Branch: *MCA (Data Science)*	**Semester:** *2*
Student Name: *Adarsh Mishra*	**UID:** *25MCD10065*
Subject Name: *Technical Training - I*	**Subject Code**: 25CAP-652
Section/Group: 25MCD – 1(A)	**Date of Performance:** *13 January,2026*

Experiment No. : 1

1. **Aim/Overview of the practical:**
   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.

2. **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.

3. **Software Used:**
   - PostgreSQL
   - pgAdmin4

4. **Code for experiment/Practical:**

```
CREATE TABLE Department (
    Dept_ID INT PRIMARY KEY,
    Dept_Name VARCHAR(50) UNIQUE NOT NULL,
    Location VARCHAR(50) NOT NULL
);

CREATE TABLE Employee (
    Emp_ID INT PRIMARY KEY,
    Emp_Name VARCHAR(50) NOT NULL,
    Salary INT CHECK (Salary > 0),
    Dept_ID INT REFERENCES Department(Dept_ID),
    Email VARCHAR(100) UNIQUE
);
```

```
CREATE TABLE Project (
    Project_ID INT PRIMARY KEY,
    Project_Name VARCHAR(50) NOT NULL,
    Budget INT CHECK (Budget >= 10000),
    Dept_ID INT REFERENCES Department(Dept_ID)

);

INSERT INTO Department VALUES
(1, 'HR', 'Mumbai'),
(2, 'IT', 'Pune'),
(3, 'Finance', 'Delhi');

INSERT INTO Employee VALUES
(101, 'Akash', 50000, 2, 'akash@org.in'),
(102, 'Boby', 45000, 1, 'boby@org.in'),
(103, 'Chetan', 60000, 2, 'chetan@org.in');

INSERT INTO Project VALUES
(201, 'Payroll System', 200000, 1),
(202, 'Web Application', 500000, 2);

SELECT * FROM Department;
SELECT * FROM Employee;
SELECT * FROM Project;

UPDATE Employee
SET Salary = 55000
WHERE Emp_ID = 101;

DELETE FROM Department
WHERE Dept_ID = 3;

CREATE ROLE Analyst
LOGIN PASSWORD 'analyst123'

GRANT SELECT ON Department TO Analyst;
GRANT SELECT ON Employee TO Analyst;
GRANT SELECT ON Project TO Analyst;

REVOKE CREATE ON DATABASE DB1 FROM analyst;
```

ALTER TABLE Employee
ADD Phone_No INT(10);

ALTER TABLE Employee
ALTER COLUMN Emp_Name TYPE VARCHAR(50);

DROP TABLE Project;

5. Result/Output/Writing Summary:

	dept_id [PK] integer	dept_name character varying (50)	location character varying (50)
1	1	HR	Mumbai
2	2	IT	Pune
3	3	Finance	Delhi

	emp_id [PK] integer	emp_name character varying (50)	salary integer	dept_id integer	email character varying (100)
1	101	Akash	50000	2	akash@org.in
2	102	Boby	45000	1	boby@org.in
3	103	Chetan	60000	2	chetan@org.in

	project_id [PK] integer	project_name character varying (50)	budget integer	dept_id integer
1	201	Payroll System	200000	1
2	202	Web Application	500000	2

After Update Query, Employee:

	emp_id [PK] integer	emp_name character varying (50)	salary integer	dept_id integer	email character varying (100)
1	102	Boby	45000	1	boby@org.in
2	103	Chetan	60000	2	chetan@org.in
3	101	Akash	55000	2	akash@org.in

After Delete Query, Department:

	dept_id [PK] integer	dept_name character varying (50)	location character varying (50)
1	1	HR	Mumbai
2	2	IT	Pune

After Alter Query, Employee:

	emp_id [PK] integer	emp_name character varying (50)	salary integer	dept_id integer	email character varying (100)	phone_no integer
1	102	Boby	45000	1	boby@org.in	[null]
2	103	Chetan	60000	2	chetan@org.in	[null]
3	101	Akash	55000	2	akash@org.in	[null]

After Drop Table Query, Project:

```
ERROR:  relation "project" does not exist
LINE 1: SELECT * FROM Project;
                      ^
```

6. I/O Analysis (Input / Output)
Input:
• Department, Employee, and Project table creation queries
• Records inserted into all tables using INSERT commands
• Update query to modify employee department
• Delete queries to remove project and employee records
• Role creation and privilege assignment queries

• ALTER and DROP table commands

Output:

• Department, Employee, and Project tables created
successfully

• Records inserted, updated, and deleted correctly

• Referential integrity maintained between tables

• Data displayed correctly using SELECT queries

• Role-based access verified using GRANT and REVOKE

• Table structure modified and project table dropped
successfully

Learning outcomes (What I have learnt):

- Understand how to design a relational database using multiple tables with proper relationships.
- Learn to apply constraints to maintain data integrity and consistency.
- Perform basic data manipulation operations such as INSERT, UPDATE, and DELETE.
- Implement database security by managing users, roles, and access privileges.