Student Name: ARNAV SHARMA **UID: 24BAI70781**
Branch: AIT-CSE (AIML) **Section/Group: 24AIT_KRG2**
Semester: 4
Subject Name: Database Management System **Subject Code: 24CSH-298**

Experiment

Experiment 1.1: Design and implementation of a Library Management System using PostgreSQL with DDL, DML and DCL commands.

Aim

The aim of this experiment is to design and implement a Library Management System database using PostgreSQL. The database is created using proper tables, primary keys, foreign keys and constraints. DML operations are performed and database security is implemented using roles and privileges.

Objective

The objective of this experiment is to gain practical knowledge of DDL, DML and DCL commands in PostgreSQL. It also helps in understanding how to create roles, grant permissions and revoke permissions to secure the database using role based access control.

Practical / Experiment Steps

1. Design the database structure for the Library Management System.
2. Create tables for books, members and issue records using DDL commands.
3. Apply primary keys, foreign keys and constraints to maintain data integrity.
4. Insert sample records into the tables using DML commands.
5. Update and delete records as required.
6. Create a database role named Librarian.
7. Grant required permissions like SELECT, INSERT and DELETE to the Librarian role.
8. Revoke permissions when needed to ensure database security.

Procedure of the Experiment

1. Start the system and log in to the computer.

2. Open pgAdmin and connect to PostgreSQL server.

3. Create a new database for the Library Management System.

4. Create tables such as Books, Members and Issue_Records using CREATE TABLE command.

5. Define primary keys and foreign keys while creating the tables.

6. Insert records into tables using INSERT command.

7. Update existing data using UPDATE command.

8. Delete unwanted records using DELETE command.

9. Create a role named Librarian with password using CREATE ROLE command.

10. Grant SELECT, INSERT and DELETE permissions to the Librarian role.

11. Revoke permissions using REVOKE command when required.

12. Execute all queries and verify the output.

CODE :

1. **ADMIN**

```
-- CURRENT USER
CREATE DATABASE library;
USE library;
SELECT CURRENT_USER();

-- BOOKS TABLE
CREATE TABLE book (
   id INT PRIMARY KEY,
   name VARCHAR(50) NOT NULL,
   authorName VARCHAR(50)
);
```

```
INSERT INTO book (id, name, authorName)
VALUES (101, 'your name', 'arnav'), (102, 'last summer', 'ayush');

ALTER TABLE book ADD COLUMN count INT CHECK(count >= 1);

SELECT * FROM book;

UPDATE book SET count = 3 WHERE id = 101;

-- VISITOR TABLE
CREATE TABLE LIBRARY_VISITOR_USER (
    USER_ID INT PRIMARY KEY,
    USER_NAME VARCHAR(20) NOT NULL,
    AGE INT NOT NULL CHECK(AGE >= 17),
    EMAIL VARCHAR(50) UNIQUE NOT NULL
);

INSERT INTO LIBRARY_VISITOR_USER (USER_ID, USER_NAME, AGE, EMAIL)
VALUES (101, 'ARNAV', 18, 'arnavsharma@gmail.com');

SELECT * FROM LIBRARY_VISITOR_USER;

-- BOOK ISSUE TABLE
-- Note: MySQL uses FOREIGN KEY syntax instead of inline REFERENCES
CREATE TABLE BOOK_ISSUE (
    BOOK_ISSUE_ID INT PRIMARY KEY,
    BOOK_ID INT,
    USER_ID INT,
    BOOK_ISSUE_DATE DATE NOT NULL,
    FOREIGN KEY (BOOK_ID) REFERENCES book(id),
    FOREIGN KEY (USER_ID) REFERENCES LIBRARY_VISITOR_USER(USER_ID)
);
```

```
INSERT INTO BOOK_ISSUE (BOOK_ISSUE_ID, BOOK_ID, USER_ID, BOOK_ISSUE_DATE)
VALUES (34899345, 102, 101, '2026-05-01');

SELECT * FROM BOOK_ISSUE;

-- ROLE & PRIVILEGES
-- Note: MySQL does not support CREATE ROLE with LOGIN/PASSWORD like PostgreSQL.
-- Instead, create a user directly and grant privileges.
CREATE USER 'LIBRARIAN1'@'%' IDENTIFIED BY 'ARNAV-SHARMA';

GRANT SELECT, INSERT, UPDATE, DELETE ON library.book TO 'LIBRARIAN1'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON library.LIBRARY_VISITOR_USER TO
'LIBRARIAN1'@'%';
GRANT SELECT, INSERT, UPDATE, DELETE ON library.BOOK_ISSUE TO 'LIBRARIAN1'@'%';

REVOKE DELETE ON library.book FROM 'LIBRARIAN1'@'%';

FLUSH PRIVILEGES;

-- FINAL SELECTS & INSERTS
SELECT * FROM book;
SELECT * FROM BOOK_ISSUE;
SELECT * FROM LIBRARY_VISITOR_USER;

INSERT INTO book VALUES (110, 'can we meet', 'rishu', 2);
INSERT INTO book VALUES (150, 'hobit', 'hobit', 7);

SELECT * FROM BOOK_ISSUE;
SELECT * FROM LIBRARY_VISITOR_USER;
```

