EXPERIMENT : 1

Student Name: Pushpit kumar gaur	UID: 24BAI70534
Branch: CSE - AIML	Section/Group: 24AIT_KRG G1
Semester: 4	Date of Performance: 9/1/26
Subject Name: DBMS	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

1. To implement Data Definition Language (DBL) commands for creating, altering, and deleting database tables with appropriate constraints.
2. To apply Data Manipulation Language (DML) commands to insert, update, retrieve, and manage records while maintaining data integrity.
3. To understand Data Control Language (DCL) by creating user roles and managing database security through granting and revoking privileges.

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.

Steps :

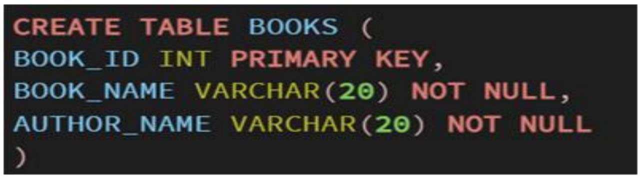
1. A table named BOOKS was created to store book details such as Book ID, Book Name, and Author Name. Book_ID is set as the primary key.
2. The table BOOKS was modified using ALTER command to add a new column BOOK_COUNT with a check constraint to ensure that the BOOK_COUNT is greater than 0 and is not null.
3. Records were inserted into the BOOKS table using INSERT commands and verified using SELECT queries.
4. A new table LIBRARY_VISITORS was created with attributes USER_ID, NAME, AGE and EMAIL. USER_ID is set as the primary key and data is inserted in the table using INSERT command.
5. A new table BOOK_ISSUE was created to maintain book issue records, implementing FOREIGN KEY relationships between BOOKS and LIBRARY_VISITORS tables. USER_ID from LIBRARY_VISITORS table and BOOK_ID from BOOKS are set as the foreign key.
6. The ALTER table command is used to add ISSUE_DATE for issued books. UPDATE command is used to update the previous records of the table.
7. A new user role named LIBRARIAN was created, and permissions were granted using GRANT.
8. Permissions were later revoked using the REVOKE command to understand access control.
9. TRUNCATE and DROP commands were executed to observe the difference between removing records and deleting tables.

Code

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

SELECT * FROM BOOKS

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

SELECT * FROM BOOKS

INSERT INTO BOOKS VALUES(101, 'Harry Potter', 'JK Rowling', 3)
INSERT INTO BOOKS VALUES(102, '1984', 'Neo', 5)

SELECT * FROM BOOKS

CREATE TABLE LIBRARY_VISITORS(
USER_ID INT PRIMARY KEY,
NAME VARCHAR(20) NOT NULL,
AGE INT CHECK(AGE>=17) NOT NULL,
EMAIL VARCHAR(20) NOT NULL UNIQUE
)

SELECT * FROM LIBRARY_VISITORS

INSERT INTO LIBRARY_VISITORS(USER_ID, NAME, AGE, EMAIL)
VALUES(101, 'Vibhuti dhiman', 18, 'vibhu190@gmail.com')

INSERT INTO LIBRARY_VISITORS(USER_ID, NAME, AGE, EMAIL)
VALUES(101, 'kartik aryan', 18, 'kartik18@gmail.com')


DROP TABLE BOOK_ISSUE
CREATE TABLE BOOK_ISSUE(
BOOK_ISSUE_ID INT PRIMARY KEY,
USER_ID INT NOT NULL,
BOOK_ID INT NOT NULL,
FOREIGN KEY(USER_ID) REFERENCES LIBRARY_VISITORS(USER_ID),
FOREIGN KEY(BOOK_ID) REFERENCES BOOKS(BOOK_ID)
)

INSERT INTO BOOK_ISSUE VALUES(10001, 501, 101)

