



**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.

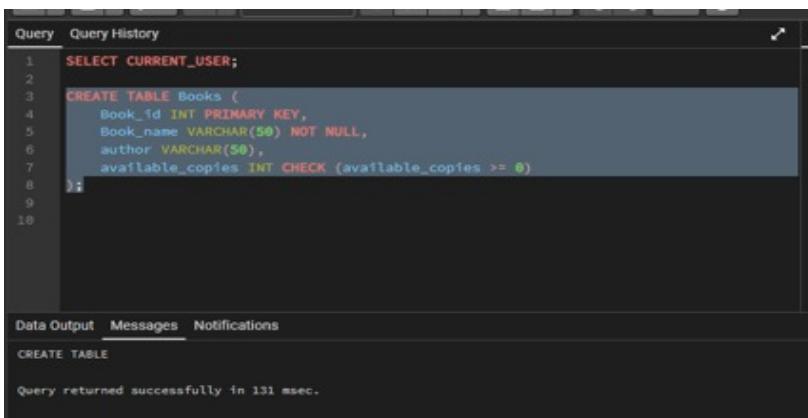
3. Insert records into the tables.
  4. Perform SELECT, UPDATE, and DELETE operations.
  5. Create a role named librarian2.
  6. Grant SELECT, INSERT, and DELETE permissions to the librarian2.
  7. Revoke DELETE permission on Books table.
  8. Login as librarian2 and test access permissions.
  9. Verify 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)
);
```



The screenshot shows the MySQL Workbench interface with a query editor window. The code entered is:

```
1 SELECT CURRENT_USER;
2
3 CREATE TABLE Books (
4     Book_id INT PRIMARY KEY,
5     Book_name VARCHAR(50) NOT NULL,
6     author VARCHAR(50),
7     available_copies INT CHECK (available_copies >= 0)
8 );
9
10
```

Below the editor, the "Data Output" tab is selected, showing the result:

```
CREATE TABLE
```

At the bottom, a status message indicates:

```
Query returned successfully in 131 msec.
```



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

```
Query  Query History Execute script F5
1  SELECT CURRENT_USER;
2
3  CREATE TABLE Books (
4      Book_id INT PRIMARY KEY,
5      Book_name VARCHAR(50) NOT NULL,
6      author VARCHAR(50),
7      available_copies INT CHECK (available_copies >= 0)
8  );
9
10 CREATE TABLE members (
11     member_id INT PRIMARY KEY,
12     member_name VARCHAR(50) NOT NULL,
13     email VARCHAR(50) UNIQUE
14 );
```

Data Output Messages Notifications

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
);
```

```
Query  Query History Execute script F5
5  Book_name VARCHAR(50) NOT NULL,
6  author VARCHAR(50),
7  available_copies INT CHECK (available_copies >= 0)
8  );
9
10 CREATE TABLE members (
11     member_id INT PRIMARY KEY,
12     member_name VARCHAR(50) NOT NULL,
13     email VARCHAR(50) UNIQUE
14 );
15
16 CREATE TABLE issue_records (
17     issue_id INT PRIMARY KEY,
18     book_id INT REFERENCES books(book_id),
19     member_id INT REFERENCES members(member_id),
20     issue_date DATE NOT NULL
21 );
```

Data Output Messages Notifications

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');

```

Query  Query History
14 );
15
16 CREATE TABLE issue_records (
17   issue_id INT PRIMARY KEY,
18   book_id INT REFERENCES books(book_id),
19   member_id INT REFERENCES members(member_id),
20   issue_date DATE NOT NULL
21 );
22
23 INSERT INTO books VALUES (1, 'DBMS', 'Korth', 5);
24 INSERT INTO books VALUES (2, 'Operating Systems', 'Silberschatz', 3);
25
26 INSERT INTO members VALUES (101, 'Bhavya', 'bhavya@gmail.com');
27 INSERT INTO members VALUES (102, 'Akash', 'akash@gmail.com');
28
29 INSERT INTO issue_records VALUES (1001, 1, 101, '2025-01-15');
30

Data Output  Messages  Notifications
INSERT 0 1

Query returned successfully in 132 msec.

