



Experiment No. 1.1

Student Name: Roshan Kumar Singh
Branch: MCA(GEN)
Semester: 2nd
Subject Name: Technical Training-1

UID: 25MCA20067
Section/Group: 25MCA-1_A
Date of Performance: - 6/01/26
Subject Code: 25CAP-652

Aim:

To design and implement a sample database system using DDL, DML, and DCL commands, including database creation, data manipulation, schema modification, and role-based access control to ensure data integrity and secure, read-only access for authorized users.

Objective:

To gain practical experience in implementing Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Language (DCL) operations in a real database environment. This will also include implementing role-based privileges to secure data.

An organization wants to design a sample database system to manage Departments, Employees, and Projects. The database must ensure data integrity, controlled access, and proper privilege management for different users.



Requirements:

1. Database Design:
 - Create multiple tables such as Department, Employee, and Project.
 - Define appropriate PRIMARY KEY and FOREIGN KEY constraints.
 - Enforce NOT NULL, UNIQUE, and CHECK constraints where necessary.
2. Data Manipulation
 - Insert sample records into all tables.
 - Perform UPDATE operations to modify existing records.
 - Perform DELETE operations while maintaining referential integrity.
3. Access Control & Security
 - Create a role/user for a reporting staff member.
 - Grant ONLY SELECT privilege on required tables to this role/user.
 - Explicitly REVOKE CREATE privilege so that the user cannot create any database objects.
 - Ensure the user has read-only access to the database.
4. Schema Modification
 - Use ALTER TABLE to add or modify a column.
 - Drop a table that is no longer required using DROP TABLE.

S/W Requirement:

Oracle Database Express Edition and pgAdmin



Procedure:

1. Start PostgreSQL Environment

Open pgAdmin and connect to the PostgreSQL server using administrator (superuser) credentials.

2. Create Tables Using DDL Commands

Create the Department, Employee, and Project tables using CREATE TABLE statements. Define PRIMARY KEY and FOREIGN KEY constraints to maintain referential integrity.

Apply NOT NULL, UNIQUE, and CHECK constraints to ensure data consistency.

3. Insert Sample Records

Insert sample data into all tables using INSERT statements to populate the database.

4. Perform Data Manipulation Operations

Modify existing records using the UPDATE command.

Remove records using the DELETE command while maintaining referential integrity.

5. Check Current User

Use SELECT CURRENT_USER; to verify the active database user in pgAdmin.

6. Create Role for Reporting Staff

Create a new role for reporting staff using the CREATE ROLE command with login privileges.

7. Grant Read-Only Access

Grant ONLY SELECT privilege on Department, Employee, and Project tables to the reporting staff role.

8. Restrict Unauthorized Operations



Revoke INSERT, UPDATE, and DELETE privileges to prevent data modification.
Revoke CREATE privilege on the public schema to restrict object creation.

9. Modify Database Schema

Use the ALTER TABLE command to modify the structure of the Employee table.

10. Drop Unnecessary Table

Remove the Project table using the DROP TABLE command when it is no longer required.

11. Verify Permissions and Output

Log in using the reporting staff role and verify that only read-only access is allowed.

Code:

```
-- 1
CREATE TABLE Department (
    dpt_id INT PRIMARY KEY,
    dpt_name VARCHAR(30) UNIQUE NOT NULL
);

CREATE TABLE Employee (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(30) NOT NULL,
    salary INT CHECK (salary > 0),
    dpt_id INT,
    CONSTRAINT fk_dpt FOREIGN KEY (dpt_id) REFERENCES Department(dpt_id)
);

CREATE TABLE Project (
    project_id INT PRIMARY KEY,
    project_name VARCHAR(30) NOT NULL,
    dpt_id INT,
    CONSTRAINT fk_project_dpt FOREIGN KEY (dpt_id) REFERENCES
Department(dpt_id)
);
```



