

# Experiment 1

Student Name: Reyansh Arora  
Branch: CSE - AIML  
Semester: 4  
Subject Name: DBMS

UID: 24BAI70273  
Section/Group: 24AIT\_KRG G1  
Date of Performance: 9/1/26  
Subject Code: 24CSH-298

## Aim

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.

## Software Requirements

- Database Management System:
  - PostgreSQL
- Database Administration Tool:
  - pgAdmin

## Objectives

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.

## Problem Statement

- A Library wants to develop a Library Management System database to manage information about books, members, and book issue records efficiently. The database should be designed using appropriate tables, primary keys, foreign keys, and constraints to ensure data integrity.

- The system must support basic database operations such as inserting records, updating existing data, and deleting obsolete entries. To ensure database security.
- To ensure database security, a database role named Librarian must be created. This role should be password protected and granted SELECT, INSERT, and DELETE permissions on the required tables. The system administrator (pgAdmin) should also have the ability to revoke these permissions when required using role-based access control.

## Code

```
TRUNCATE TABLE BOOK_S RESTART IDENTITY
```

```
CREATE TABLE BOOK_S(
BOOK_ID INT PRIMARY KEY,
BOOK_NAME VARCHAR(20) NOT NULL UNIQUE,
AUTHOR_NAME VARCHAR(40) NOT NULL
)
```

```
ALTER TABLE BOOK_S
ADD BOOK_COUNT INT CHECK (BOOK_COUNT>0) NOT NULL
```

```
INSERT INTO BOOK_S VALUES(101,'HARRY POTTER','DAVID',1)
```

```
SELECT * FROM BOOK_S
```

```
CREATE TABLE LIBRARY_VISITORS(
USER_ID INT PRIMARY KEY,
USER_NAME VARCHAR(40) NOT NULL,
AGE INT CHECK(AGE>=17) NOT NULL,
EMAIL VARCHAR(40) NOT NULL UNIQUE
)
```

```
INSERT INTO LIBRARY_VISITORS(USER_ID,USER_NAME,AGE,EMAIL)
VALUES(501,'REYANSH ',19,'REYANSH@GMAIL.COM')
```

```
CREATE TABLE BOOK_ISSUE(
BOOK_ISSUE_ID INT PRIMARY KEY,
USER_ID INT NOT NULL,
BOOK_ID INT NOT NULL,
```

```
FOREIGN KEY(BOOK_ID) REFERENCES BOOK_S(BOOK_ID),
FOREIGN KEY(USER_ID) REFERENCES LIBRARY_VISITORS(USER_ID)
)
```

```
ALTER TABLE BOOK_ISSUE
ADD ISSUE_DATE DATE
```

```
INSERT INTO BOOK_ISSUE VALUES(1001,501,101,'2026-01-09')
```

```
SELECT * FROM BOOK_ISSUE
```

```
UPDATE BOOK_ISSUE
SET ISSUE_DATE ='2026-01-08'
WHERE BOOK_ISSUE_ID=1001
```

```
SELECT * FROM BOOK_ISSUE
```

```
CREATE ROLE LIBRARIAN_2
WITH LOGIN PASSWORD 'REYANSH'
```

```
SELECT CURRENT_USER
```

```
GRANT SELECT, INSERT, DELETE , UPDATE ON BOOK_S TO LIBRARIAN_2
```

```
GRANT SELECT, INSERT, DELETE , UPDATE ON BOOK_ISSUE TO LIBRARIAN_2
```

```
GRANT SELECT, INSERT, DELETE , UPDATE ON LIBRARY_VISITORS TO LIBRARIAN_2
```

```
REVOKE SELECT, INSERT, DELETE , UPDATE ON LIBRARY_VISITORS FROM
LIBRARIAN_2
```

## Output

Table books:

Data Output    Messages    Notifications

The screenshot shows a database interface with a toolbar at the top containing icons for new, open, save, copy, delete, refresh, download, and SQL. The SQL button is highlighted. Below the toolbar is a table structure with four columns: book\_id, book\_name, author\_name, and book\_count. The book\_id column is defined as [PK] integer, book\_name as character varying (20), author\_name as character varying (40), and book\_count as integer. A single row is displayed with values: book\_id 1, book\_name HARRY POTTER, author\_name DAVID, and book\_count 1.

	book_id [PK] integer	book_name character varying (20)	author_name character varying (40)	book_count integer
1	101	HARRY POTTER	DAVID	1

Table library\_visitors:

Data Output    Messages    Notifications

The screenshot shows a database interface with a toolbar at the top containing icons for new, open, save, copy, delete, refresh, download, and SQL. The SQL button is highlighted. Below the toolbar is a table structure with four columns: user\_id, user\_name, age, and email. The user\_id column is defined as [PK] integer, user\_name as character varying (40), age as integer, and email as character varying (20). A single row is displayed with values: user\_id 1, user\_name REYANSIH, age 19, and email REYANSIH@GMAIL.CO...

	user_id [PK] integer	user_name character varying (40)	age integer	email character varying (20)
1	501	REYANSIH	19	REYANSIH@GMAIL.CO...

Table book\_issue:

Data Output    Messages    Notifications

The screenshot shows a database interface with a toolbar at the top containing icons for new, open, save, copy, delete, refresh, download, and SQL. The SQL button is highlighted. Below the toolbar is a table structure with four columns: book\_issue\_id, user\_id, book\_id, and issue\_date. The book\_issue\_id column is defined as [PK] integer, user\_id as integer, book\_id as integer, and issue\_date as date. A single row is displayed with values: book\_issue\_id 1, user\_id 501, book\_id 101, and issue\_date 2026-01-08.

	book_issue_id [PK] integer	user_id integer	book_id integer	issue_date date
1	1001	501	101	2026-01-08

Access granted to role – librarian:

```
GRANT
```

```
Query returned successfully in 44 msec.
```

## **Learning Outcomes**

- Gained hands-on experience to work with PostgreSQL and pgAdmin
- Writing queries to create and delete tables
- Learnt to alter tables, view tables, create roles, granting and revoking access to the roles
- Primary and foreign keys implementations and roles