```

SELECT book\_name

FROM books

WHERE book\_id = 1;

```

Query  Query History
17   issue_id INT PRIMARY KEY,
18   book_id INT REFERENCES books(book_id),
19   member_id INT REFERENCES members(member_id),
20   issue_date DATE NOT NULL
21 );
22
23 INSERT INTO books VALUES (1, 'DBMS', 'Korth', 5);
24 INSERT INTO books VALUES (2, 'Operating Systems', 'Silberschatz', 3);
25
26 INSERT INTO members VALUES (101, 'Bhavya', 'bhavya@gmail.com');
27 INSERT INTO members VALUES (102, 'Akash', 'akash@gmail.com');
28
29 INSERT INTO issue_records VALUES (1001, 1, 101, '2025-01-15');
30
31 SELECT book_name
32 FROM books
33 WHERE book_id = 1;

Data Output  Messages  Notifications
Showing rows: 1 to 1 | Page
SQL

book_name
character varying (50) 🔒
1 DBMS

```

## UPDATE books

SET available\_copies = 4

WHERE book\_id = 1;

```

Query  Query History
21 );
22
23 INSERT INTO books VALUES (1, 'DBMS', 'Korth', 5);
24 INSERT INTO books VALUES (2, 'Operating Systems', 'Silberschatz', 3);
25
26 INSERT INTO members VALUES (101, 'Bhavya', 'bhavya@gmail.com');
27 INSERT INTO members VALUES (102, 'Akash', 'akash@gmail.com');
28
29 INSERT INTO issue_records VALUES (1001, 1, 101, '2025-01-15');
30
31 SELECT book_name
32 FROM books
33 WHERE book_id = 1;
34
35 UPDATE books
36 SET available_copies = 4
37 WHERE book_id = 1;

```

Data Output    Messages    Notifications

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;

```

Query  Query History
29 INSERT INTO issue_records VALUES (1001, 1, 101, '2025-01-15');
30 INSERT INTO issue_records VALUES (1002, 2, 102, '2025-06-15');
31
32 SELECT book_name
33 FROM books
34 WHERE book_id = 1;
35
36 UPDATE books
37 SET available_copies = 4
38 WHERE book_id = 1;
39
40 DELETE FROM issue_records
41 WHERE issue_id = 1001;
42
43 SELECT * FROM books;
44 SELECT * FROM members;
45 SELECT * FROM issue_records;

```

Data Output    Messages    Notifications

issue_id	book_id	member_id	issue_date	
1	1002	2	102	2025-06-15

Showing rows: 1 to 1    Page



NAAC  
GRADE A+

## (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;
```

The screenshot shows a database query editor with the following content:

```
Query  Query History
35
36 UPDATE books
37 SET available_copies = 4
38 WHERE book_id = 1;
39
40 DELETE FROM issue_records
41 WHERE issue_id = 1001;
42
43 SELECT * FROM books;
44 SELECT * FROM members;
45 SELECT * FROM issue_records;
46
47 CREATE ROLE librarian2 WITH LOGIN PASSWORD 'lib123';
48
49 GRANT SELECT, INSERT, DELETE ON books TO librarian2;
50 GRANT SELECT, INSERT, DELETE ON members TO librarian2;
51 GRANT SELECT, INSERT, DELETE ON issue_records TO librarian2;

Data Output  Messages  Notifications
GRANT

Query returned successfully in 114 msec.
```

```
REVOKE DELETE ON books FROM librarian2;
```

The screenshot shows a database query editor with the following content:

```
Query  Query History
37
38 SET available_copies = 4
39 WHERE book_id = 1;
40
41 DELETE FROM issue_records
42 WHERE issue_id = 1001;
43
44 SELECT * FROM books;
45 SELECT * FROM members;
46 SELECT * FROM issue_records;
47
48 CREATE ROLE librarian2 WITH LOGIN PASSWORD 'lib123';
49
50 GRANT SELECT, INSERT, DELETE ON books TO librarian2;
51 GRANT SELECT, INSERT, DELETE ON members TO librarian2;
52 GRANT SELECT, INSERT, DELETE ON issue_records TO librarian2;
53 REVOKE DELETE ON books FROM librarian2;

