Apex Institute of Technology

Computer Science & Engineering

Worksheet 1

Name: Trimann Kaur **UID: 24BAI70511**

Branch: B.E. CSE (AIML) **Section: 24AIT_KRG-G1**

Semester: 4 **Date of Performance: 09.01.2026**

Subject Name: Database Management System **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.

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.

SOFTWARE REQUIREMENTS:

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

PROBLEM STATEMENT:

1. 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.

2. The system must support basic database operations such as inserting records, updating existing data, and deleting obsolete entries. To ensure database security.

3. 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:

```
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', 'Rowling', 3)
INSERT INTO BOOKS VALUES(102, 'The Alchemist', 'Paulo', 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(501, 'vansh', 19, 'vansh@gmail.com')
INSERT INTO LIBRARY_VISITORS(USER_ID, NAME, AGE, EMAIL)
VALUES(502, 'ansh', 19, 'ansh@gmail.com')

SELECT * FROM LIBRARY_VISITORS

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)
)
INSERT INTO BOOK_ISSUE VALUES(10001, 101, 501)

SELECT * FROM BOOK_ISSUE

ALTER TABLE BOOK_ISSUE
ADD ISSUE_DATE DATE

SELECT * FROM BOOK_ISSUE

UPDATE BOOK_ISSUE
SET ISSUE_DATE='2026-01-08'
WHERE BOOK_ID=101

SELECT * FROM BOOK_ISSUE

CREATE ROLE LIBRARIAN
WITH LOGIN PASSWORD '12345678'

SELECT CURRENT_USER

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
```

Apex Institute of Technology

Computer Science & Engineering

-- LIBRARIAN
SELECT * FROM BOOKS
SELECT * FROM BOOK_ISSUE
SELECT * FROM LIBRARY_VISITORS

REVOKE SELECT, INSERT, DELETE, UPDATE ON BOOKS, BOOK_ISSUE, LIBRARY_VISITORS FROM LIBRARIAN

SELECT * FROM BOOKS

TRUNCATE TABLE BOOK_ISSUE
SELECT * FROM BOOK_ISSUE

DROP TABLE BOOK_ISSUE
SELECT * FROM BOOK_ISSUE

I/O ANALYSIS:

- Create table BOOKS.

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

book_id [PK] integer	book_name character varying (20)	author_name character varying (20)

· Alter and insert into table BOOKS.

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

```
INSERT INTO BOOKS VALUES(101, 'Harry Potter', 'Rowling', 3)
INSERT INTO BOOKS VALUES(102, 'The Alchemist', 'Paulo', 5);
```

	book_id [PK] integer	book_name character varying (20)	author_name character varying (20)	book_count integer
1	101	Harry Potter	Rowling	3
2	102	The Alchemist	Paulo	5

· Create table LIBRARY_VISITORS.

Apex Institute of Technology

Computer Science & Engineering

```
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)
)
```

	user_id [PK] integer	name character varying (20)	age integer	email character varying (20)

- Insert into LIBRARY_VISITORS.

```
INSERT INTO LIBRARY_VISITORS(USER_ID, NAME, AGE, EMAIL)
VALUES(501, 'vansh', 19, 'vansh@gmail.com')
```

```
INSERT INTO LIBRARY_VISITORS(USER_ID, NAME, AGE, EMAIL)
VALUES(502, 'ansh', 19, 'ansh@gmail.com')
```

	user_id [PK] integer	name character varying (20)	age integer	email character varying (20)
1	501	vansh	19	vansh@gmail.com
2	502	ansh	19	ansh@gmail.com

- Create table BOOK_ISSUE with foreign key. Alter and update it.

	book_issue_id [PK] integer	book_id integer	user_id integer
1	10001	101	501

- Alter table BOOK_ISSUE.

	book_issue_id [PK] integer	book_id integer	user_id integer	issue_date date
1	10001	101	501	[null]

- Update BOOK_ISSUE.

	book_issue_id [PK] integer	book_id integer	user_id integer	issue_date date
1	10001	101	501	2026-01-08

- Creating a new role, and granting access.

```
GRANT

Query returned successfully in 116 msec.
```

- Revoke access.

```
REVOKE

Query returned successfully in 84 msec.
```

```
ERROR:  permission denied for table books

SQL state: 42501
```

- Truncate table BOOK_ISSUE.

book_issue_id [PK] integer	book_id integer	user_id integer	issue_date date

- Drop table BOOK_ISSUE.

```
ERROR:  relation "book_issue" does not exist
```

LEARNING OUTCOMES:

1. Understanding of creating and managing database structures using DBL commands.
2. Inserting, updating, and retrieving data using DML queries while maintaining data integrity.
3. Knowledge of controlling database access by creating roles and applying DCL commands.