Student Name: Bhavya
Branch: CSE(AI & ML)
Semester: 4
Subject Name: Database Management System

UID: 24BAI70791
Section/Group: 24AIT_KRG-1/G2
Date of Performance: 13/01/2026
Subject Code: 24CSH-298

Experiment 1

Aim

To design and implement a **Library Management System database** using PostgreSQL that stores information about books, members, and issue records, and to apply **role-based access control** using SQL commands.

Software Requirements

- PostgreSQL/pgAdmin
- MSWord

Objective

- To create tables for **Books, Members, and Issue Records**.
- To perform **INSERT, SELECT, UPDATE, and DELETE** operations.
- To apply **Primary Key, Foreign Key, and CHECK constraints**.
- To create a **librarian role** and assign permissions.
- To understand **database security using GRANT and REVOKE** commands.

Practical / Experiment Steps

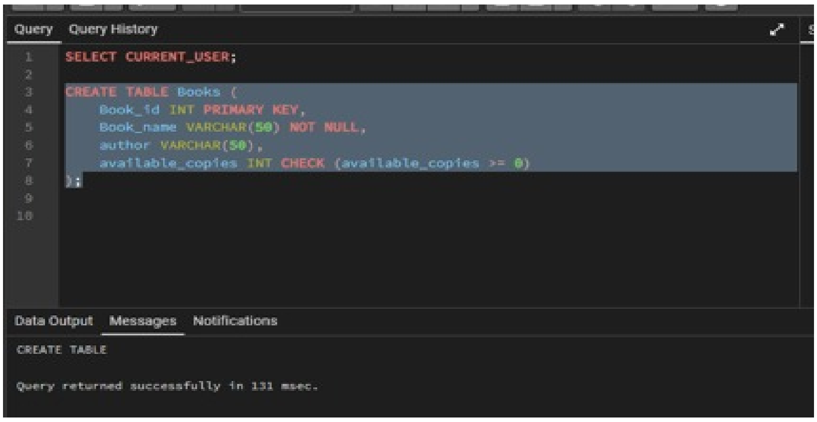
1. Start the system and log in to PostgreSQL.
2. Create the required tables for the library database.

3. Insert records into the tables.
4. Perform SELECT, UPDATE, and DELETE operations.
5. Create a role named librarian2.
6. Grant SELECT, INSERT, and DELETE permissions to the librarian2.
7. Revoke DELETE permission on Books table.
8. Login as librarian2 and test access permissions.
9. Verify output after execution.
10. Save the work and take screenshots.

Practical / Experiment Steps

(A) Database Creation and Operations

```
SELECT CURRENT_USER;


CREATE TABLE Books (
    Book_id INT PRIMARY KEY,
    Book_name VARCHAR(50) NOT NULL,
    author VARCHAR(50),
    available_copies INT CHECK (available_copies >= 0)
);
```



```
SELECT CURRENT_USER;

CREATE TABLE Books (
    Book_id INT PRIMARY KEY,
    Book_name VARCHAR(50) NOT NULL,
    author VARCHAR(50),
    available_copies INT CHECK (available_copies >= 0)
);
```

```
CREATE TABLE

Query returned successfully in 131 msec.
```