Data Output  Messages  Notifications
REVOKE

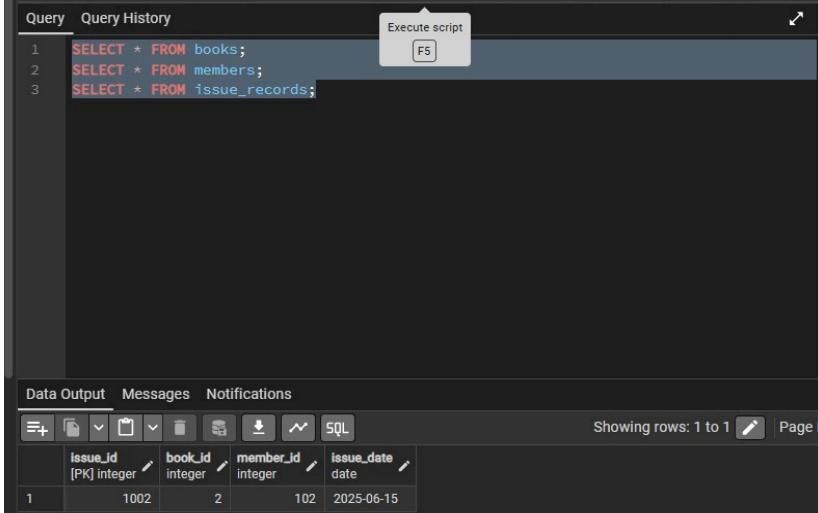
Query returned successfully in 125 msec.
```

## (C) Librarian Access Operations

SELECT \* FROM books;

SELECT \* FROM members;

SELECT \* FROM issue\_records;



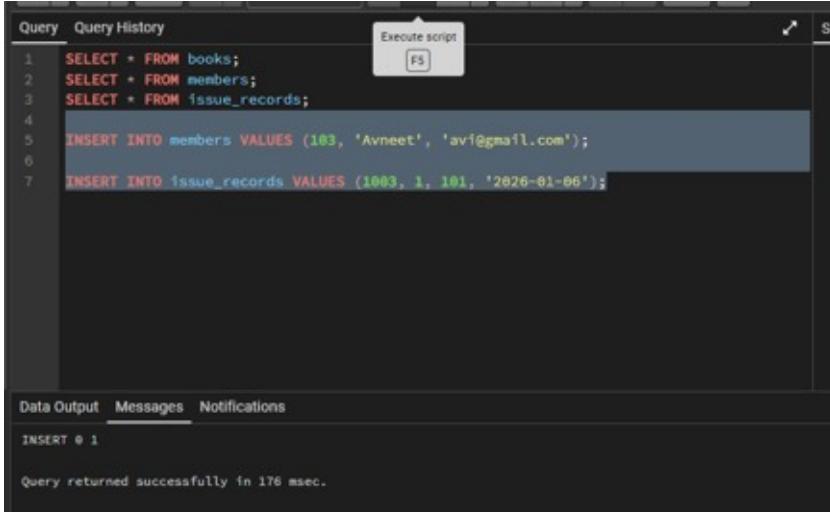
```
Query  Query History
1  SELECT * FROM books;
2  SELECT * FROM members;
3  SELECT * FROM issue_records;

Data Output  Messages  Notifications
SQL
issue_id | book_id | member_id | issue_date |
1        | 1002    | 2          | 2025-06-15 |

Showing rows: 1 to 1
```

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

INSERT INTO issue\_records VALUES (1003, 1, 101, '2026-01-06');



```
Query  Query History
1  SELECT * FROM books;
2  SELECT * FROM members;
3  SELECT * FROM issue_records;
4
5  INSERT INTO members VALUES (103, 'Avneet', 'avi@gmail.com');
6
7  INSERT INTO issue_records VALUES (1003, 1, 101, '2026-01-06');

Data Output  Messages  Notifications
SQL
INSERT 0 1

Query returned successfully in 176 msec.
```

DELETE FROM issue\_records WHERE issue\_id = 1002;

DELETE FROM books WHERE book\_id = 1;

Query    Query History   

```

1 SELECT * FROM books;
2 SELECT * FROM members;
3 SELECT * FROM issue_records;
4
5 INSERT INTO members VALUES (103, 'Avneet', 'avi@gmail.com');
6
7 INSERT INTO issue_records VALUES (1003, 1, 101, '2026-01-06');
8
9 DELETE FROM issue_records WHERE issue_id = 1002;
10
11 DELETE FROM books WHERE book_id = 1;

```

Data Output    Messages    Notifications

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**.