-- 2

```
INSERT INTO Department VALUES
(1, 'HR'),
(2, 'IT'),
(3, 'Sales'),
(4, 'Finance');
```

```
INSERT INTO Employee VALUES
(101, 'Roshan', 50000, 2),
(102, 'Sanchit', 45000, 1),
(103, 'Riya', 55000, 3),
(104, 'Swayam', 60000, 4);
```

```
INSERT INTO Project VALUES
(201, 'Payroll System', 2),
(202, 'Web Application', 2);
```

```
UPDATE Employee SET salary = 55000 WHERE emp_id = 101;
```

```
DELETE FROM Project WHERE project_id = 202;
```

--3

```
SELECT CURRENT_USER;
```

```
CREATE ROLE report_staff WITH LOGIN PASSWORD 'staff1';
```

```
SELECT CURRENT_USER;
```

```
GRANT SELECT ON Department TO report_staff;
GRANT SELECT ON Employee TO report_staff;
GRANT SELECT ON Project TO report_staff;
```

```
REVOKE CREATE ON SCHEMA public FROM report_staff;
```

```
REVOKE INSERT, UPDATE, DELETE ON Department FROM report_staff;
REVOKE INSERT, UPDATE, DELETE ON Employee FROM report_staff;
REVOKE INSERT, UPDATE, DELETE ON Project FROM report_staff;
```

--4

```
ALTER TABLE Employee
```



ALTER COLUMN emp_name TYPE VARCHAR(50);

DROP TABLE Project;

Output:

Department Table

Data Output Messages Notifications			
	dpt_id [PK] integer	dpt_name character varying (30)	
1	1	HR	
2	2	IT	
3	3	Sales	
4	4	Finance	

Employee Table

Data Output Messages Notifications				
	emp_id [PK] integer	emp_name character varying (50)	salary integer	dpt_id integer
1	101	Roshan	50000	2
2	102	Sanchit	45000	1
3	103	Riya	55000	3
4	104	Swayam	60000	4



Project Table

	project_id [PK] integer	project_name character varying (30)	dpt_id integer
1	201	Payroll System	2
2	202	Web Application	2

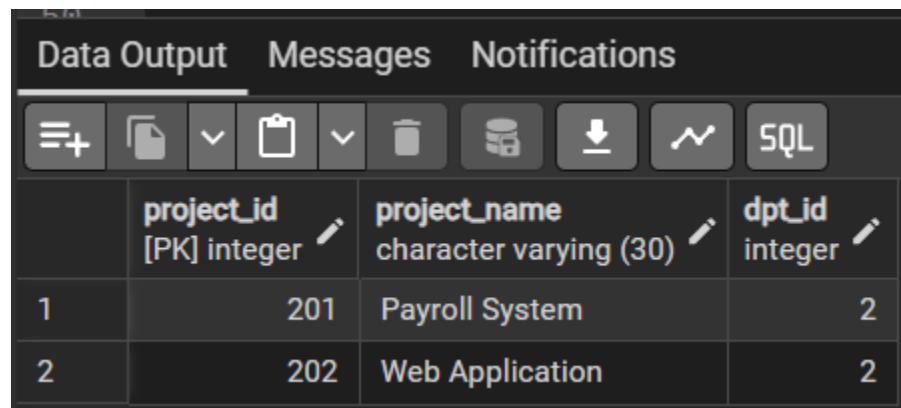
Created new reporting staff

	current_user name
1	report_staff

Granted only select permission and restricted insert, update, delete and create privileges.

```
ERROR: must be owner of table employee
SQL state: 42501
```

Select project table accessed using report staff user



	project_id [PK] integer	project_name character varying (30)	dpt_id integer
1	201	Payroll System	2
2	202	Web Application	2

Learning Outcomes:

- Learnt to create tables using DDL commands with appropriate constraints.
- Learnt Insert, update, and delete records using DML commands.
- Learnt to modify database schema using ALTER TABLE.
- Learnt to implement role-based access control using DCL commands.
- Learnt to grant restricted privileges and revoke permissions to ensure secure, read only access for authorized users.