisher_id
WHERE book.publish_date >= DATE_SUB(NOW(), INTERVAL 1 YEAR);
```

7. Retrieve all books published by a specific publisher:



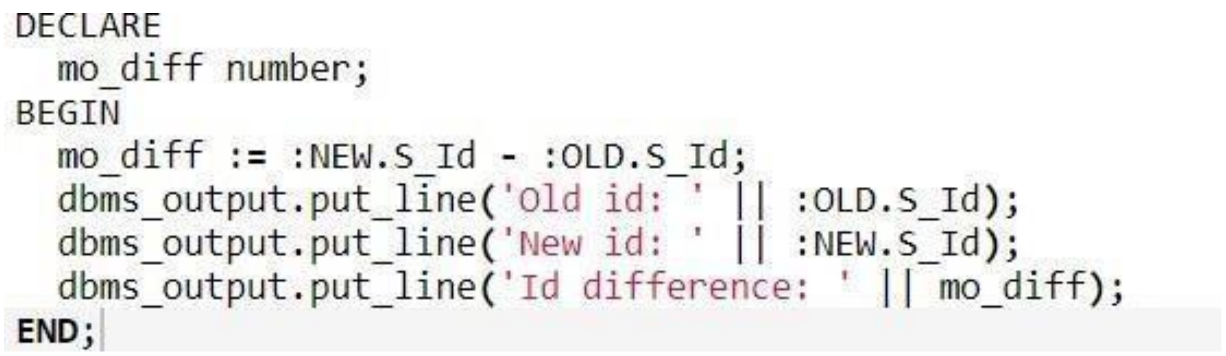
```sql
SELECT * FROM book WHERE publisher_id = 'publisher_id';
```

TRIGGERS

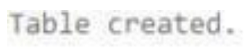
Description:

Triggers are the SQL statements that are automatically executed when there is any change in the database. The triggers are executed in response to certain events (INSERT, UPDATE or DELETE) in a particular table. These triggers help in maintaining the integrity of the data by changing the data of the database in a systematic fashion.

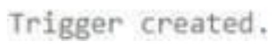
Code and Output:



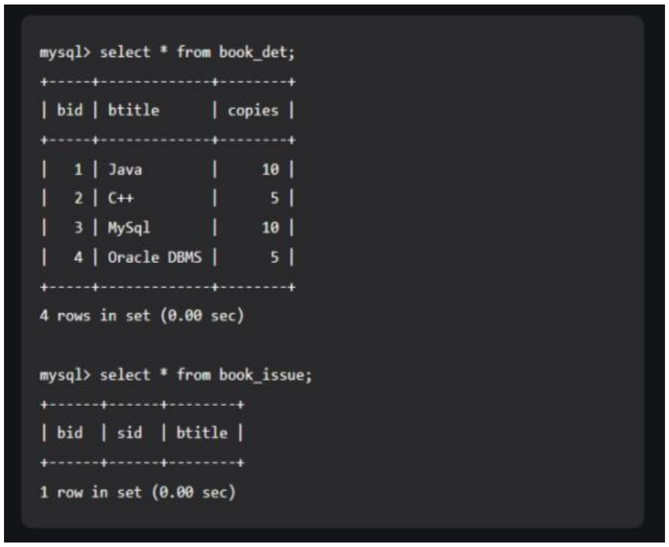
```
DECLARE
  mo_diff number;
BEGIN
  mo_diff := :NEW.S_Id - :OLD.S_Id;
  dbms_output.put_line('Old id: ' || :OLD.S_Id);
  dbms_output.put_line('New id: ' || :NEW.S_Id);
  dbms_output.put_line('Id difference: ' || mo_diff);
END;
```



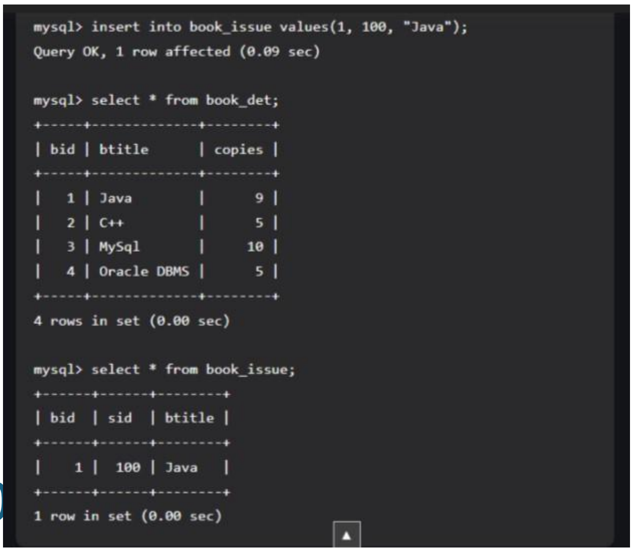
```
Table created.
```