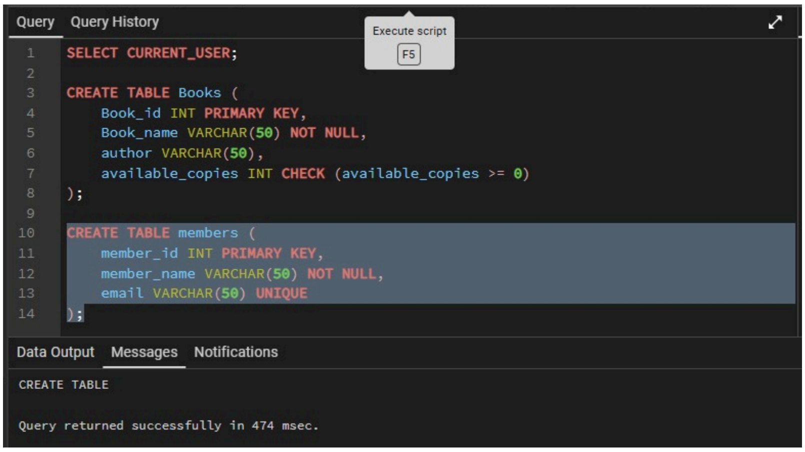
CREATE TABLE members (

member_id INT PRIMARY KEY,

member_name VARCHAR(50) NOT NULL,

email VARCHAR(50) UNIQUE

);



```sql
SELECT CURRENT_USER;

CREATE TABLE Books (
    Book_id INT PRIMARY KEY,
    Book_name VARCHAR(50) NOT NULL,
    author VARCHAR(50),
    available_copies INT CHECK (available_copies >= 0)
);

CREATE TABLE members (
    member_id INT PRIMARY KEY,
    member_name VARCHAR(50) NOT NULL,
    email VARCHAR(50) UNIQUE
);
```

```
CREATE TABLE

Query returned successfully in 474 msec.
```

CREATE TABLE issue_records (

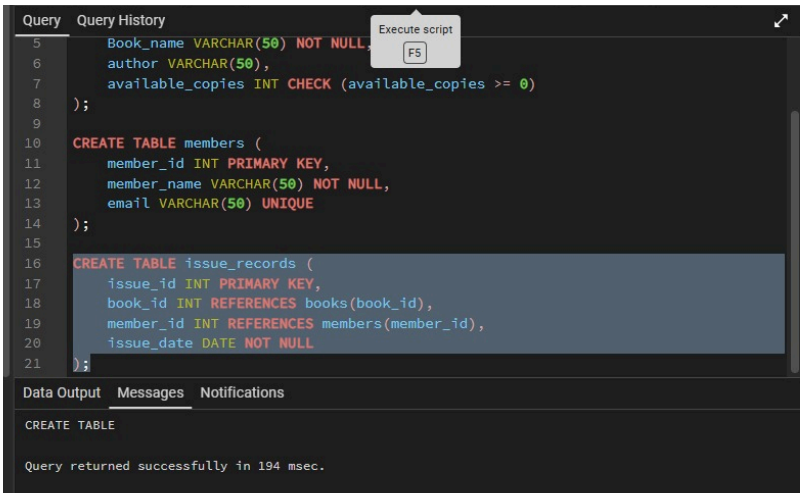
issue_id INT PRIMARY KEY,

book_id INT REFERENCES books(book_id),

member_id INT REFERENCES members(member_id),

issue_date DATE NOT NULL

);



```sql
    Book_name VARCHAR(50) NOT NULL,
    author VARCHAR(50),
    available_copies INT CHECK (available_copies >= 0)
);

CREATE TABLE members (
    member_id INT PRIMARY KEY,
    member_name VARCHAR(50) NOT NULL,
    email VARCHAR(50) UNIQUE
);

CREATE TABLE issue_records (
    issue_id INT PRIMARY KEY,
    book_id INT REFERENCES books(book_id),
    member_id INT REFERENCES members(member_id),
    issue_date DATE NOT NULL
);
```

```
CREATE TABLE

Query returned successfully in 194 msec.
```

INSERT INTO books VALUES (1, 'DBMS', 'Korth', 5);

