



Student Name: Punya Arora

Branch: AIT-CSE (AIML)

Semester: 4

Subject Name: Database Management System

UID: 24BAI70928

Section/Group: 24AIT_KRG2

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

```
create table BOOKS (  
    ID INT PRIMARY KEY,  
    NAME VARCHAR(20) NOT NULL,  
    AUTHOR_NAME VARCHAR(20) NOT NULL  
);
```

```
insert into BOOKS  
values(1, 'Harry Potter', 'JK ROWLY');
```

```
alter table BOOKS
```



```
add count INT check(count>=1);
```

```
select * from BOOKS;
```

```
update BOOKS
```

```
set COUNT=3
```

```
where ID=1;
```

```
create table LIBRARY_VISITOR_USER (
```

```
    USER_ID INT PRIMARY KEY,
```

```
    USER_NAME VARCHAR(20),
```

```
    AGE INT CHECK(AGE>=17) NOT NULL,
```

```
    EMAIL VARCHAR(20) UNIQUE NOT NULL
```

```
);
```

```
insert into LIBRARY_VISITOR_USER
```

```
values(101, 'AKASH SHARMA', 17, 'a@gmail.com');
```

```
alter table LIBRARY_VISITOR_USER
```

```
drop column EMAIL;
```

```
alter table LIBRARY_VISITOR_USER
```

```
add column EMAIL VARCHAR(30) UNIQUE;
```

```
update LIBRARY_VISITOR_USER
```

```
set EMAIL='a@gmail.com';
```

```
select * from LIBRARY_VISITOR_USER;
```

```
create table BOOK_ISSUE (
```

```
    BOOK_ISSUE_ID INT PRIMARY KEY,
```



```
BOOK_ID INT REFERENCES BOOKS(ID) NOT NULL,  
USER_ID INT REFERENCES LIBRARY_VISITOR_USER(USER_ID) NOT NULL,  
BOOK_ISSUE DATE NOT NULL  
);
```

```
insert into BOOK_ISSUE  
values(5552, 1, 101, '2026-01-09');
```

```
select * from BOOK_ISSUE;
```

```
create role LIBRARIAN_1  
with LOGIN PASSWORD 'password123';
```

```
grant select, insert, delete, update on BOOKS to LIBRARIAN_1;  
grant select, insert, delete, update on BOOKS_ISSUE to LIBRARIAN_1;  
grant select, insert, delete, update on LIBRARY_VISITOR_USER to LIBRARIAN_1;  
  
revoke select, insert, delete, update on BOOKS from LIBRARIAN_1;
```

2. LIBRARIAN

```
SELECT * FROM books;  
select * from book_issue;  
select * from LIBRARY_VISITOR_USER;
```

```
INSERT INTO books VALUES(110,'ABCD','CLANS',2);  
INSERT INTO books VALUES(150,'THE LORD','HRM',7);
```

```
DELETE FROM books  
WHERE ID=150;
```

```
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 changes.

SCREENSHOTS

25

Data Output Messages Notifications

	id [PK] integer	name character varying (20)	author_name character varying (20)	count integer
1	1	Harry Potter	JK ROWLY	3

Data Output Messages Notifications

	user_id [PK] integer	user_name character varying (20)	age integer	email character varying (30)
1	101	AKASH SHARMA	17	a@gmail.com
2	102	Punya	17	b@gmail.com

Data Output Messages Notifications

	book_issue_id [PK] integer	book_id integer	user_id integer	book_issue date
1	5552	1	101	2026-01-09