```
Trigger created.
```



```
mysql> select * from book_det;
+-----+-------------+--------+
| bid | btitle      | copies |
+-----+-------------+--------+
|   1 | Java        |     10 |
|   2 | C++         |      5 |
|   3 | MySql       |     10 |
|   4 | Oracle DBMS |      5 |
+-----+-------------+--------+
4 rows in set (0.00 sec)

mysql> select * from book_issue;
+------+------+--------+
| bid  | sid  | btitle |
+------+------+--------+
1 row in set (0.00 sec)
```



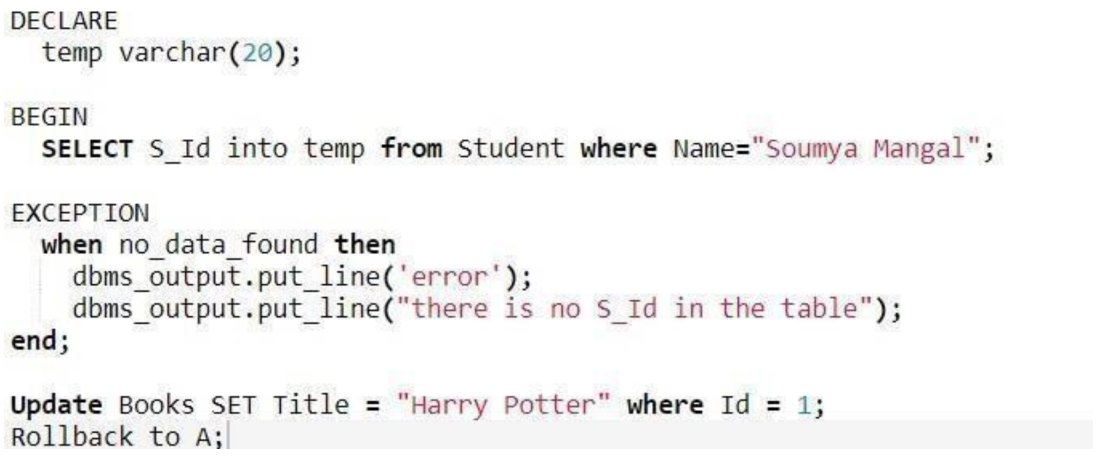
```
mysql> insert into book_issue values(1, 100, "Java");
Query OK, 1 row affected (0.09 sec)

mysql> select * from book_det;
+-----+-------------+--------+
| bid | btitle      | copies |
+-----+-------------+--------+
|   1 | Java        |      9 |
|   2 | C++         |      5 |
|   3 | MySql       |     10 |
|   4 | Oracle DBMS |      5 |
+-----+-------------+--------+
4 rows in set (0.00 sec)