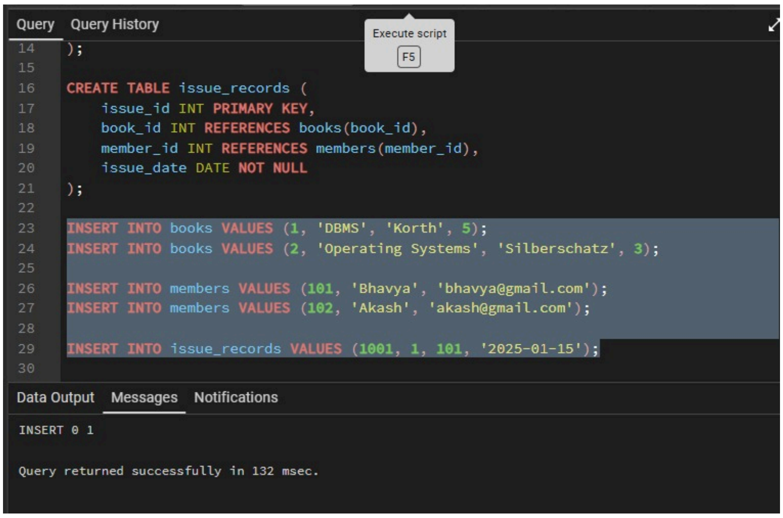
INSERT INTO books VALUES (2, 'Operating Systems', 'Silberschatz', 3);

INSERT INTO members VALUES (101, 'Bhavya', 'bhavya@gmail.com');

INSERT INTO members VALUES (102, 'Akash', 'akash@gmail.com');

INSERT INTO issue_records VALUES (1001, 1, 101, '2025-01-15');

INSERT INTO issue_records VALUES (1002, 2, 102, '2025-06-15');



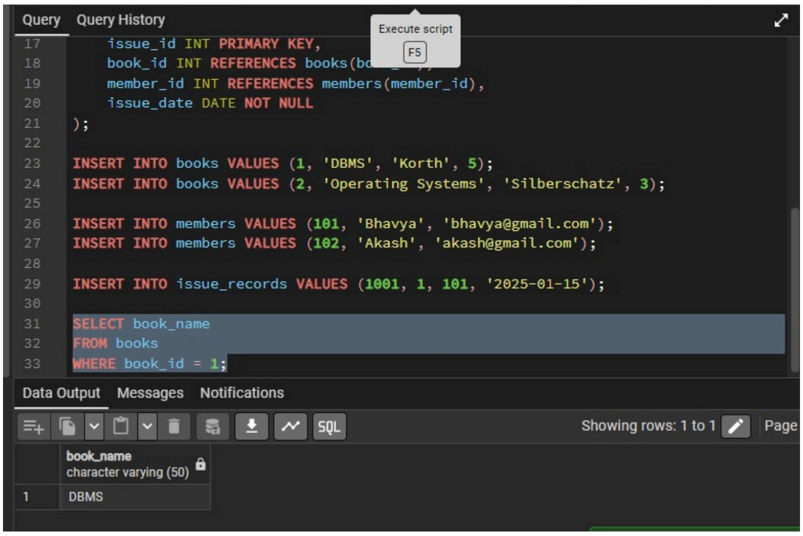
```sql
);

CREATE TABLE issue_records (
    issue_id INT PRIMARY KEY,
    book_id INT REFERENCES books(book_id),
    member_id INT REFERENCES members(member_id),
    issue_date DATE NOT NULL
);

INSERT INTO books VALUES (1, 'DBMS', 'Korth', 5);
INSERT INTO books VALUES (2, 'Operating Systems', 'Silberschatz', 3);

INSERT INTO members VALUES (101, 'Bhavya', 'bhavya@gmail.com');
INSERT INTO members VALUES (102, 'Akash', 'akash@gmail.com');

INSERT INTO issue_records VALUES (1001, 1, 101, '2025-01-15');
```

```
INSERT 0 1

Query returned successfully in 132 msec.
```

