Library Management System Database (PostgreSQL)

Student Name: Mohd Kaif Malik

UID: 24BAI70907

Section: 24AIT-KRG-G2

Aim

To design and implement a Library Management System database using PostgreSQL by creating tables with Primary Keys, Foreign Keys, and constraints, and performing DDL, DML, and DCL operations for data integrity and security.

Software Requirements

- DBMS: PostgreSQL
- Tool: pgAdmin 4

Objective

To gain hands-on practice of DDL (table creation + constraints), DML (insert, update, delete), and DCL (role creation, granting and revoking privileges) in a real database environment.

Problem Statement

A library needs a database system to manage books, members, and issue/return records efficiently. The database must be designed with proper constraints to avoid duplication and maintain accuracy. A secure role named librarian should be created and given limited permissions to ensure role-based access control.

Learning Outcomes

- Learned to operate pgAdmin 4
- Learned to execute SQL queries in PostgreSQL
- Learned to create tables and insert data
- Implemented role-based security using GRANT and REVOKE

Procedure / Steps

Step 1: Database Setup

1. Install PostgreSQL and pgAdmin 4
2. Open pgAdmin 4 and connect to PostgreSQL server
3. Create a database named LibraryManagement
4. Verify the database is created successfully

Step 2: Table Design and Creation

Create the following tables:

Books Table

- BookID (Primary Key)
- Title (NOT NULL)
- Author (NOT NULL)
- ISBN (UNIQUE)
- PublishedYear
- Availability (BOOLEAN)

Members Table

- MemberID (Primary Key)
- Name (NOT NULL)
- Email (UNIQUE)
- PhoneNumber
- MembershipDate
- Status (Active/Inactive)

IssueRecords Table

- IssueID (Primary Key)
- BookID (Foreign Key -> Books.BookID)
- MemberID (Foreign Key -> Members.MemberID)
- IssueDate
- ReturnDate
- ActualReturnDate

Step 3: DML Operations

- INSERT: Added sample records in Books, Members, and IssueRecords
- UPDATE: Updated member details and book availability status
- DELETE: Removed obsolete/damaged books and inactive members

Step 4: DCL Operations (Security)

Create Role: CREATE ROLE librarian WITH PASSWORD 'secure_password';

Grant Privileges: GRANT SELECT, INSERT, DELETE ON Books TO librarian; GRANT SELECT, INSERT, DELETE ON Members TO librarian; GRANT SELECT, INSERT, DELETE ON IssueRecords TO librarian;

Revoke Privileges: REVOKE DELETE ON Books FROM librarian; REVOKE DELETE ON Members FROM librarian; REVOKE DELETE ON IssueRecords FROM librarian;

Step 5: Query Execution and Verification

- Executed SELECT queries to verify inserted records
- Tested INSERT and UPDATE operations
- Verified role-based access by switching to librarian role
- Checked restricted access after REVOKE commands

Step 6: Documentation and Analysis

- Documented all SQL commands used
- Verified data integrity constraints are enforced
- Analyzed the effectiveness of role-based access control
- Summarized security measures implemented

CODE:

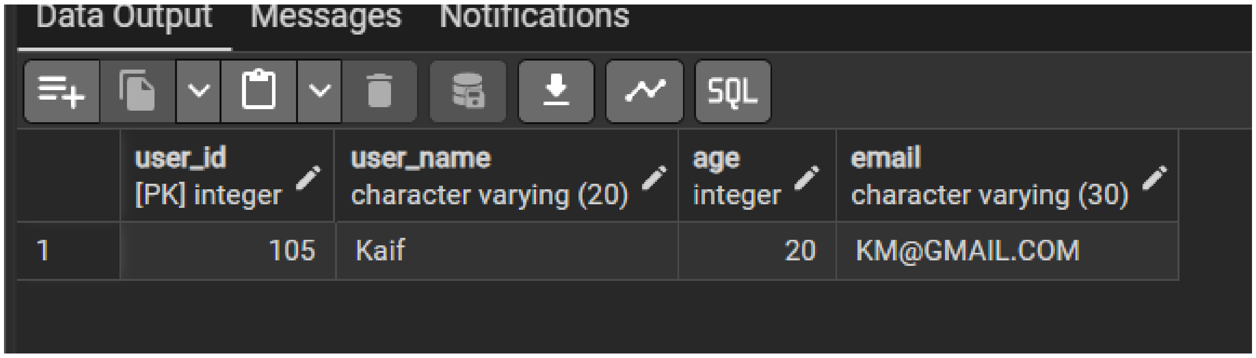
```
CREATE TABLE BOOKS( ID INT PRIMARY KEY, NAME VARCHAR(20) NOT NULL,
AUTHOR_NAME VARCHAR(20) NOT NULL )

INSERT INTO BOOKS VALUES(1,'XYZ','ABC')

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(USER_ID,USER_NAME,AGE,EMAIL) VALUES
(105,'Kaif',20, 'KM@GMAIL.COM')

ALTER TABLE LIBRARY_VISITOR_USER DROP COLUMN EMAIL

SELECT * FROM LIBRARY_VISITOR_USER

ALTER TABLE LIBRARY_VISITOR_USER ADD COLUMN EMAIL VARCHAR(30) UNIQUE

UPDATE LIBRARY_VISITOR_USER SET EMAIL='KM@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(5552,1,105,'2026-01-09')

SELECT * FROM BOOK_ISSUE

CREATE ROLE LIBRARIAN_1 WITH LOGIN PASSWORD 'Malik2026@'
```

