Experiment 1.1 – Library Management System Database

Student Name: Samridh Srivastava

UID: 24BDA70208

Section: 24_AIT_KRG_G2

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.

13. Save the work and take screenshots of execution and results.

Input / Output Details

Input

SQL commands for creating database, tables, inserting records, updating data, deleting records, creating roles and granting or revoking permissions.

Output

Successful creation of tables, insertion of records, updated and deleted data, role creation and permission control. Screenshots of SQL execution and output are attached in the repository.

```
18   create table if not exists LIBRARY_VISITOR_USER
19   (
20   USER_ID INT PRIMARY KEY,
21   USER_NAME VARCHAR(20),
22   USER_AGE INT CHECK(USER_AGE>=17),
23   USER_EMAIL VARCHAR(20) UNIQUE NOT NULL
24   );
25
26   INSERT INTO LIBRARY_VISITOR_USER(USER_ID,USER_NAME,USER_AGE,USER_EMAIL)
27   VALUES(101,'SAMRIDH SRIVASTAVA',20,'SAMRIDH@GMAIL.COM');
28
29   SELECT * FROM LIBRARY_VISITOR_USER;
```

Data Output   Messages   Notifications

	user_id [PK] integer	user_name character varying (20)	user_age integer	user_email character varying (20)
1	101	SAMRIDH SRIVASTAVA	20	SAMRIDH@GMAIL.CO...

```
--hw
-- insert delete update on books from library user

INSERT INTO books values(2,'nights','A.D');
select * from books;
UPDATE BOOKS
set book_count = 10
where id = 2;

select * from books;
```

id [PK] integer	name character varying (20)	author_name character varying (20)	book_count integer
1	Harry Porter	JK Rowling	3
2	nights	A.D	10

Learning Outcome

After completing this experiment, the student understood how to design a database using tables and constraints. The student learned how to use DDL, DML and DCL commands in PostgreSQL. This experiment also provided practical exposure to role based access control and database security.