SELECT book_name

FROM books

WHERE book_id = 1;



```sql
    issue_id INT PRIMARY KEY,
    book_id INT REFERENCES books(bo
    member_id INT REFERENCES members(member_id),
    issue_date DATE NOT NULL
);

INSERT INTO books VALUES (1, 'DBMS', 'Korth', 5);
INSERT INTO books VALUES (2, 'Operating Systems', 'Silberschatz', 3);

INSERT INTO members VALUES (101, 'Bhavya', 'bhavya@gmail.com');
INSERT INTO members VALUES (102, 'Akash', 'akash@gmail.com');

INSERT INTO issue_records VALUES (1001, 1, 101, '2025-01-15');

SELECT book_name
FROM books
WHERE book_id = 1;
```

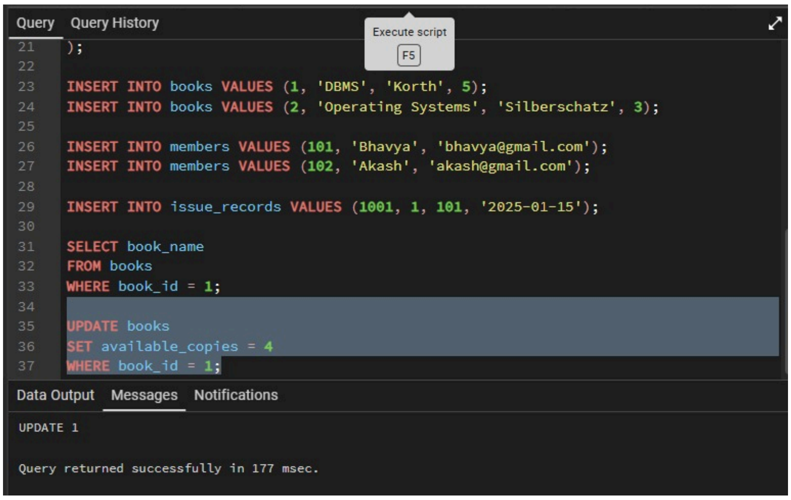
Showing rows: 1 to 1

	book_name character varying (50)
1	DBMS

UPDATE books

SET available_copies = 4

WHERE book_id = 1;



```
);

INSERT INTO books VALUES (1, 'DBMS', 'Korth', 5);
INSERT INTO books VALUES (2, 'Operating Systems', 'Silberschatz', 3);

INSERT INTO members VALUES (101, 'Bhavya', 'bhavya@gmail.com');
INSERT INTO members VALUES (102, 'Akash', 'akash@gmail.com');

INSERT INTO issue_records VALUES (1001, 1, 101, '2025-01-15');

SELECT book_name
FROM books
WHERE book_id = 1;

UPDATE books
SET available_copies = 4
WHERE book_id = 1;
```

```
UPDATE 1

Query returned successfully in 177 msec.
```

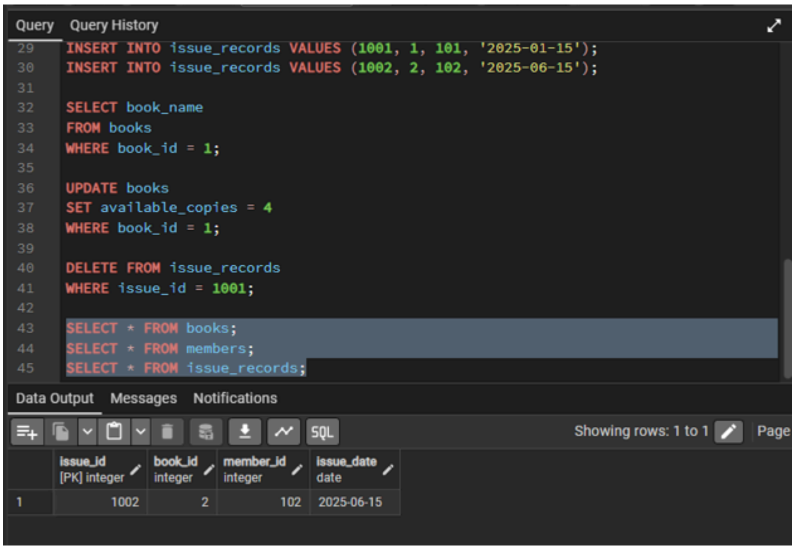
DELETE FROM issue_records

WHERE issue_id = 1001;

