



**University Institute of Engineering**  
**Department of Computer Science & Engineering**

**Experiment: 1**

**Student Name: Jagrath**

**UID:24BDA70365**

**Branch: CSE**

**Section/Group: AIT-KRG-GP2**

**Semester: 4th**

**Date of Performance:7/01/25**

**Subject Name:DBMS**

**1. Aim of the practical:** To design and implement a Library Management System database using appropriate tables, primary keys, foreign keys, and constraints, and to perform DML operations along with DCL commands such as role creation, privilege granting, and revoking to ensure database security.

**2. Tool Used:**

- **Database Management System:**

- PostgreSQL

- **Database Administration Tool:**

- pgAdmin

**3. Objective:** To gain practical experience in implementing Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Language (DCL) operations in a real database environment. This will also include implementing role-based privileges to secure data.



# University Institute of Engineering

## Department of Computer Science & Engineering

### 4. Practical / Experimental Steps

**Step 1:** Open pgAdmin and connect to the PostgreSQL server using administrator credentials.

**Steps 2:** Create the Books table with appropriate data types, primary key, and constraints to maintain data integrity.

**Steps 3:** Insert sample records into the Books table and perform basic operations such as SELECT, UPDATE, and DELETE.

**Steps 4:** Create the Members table with primary key, NOT NULL, UNIQUE, and CHECK constraints, and insert sample member records.

**Steps 5:** Create the Books\_Issue table with primary key and foreign key references to Books and Members, along with date and validation constraints.

**Steps 6:** Insert sample book issue records and verify the data using SELECT queries.

**Steps 7:** Create a database role named Librarian with LOGIN and password for role-based access control.

**Steps 8:** Grant SELECT, INSERT, UPDATE, and DELETE permissions on required tables to the Librarian role.

**Steps 9:** Revoke the granted permissions from the Librarian role to demonstrate database security control by the administrator.



# University Institute of Engineering

## Department of Computer Science & Engineering

### 5. I / O Analysis

#### DATABASE DESIGN

Query to create Table Books :

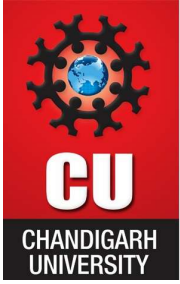
```
CREATE TABLE Books(  
ID INT Primary Key,  
Book_Title Varchar(30),  
Author_Name Varchar(30),  
Count int check(Count>=1) Not NULL,  
)
```

#### To create table members

```
create table members(  
MID INT Primary Key,  
Name varchar(30) NOT NULL,  
age int check(age>=16) NOT NULL,  
email varchar(30) UNIQUE NOT NULL  
)
```

#### To create table Books\_issue

```
create table Books_Issue(  
Issue_No INT Primary Key,  
Books_ID INT references Books(ID) NOT NULL,
```



# University Institute of Engineering

## Department of Computer Science & Engineering

MID INT references members(MID) NOT NULL,  
Issue\_Date DATE NOT NULL,  
Return\_Date DATE check(Issue\_Date <= Return\_date or Return\_Date is NULL),  
Overdue BOOL NOT NULL default false  
)

### To Generate A Role ( by Admin)

create role librarian

with LOGIN PASSWORD '123';

### To Grant Access (by Admin)

grant select,update,delete,insert on Books to librarian;

grant select,update,delete,insert on members to librarian;

grant select,update,delete,insert on Books\_Issue to librarian;



# University Institute of Engineering

## Department of Computer Science & Engineering

### DATA MANIPULATION

Insert Sample records in all tables

insert Into Books(id, Book\_Title, Author\_Name,Count)  
Values (1,'Harry Potter','J K Howling',2),  
(2,'Overlord','Kugane Maruyama',4),  
(3,'5 Second Rule','abc',1);

Showing rows: 1 to 3 Page No: 1 of 1

	id [PK] integer	book_title character varying (30)	author_name character varying (30)	count integer
1	1	Harry Potter	J K Howling	2
2	2	Overlord	Kugane Maruyama	4
3	3	5 Seonds Rule	abc	1

insert into members(MID, Name, age, email)  
Values (101,'Jagrath',18,'jagrath@gmail.com'),  
(102,'Daksh',19,'daksh@gmail.com')

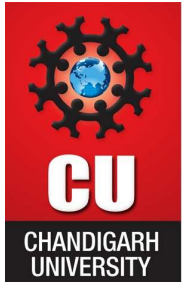
Showing rows: 1 to 2 Page No: 1 of 1

	mid [PK] integer	name character varying (30)	age integer	email character varying (30)
1	101	Jagrath	18	jagrath@gmail.com
2	102	Daksh	19	daksh@gmail.com

insert into Books\_Issue(Issue\_No,Books\_ID,MID,Issue\_Date, Return\_Date)  
Values (500,1,101,'2026-01-01','2026-01-10')

Showing rows: 1 to 1 Page No: 1 of 1

	issue_no [PK] integer	books_id integer	mid integer	issue_date date	return_date date	overdue boolean
1	500	1	101	2026-01-01	2026-01-10	false



# University Institute of Engineering

## Department of Computer Science & Engineering

### Accessing Data Using created Role

```
select * from books;  
select * from members;  
select * from books Issue;
```

Showing rows: 1 to 1		Page No: 1		of 1			
	issue_no [PK] integer	books_id integer	mid integer	issue_date date	return_date date	overdue boolean	
1	500	1	101	2026-01-01	2026-01-10	false	

UPDATE books

set author name = 'abcdef' where id=3

	id [PK] integer	book_title character varying (30)	author_name character varying (30)	count integer
1	1	Harry Potter	J K Howling	2
2	2	Overlord	Kugane Maruyama	4
3	3	5 Seonds Rule	abcdef	1

### Revoke the Access to the Role (by Admin)

Query :

```
revoke select, update, delete, insert on Books from librarian;  
revoke select, update, delete, insert on members from librarian;  
revoke select, update, delete, insert on Books_issue from librarian;
```

(Role attempting access AFTER REVOKE)

Select \* from books;

ERROR: permission denied for table books

SQL state: 42501



# University Institute of Engineering

## Department of Computer Science & Engineering

### 6. Learning outcomes (What I have learnt):

- Understood the basics of **relational database design** using tables, keys, and relationships.
- Learned to apply **primary key and foreign key constraints** to maintain data integrity.
- Gained hands-on experience with **INSERT, UPDATE, and DELETE** operations safely.
- Understood how **roles and privileges** control access to database objects.
- Learned to use **GRANT and REVOKE** for implementing **read-only users**.
- Practiced **ALTER TABLE and DROP TABLE** for managing database changes.