GRANT SELECT ,INSERT, DELETE,UPDATE ON BOOKS TO LIBRARIAN_1 GRANT SELECT ,INSERT, DELETE,UPDATE ON BOOK_ISSUE TO LIBRARIAN_1 GRANT SELECT ,INSERT, DELETE,UPDATE ON LIBRARY_VISITOR_USER TO LIBRARIAN_1

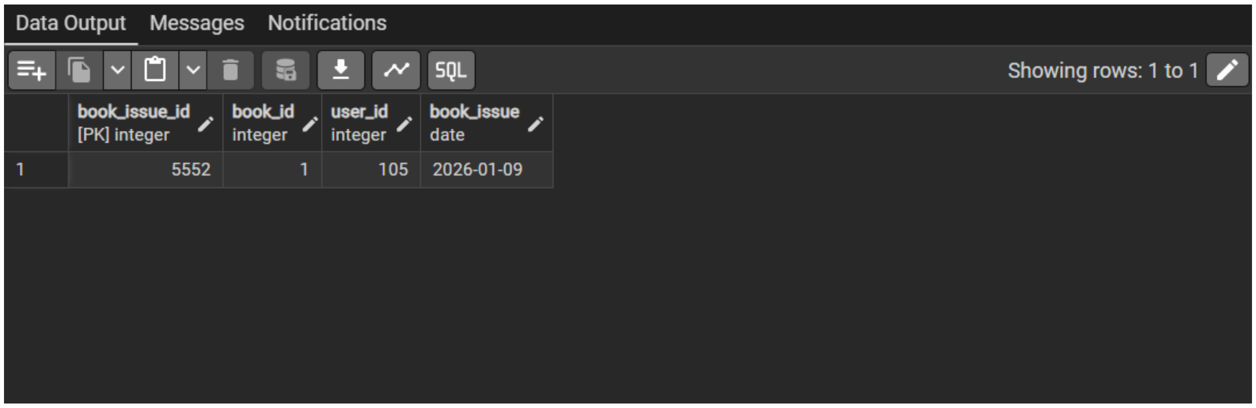
-- HW

REVOKE SELECT ,INSERT, DELETE,UPDATE ON BOOKS FROM LIBRARIAN_1

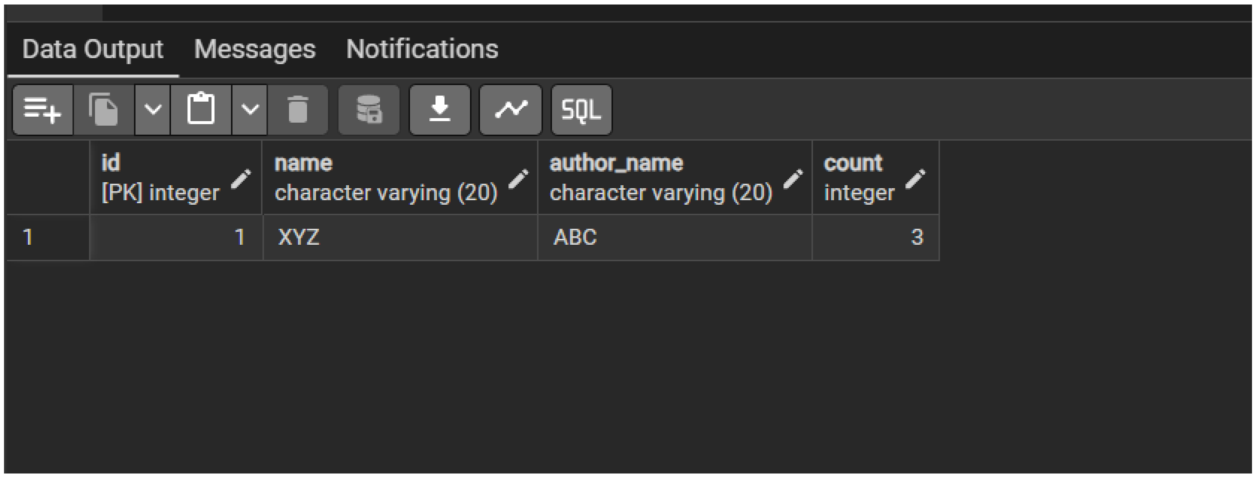
SCREENSHOT:



	user_id [PK] integer	user_name character varying (20)	age integer	email character varying (30)
1	105	Kaif	20	KM@GMAIL.COM



	book_issue_id [PK] integer	book_id integer	user_id integer	book_issue date
1	5552	1	105	2026-01-09



	id [PK] integer	name character varying (20)	author_name character varying (20)	count integer
1	1	XYZ	ABC	3