SELECT * FROM books;

SELECT * FROM members;

SELECT * FROM issue_records;



```
INSERT INTO issue_records VALUES (1001, 1, 101, '2025-01-15');
INSERT INTO issue_records VALUES (1002, 2, 102, '2025-06-15');

SELECT book_name
FROM books
WHERE book_id = 1;

UPDATE books
SET available_copies = 4
WHERE book_id = 1;

DELETE FROM issue_records
WHERE issue_id = 1001;

SELECT * FROM books;
SELECT * FROM members;
SELECT * FROM issue_records;
```

	issue_id [PK] integer	book_id integer	member_id integer	issue_date date
1	1002	2	102	2025-06-15

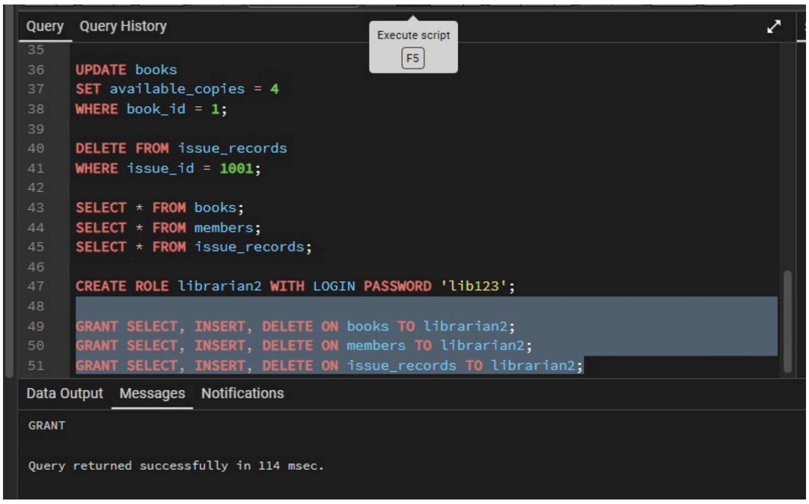
(B) Role Creation and Privileges

CREATE ROLE librarian2 WITH LOGIN PASSWORD 'lib123';

GRANT SELECT, INSERT, DELETE ON books TO librarian2;

GRANT SELECT, INSERT, DELETE ON members TO librarian2;

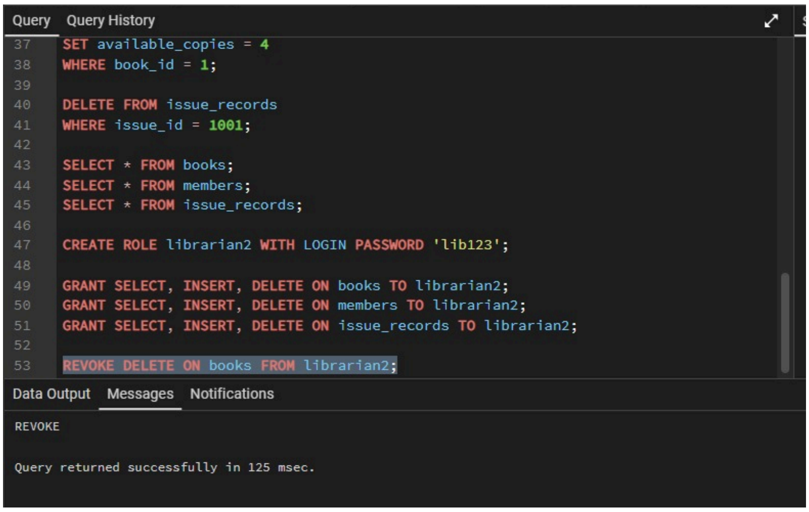
GRANT SELECT, INSERT, DELETE ON issue_records TO librarian2;



```

UPDATE books
SET available_copies = 4
WHERE book_id = 1;

DELETE FROM issue_records
WHERE issue_id = 1001;

SELECT * FROM books;
SELECT * FROM members;
SELECT * FROM issue_records;

CREATE ROLE librarian2 WITH LOGIN PASSWORD 'lib123';

GRANT SELECT, INSERT, DELETE ON books TO librarian2;
GRANT SELECT, INSERT, DELETE ON members TO librarian2;
GRANT SELECT, INSERT, DELETE ON issue_records TO librarian2;
```

```
GRANT

Query returned successfully in 114 msec.
```

