

<b>Student Name:</b> Drishti Bhatia	<b>UID:</b> 25MCD10008
<b>Branch:</b> Master in Computer Applications (Data Science)	<b>Section/Group:</b> A
<b>Semester:</b> 2	<b>Date of Performance:</b> 13-01-2026
<b>Subject Name:</b> Technical Training	<b>Subject Code:</b> 25CAP-652

## **Experiment No: 1.1**

### **1. Aim 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. S/W Requirement:** Oracle Database Express Edition and PGAdmin.

### **3. 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.

### **4. Code :**

#### **Creation of tables:**

```

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)
);
```

**Inserting values into tables:**

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

```
INSERT INTO Employee VALUES
(101, 'Alice', 50000, 2, 'alice@org.com'),
(102, 'Bob', 45000, 1, 'bob@org.com'),
(103, 'Charlie', 60000, 2, 'charlie@org.com');
```

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

**Viewing tables:**

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

**Updating table Employee:**

```
UPDATE Employee
SET Salary = 55000
WHERE Emp_ID = 101;
```

**Deleting from table:**

```
DELETE FROM Department
WHERE Dept_ID = 3;
```

**Creating role:**

CREATE ROLE Analyst  
With LOGIN PASSWORD 'analyst111'

**Grant permissions:**

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

**Revoke permissions:**

REVOKE CREATE ON DATABASE tpp FROM Analyst;

**Altering table:**

ALTER TABLE Employee  
ADD Phone\_No VARCHAR(15);

ALTER TABLE Employee  
ALTER COLUMN Emp\_Name TYPE VARCHAR(100);

**Deleting table:**

DROP TABLE Project;

## 5. Output:

**Department table:**

	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

**Employee table:**

	emp_id [PK] integer	emp_name character varying (50)	salary integer	dept_id integer	email character varying (100)
1	101	Alice	50000	2	alice@org.com
2	102	Bob	45000	1	bob@org.com
3	103	Charlie	60000	2	charlie@org.com

**Project table:**

	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:**

	emp_id [PK] integer	emp_name character varying (50)	salary integer	dept_id integer	email character varying (100)
1	102	Bob	45000	1	bob@org.com
2	103	Charlie	60000	2	charlie@org.com
3	101	Alice	55000	2	alice@org.com

**After deleting from table:**

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

**After alter query:**

	emp_id [PK] integer	emp_name character varying (100)	salary integer	dept_id integer	email character varying (100)	phone_no character varying (15)
1	102	Bob	45000	1	bob@org.com	[null]
2	103	Charlie	60000	2	charlie@org.com	[null]
3	101	Alice	55000	2	alice@org.com	[null]

**Revoke permissions:**

```
ERROR: permission denied for schema public
LINE 1: create table manager
          ^

SQL state: 42501
Character: 14
```

## 6. Learning Outcomes:

- 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.