mysql> select * from book_issue;
+------+------+--------+
| bid  | sid  | btitle |
+------+------+--------+
|    1 |  100 | Java   |
+------+------+--------+
1 row in set (0.00 sec)
```

EXCEPTION HANDLING

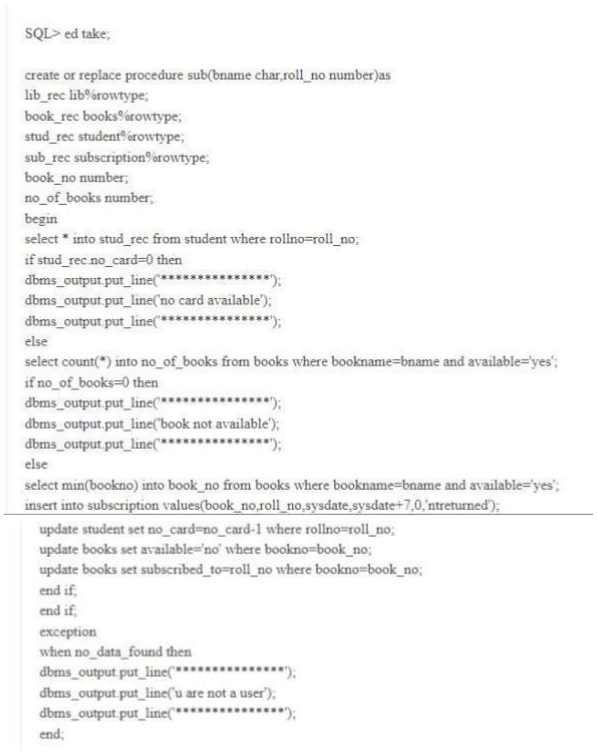
Description:

An exception is an error which disrupts the normal flow of program instructions. PL/SQL provides us the exception block which raises the exception thus helping the programmer to find out the fault and resolve it.

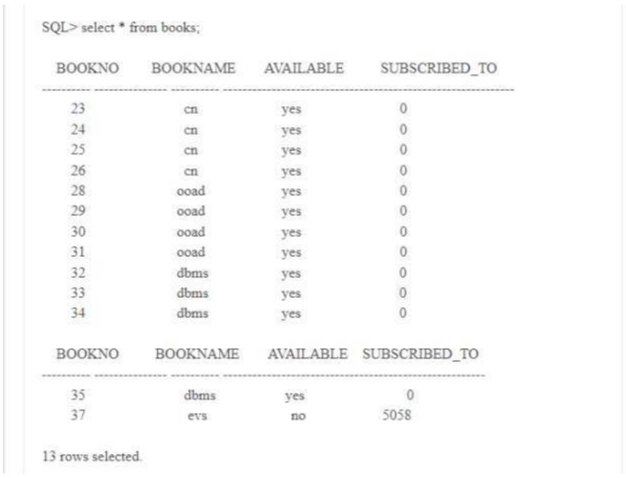
Code and Output:



```
DECLARE
  temp varchar(20);

BEGIN
  SELECT S_Id into temp from Student where Name="Soumya Mangal";

EXCEPTION
  when no_data_found then
    dbms_output.put_line('error');
    dbms_output.put_line("there is no S_Id in the table");
end;

Update Books SET Title = "Harry Potter" where Id = 1;
Rollback to A;
```



```
SQL> ed take;

create or replace procedure sub(bname char,roll_no number)as
lib_rec lib%rowtype;
book_rec books%rowtype;
stud_rec student%rowtype;
sub_rec subscription%rowtype;
book_no number;
no_of_books number;
begin
select * into stud_rec from student where rollno=roll_no;
if stud_rec.no_card=0 then
dbms_output.put_line('***************');
dbms_output.put_line('no card available');
dbms_output.put_line('***************');
else
select count(*) into no_of_books from books where bookname=bname and available='yes';
if no_of_books=0 then
dbms_output.put_line('***************');
dbms_output.put_line('book not available');
dbms_output.put_line('***************');
else
select min(bookno) into book_no from books where bookname=bname and available='yes';
insert into subscription values(book_no,roll_no,sysdate,sysdate+7,0,'ntreturned');
update student set no_card=no_card-1 where rollno=roll_no;
update books set available='no' where bookno=book_no;
update books set subscribed_to=roll_no where bookno=book_no;
end if;
end if;
exception
when no_data_found then
dbms_output.put_line('***************');
dbms_output.put_line('u are not a user');
dbms_output.put_line('***************');
end;
```



```
SQL> select * from books;

  BOOKNO    BOOKNAME    AVAILABLE    SUBSCRIBED_TO
---------- --------------- ---------- ----------------------------------------------------------
    23          cn          yes           0
    24          cn          yes           0
    25          cn          yes           0
    26          cn          yes           0
    28          ooad        yes           0
    29          ooad        yes           0
    30          ooad        yes           0
    31          ooad        yes           0
    32          dbms        yes           0
    33          dbms        yes           0
    34          dbms        yes           0

  BOOKNO    BOOKNAME    AVAILABLE   SUBSCRIBED_TO
---------- --------------- ---------- -----------------------------------------------------
    35          dbms        yes           0
    37          evs          no        5058

13 rows selected.
```

CONCLUSION:

Library management system using MySQL robust solution to streamline library operations. Through efficient database management and automation of tasks, such as book and borrower management, loan tracking, and report generation, the system enhances overall efficiency and user experience.

By leveraging SQL commands, including DDL, DML, TCL, and inbuilt functions, alongside concepts like joins, nested queries, triggers, and exception handling, libraries can transition seamlessly into the digital era while improving accessibility and service quality for patrons.