ALTER TABLE BOOK_ISSUE
ADD ISSUE_DATE DATE

SELECT * FROM BOOK_ISSUE

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

UPDATE BOOK_ISSUE
SET ISSUE_DATE='2026-01-09'
WHERE BOOK_ISSUE_ID=1001


DELETE FROM BOOKS WHERE BOOK_ID=102

SELECT * FROM BOOKS

CREATE ROLE LIBRARIAN
WITH LOGIN PASSWORD 'Pushpit9674'

SELECT CURRENT_USER

GRANT SELECT, INSERT, DELETE, UPDATE ON BOOKS TO LIBRARIAN2
GRANT SELECT, INSERT, DELETE, UPDATE ON BOOK_ISSUE TO LIBRARIAN2
GRANT SELECT, INSERT, DELETE, UPDATE ON LIBRARY_VISITORS TO LIBRARIAN2

REVOKE SELECT, INSERT, DELETE, UPDATE ON BOOKS, BOOK_ISSUE,
LIBRARY_VISITORS FROM LIBRARIAN2
```

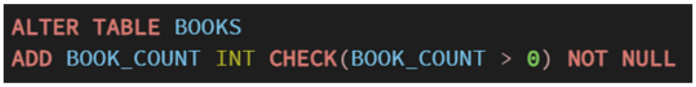
Output

CREATE TABLE :



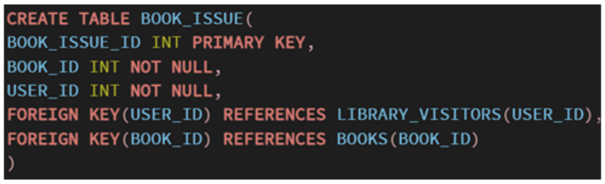
```
CREATE TABLE BOOKS (
BOOK_ID INT PRIMARY KEY,
BOOK_NAME VARCHAR(20) NOT NULL,
AUTHOR_NAME VARCHAR(20) NOT NULL
)
```

ALTER TABLE :



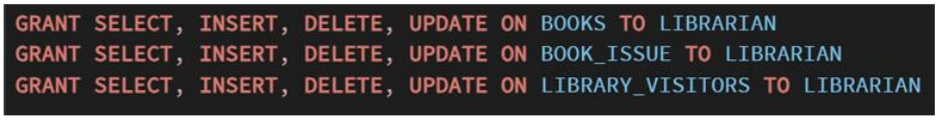
```
ALTER TABLE BOOKS
ADD BOOK_COUNT INT CHECK(BOOK_COUNT > 0) NOT NULL
```

CREATE NEW TABLE AND INSERT DATA INTO IT :



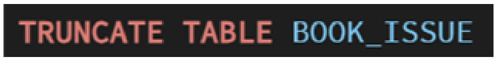
```
CREATE TABLE BOOK_ISSUE(
BOOK_ISSUE_ID INT PRIMARY KEY,
BOOK_ID INT NOT NULL,
USER_ID INT NOT NULL,
FOREIGN KEY(USER_ID) REFERENCES LIBRARY_VISITORS(USER_ID),
FOREIGN KEY(BOOK_ID) REFERENCES BOOKS(BOOK_ID)
)
```

GRANT COMMAND :