Learning Outcomes:

1. Understood the basics of **relational database design** using tables, keys, and relationships.
2. Learned to apply **primary key and foreign key constraints** to maintain data integrity.

3. Gained hands-on experience with **INSERT, UPDATE, and DELETE** operations safely.
4. Understood how **roles and privileges** control access to database objects.
5. Learned to use **GRANT and REVOKE** for implementing **read-only users**.
6. Practiced **ALTER TABLE and DROP TABLE** for managing database

SCREENSHOTS →

```sql
-- CURRENT USER
CREATE DATABASE library;
USE library;
SELECT CURRENT_USER();

-- BOOKS TABLE
CREATE TABLE book (
    id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    authorName VARCHAR(50)
);

INSERT INTO book (id, name, authorName)
VALUES (101, 'your name', 'arnav'), (102, 'last summer', 'ayush');

ALTER TABLE book ADD COLUMN count INT CHECK(count >= 1);

SELECT * FROM book;

UPDATE book SET count = 3 WHERE id = 101;
```

```
);

INSERT INTO LIBRARY_VISITOR_USER (USER_ID, USER_NAME, AGE, EMAIL)
VALUES (101, 'ARNAV', 18, 'arnavsharma@gmail.com');

SELECT * FROM LIBRARY_VISITOR_USER;

-- BOOK ISSUE TABLE
-- Note: MySQL uses FOREIGN KEY syntax instead of inline REFERENCES
CREATE TABLE BOOK_ISSUE (
    BOOK_ISSUE_ID INT PRIMARY KEY,
    BOOK_ID INT,
    USER_ID INT,
    BOOK_ISSUE_DATE DATE NOT NULL,
    FOREIGN KEY (BOOK_ID) REFERENCES book(id),
    FOREIGN KEY (USER_ID) REFERENCES LIBRARY_VISITOR_USER(USER_ID)
);

INSERT INTO BOOK_ISSUE (BOOK_ISSUE_ID, BOOK_ID, USER_ID, BOOK_ISSUE_DATE)
VALUES (34899345, 102, 101, '2026-05-01');

SELECT * FROM BOOK_ISSUE;

-- ROLE & PRIVILEGES

REVOKE DELETE ON library.book FROM 'LIBRARIAN1'@'%';

FLUSH PRIVILEGES;

-- FINAL SELECTS & INSERTS
SELECT * FROM book;
SELECT * FROM BOOK_ISSUE;
SELECT * FROM LIBRARY_VISITOR_USER;

INSERT INTO book VALUES (110, 'can we meet', 'rishu', 2);
INSERT INTO book VALUES (150, 'hobit', 'hobit', 7);

SELECT * FROM BOOK_ISSUE;
SELECT * FROM LIBRARY_VISITOR_USER;
```

Result Grid | Filter Rows: | Edit: | Export/Impor

id	name	authorName	count
101	your name	arnav	3
102	last summer	ayush	NULL
110	can we meet	rishu	2
NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Expo

BOOK_ISSUE_ID	BOOK_ID	USER_ID	BOOK_ISSUE_DATE
34899345	102	101	2026-05-01
NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit:

USER_ID	USER_NAME	AGE	EMAIL
101	ARNAV	18	arnavsharma@gmail.com
NULL	NULL	NULL	NULL