REVOKE DELETE ON books FROM librarian2;



```
SET available_copies = 4
WHERE book_id = 1;

DELETE FROM issue_records
WHERE issue_id = 1001;

SELECT * FROM books;
SELECT * FROM members;
SELECT * FROM issue_records;

CREATE ROLE librarian2 WITH LOGIN PASSWORD 'lib123';

GRANT SELECT, INSERT, DELETE ON books TO librarian2;
GRANT SELECT, INSERT, DELETE ON members TO librarian2;
GRANT SELECT, INSERT, DELETE ON issue_records TO librarian2;

REVOKE DELETE ON books FROM librarian2;
```

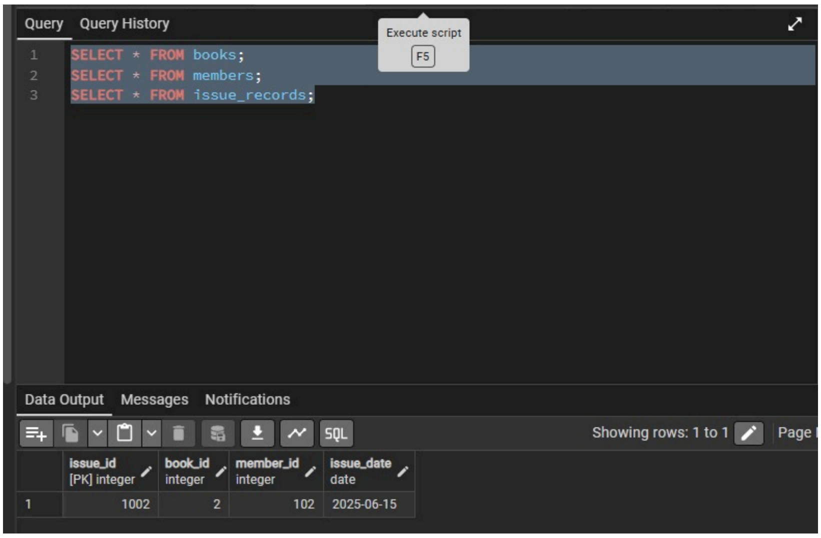
```
REVOKE

Query returned successfully in 125 msec.
```

(C) Librarian Access Operations

SELECT * FROM books;

SELECT * FROM members;

SELECT * FROM issue_records;

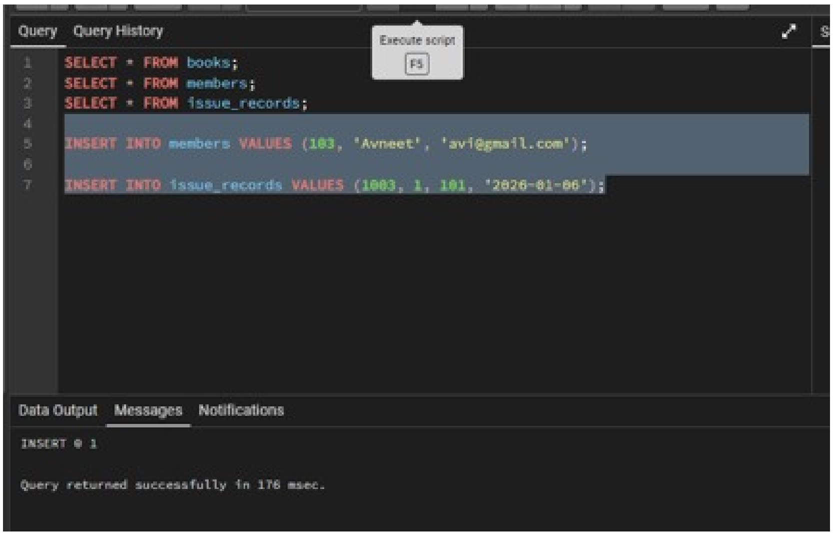


```sql
SELECT * FROM books;
SELECT * FROM members;
SELECT * FROM issue_records;
```