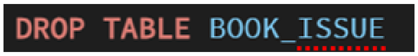
```
GRANT SELECT, INSERT, DELETE, UPDATE ON BOOKS TO LIBRARIAN
GRANT SELECT, INSERT, DELETE, UPDATE ON BOOK_ISSUE TO LIBRARIAN
GRANT SELECT, INSERT, DELETE, UPDATE ON LIBRARY_VISITORS TO LIBRARIAN
```

Truncate :



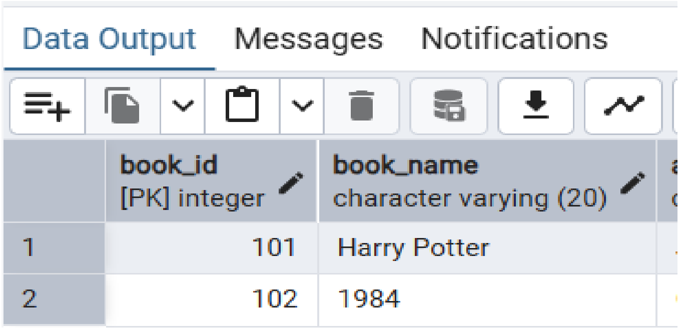
```
TRUNCATE TABLE BOOK_ISSUE
```

DROP COMMAND :



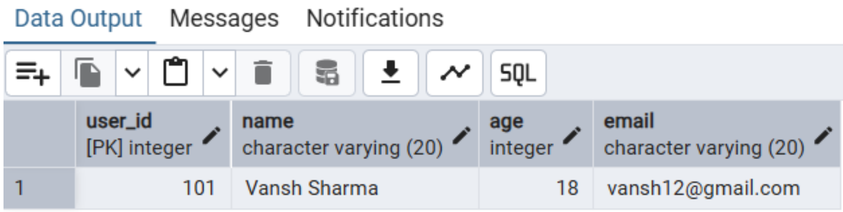
```
DROP TABLE BOOK_ISSUE
```

Table books:



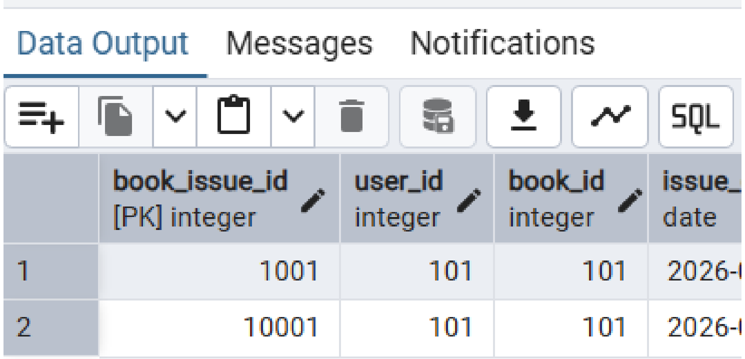
	book_id [PK] integer	book_name character varying (20)
1	101	Harry Potter
2	102	1984

Table library_visitors:



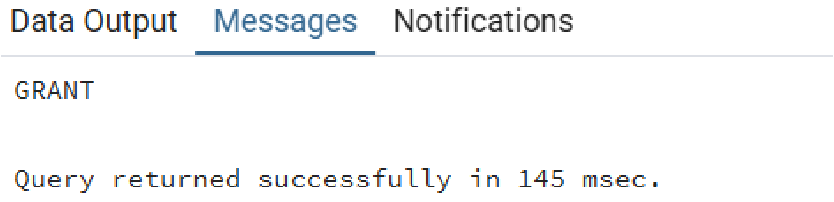
	user_id [PK] integer	name character varying (20)	age integer	email character varying (20)
1	101	Vansh Sharma	18	vansh12@gmail.com

Table book_issue:



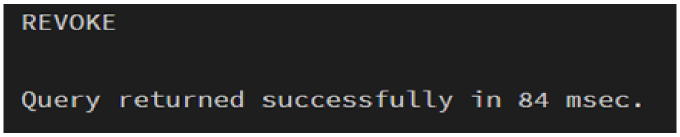
	book_issue_id [PK] integer	user_id integer	book_id integer	issue_ date
1	1001	101	101	2026-
2	10001	101	101	2026-

Access granted to role – librarian:



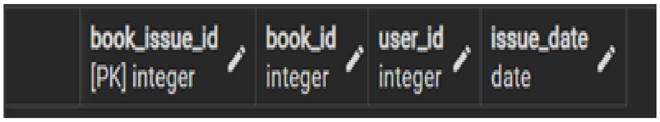
```
GRANT

Query returned successfully in 145 msec.
```

QUERY REVOKE :



```
REVOKE

Query returned successfully in 84 msec.
```

TRUNCATE TABLE :



	book_issue_id [PK] integer	book_id integer	user_id integer	issue_date date

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