UNIVERSITY INSTITUTE OF COMPUTING
Discover. Learn. Empower.

Experiment 1

Student Name: Sumanyu Seth **UID: 25MCI10040**
Branch: MCA (AI & ML) **Section/Group: 25MAM_KAR-1_A**
Semester: II **Date of Performance: 12/01/26**
Subject Name: Technical Training **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.

S/W Requirement: Oracle Database Express Edition and pgAdmin

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**.

Implementation/Code:

```
CREATE TABLE Department ( dept_id INT PRIMARY KEY, dept_name VARCHAR(50) NOT NULL UNIQUE, location VARCHAR(50) );

CREATE TABLE Employee ( emp_id INT PRIMARY KEY, emp_name VARCHAR(50) NOT NULL, salary INT CHECK (salary > 0), dept_id INT, CONSTRAINT fk_dept FOREIGN KEY (dept_id) REFERENCES Department(dept_id) ON DELETE SET NULL );
```

```
CREATE TABLE Project ( project_id INT PRIMARY KEY, project_name VARCHAR(50) NOT NULL, dept_id INT NOT NULL, CONSTRAINT fk_project_dept FOREIGN KEY (dept_id) REFERENCES Department(dept_id) );

INSERT INTO Department VALUES (1, 'HR', 'Delhi');
INSERT INTO Department VALUES (2, 'IT', 'Bangalore');
INSERT INTO Department VALUES (3, 'Finance', 'Mumbai');
INSERT INTO Employee VALUES (101, 'Amit', 50000, 2);
INSERT INTO Employee VALUES (102, 'Riya', 45000, 1);
INSERT INTO Employee VALUES (103, 'Karan', 60000, 2);

INSERT INTO Project VALUES (201, 'Payroll System', 1);
INSERT INTO Project VALUES (202, 'Website Upgrade', 2);

UPDATE Employee SET salary = 55000 WHERE emp_id = 101;

DELETE FROM Department WHERE dept_id = 3;

CREATE ROLE reporting_staff LOGIN PASSWORD 'reportS1' ;

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

REVOKE CREATE ON SCHEMA public FROM reporting_staff;

ALTER TABLE Employee ADD email VARCHAR(50);

DROP TABLE Project;
```

OUTPUT:

SELECT * FROM department;

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

SELECT * FROM employee;

	emp_id [PK] integer	emp_name character varying (50)	salary integer	dept_id integer	email character varying (50)
1	102	Riya	45000	1	[null]
2	103	Karan	60000	2	[null]
3	101	Amit	55000	2	[null]

LEARNING OUTCOMES:

- Design a relational database using proper table structure and relationships.
- Apply **DDL commands** to create, alter, and drop database objects.
- Use **DML commands** to insert, update, and delete records while maintaining data integrity.
- Implement **PRIMARY KEY and FOREIGN KEY constraints** to enforce referential integrity.
- Understand and apply **CHECK, NOT NULL, and UNIQUE constraints** effectively.
- Implement **role-based access control** using DCL commands.
- Grant and revoke privileges to ensure **secure, read-only access** for authorized users.
- Gain hands-on experience with database security and schema modification.