	issue_id [PK] integer	book_id integer	member_id integer	issue_date date
1	1002	2	102	2025-06-15

INSERT INTO members VALUES (103, 'Avneet', 'avi@gmail.com');

INSERT INTO issue_records VALUES (1003, 1, 101, '2026-01-06');

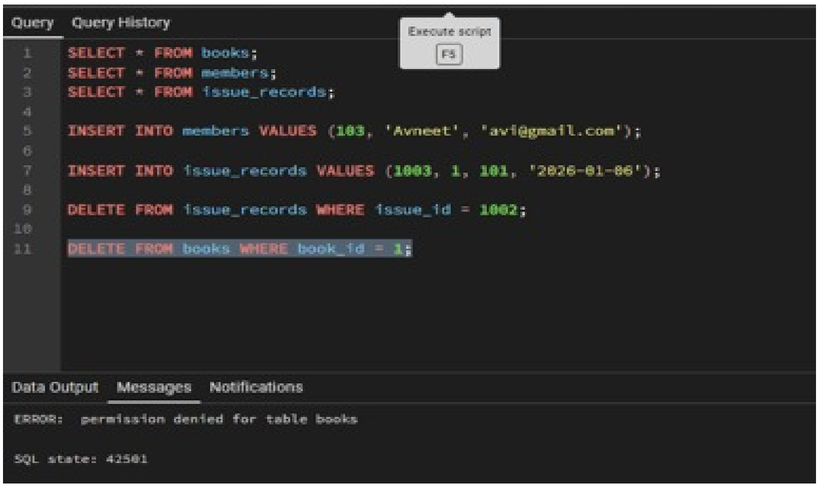


```sql
SELECT * FROM books;
SELECT * FROM members;
SELECT * FROM issue_records;

INSERT INTO members VALUES (103, 'Avneet', 'avi@gmail.com');

INSERT INTO issue_records VALUES (1003, 1, 101, '2026-01-06');
```

```
INSERT 0 1

Query returned successfully in 176 msec.
```

DELETE FROM issue_records WHERE issue_id = 1002;

DELETE FROM books WHERE book_id = 1;



```
SELECT * FROM books;
SELECT * FROM members;
SELECT * FROM issue_records;

INSERT INTO members VALUES (103, 'Avneet', 'avi@gmail.com');

INSERT INTO issue_records VALUES (1003, 1, 101, '2026-01-06');

DELETE FROM issue_records WHERE issue_id = 1002;

DELETE FROM books WHERE book_id = 1;
```

```
ERROR:  permission denied for table books

SQL state: 42501
```

Input/Output Details

Input:

- Book ID, Book Name, Author, Available Copies
- Member ID, Member Name, Email
- Issue ID, Book ID, Member ID, Issue Date

Output:

- Tables created successfully
- Records inserted into tables
- Book details retrieved using SELECT
- Available copies updated
- Issue record deleted
- Role created and permissions assigned

Learning Outcome

- Learned to design relational tables using **primary and foreign keys**.

- Understood the use of **constraints** to maintain data integrity.
- Gained knowledge of **CRUD operations** in SQL.
- Learned how to create and manage **user roles**.
- Understood **database security using GRANT and REVOKE** commands.
- Acquired practical exposure to **Library Management System database design**.