UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+
Accredited University

Experiment- 1

Branch: MCA (AI&ML)	**Semester: 2**
Student Name: Ajay Rakwal	**UID:** 25MCI10329
Subject Name: Technical Training	**Subject Code**: 25CAP-652
Section/Group: MAM-1(A)	**Date of Performance:** 11-01-2026

Aim of the program:

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.

Software Requirements

- **Operating System:** Windows / Linux
- **Database Management System:** MySQL / Oracle / PostgreSQL
- **SQL Interface:** MySQL Workbench /Web Based / 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.

Procedure of the Experiment

1. Open pgAdmin and connect to the PostgreSQL server.
2. Create the Course table with appropriate primary key and NOT NULL constraints.

3. Create the Student table with primary key, UNIQUE, and CHECK constraints.
4. Create the Enrollment table with foreign key relationships to Student and Course tables.
5. Insert sample data into the Course, Student, and Enrollment tables.
6. Update records in the Student table and delete records from the Enrollment table.
7. Create a new role and grant SELECT privileges on the tables.
8. Alter the Student table to add a new column.
9. Revoke privileges from the role and drop the Enrollment table.

Practical / Experiment Steps

```
-- course table
CREATE TABLE Course (
    course_id NUMBER PRIMARY KEY,
    course_name VARCHAR(50) NOT NULL
);
--student table
CREATE TABLE Student (
    student_id NUMBER PRIMARY KEY,
    student_name VARCHAR(50) NOT NULL,
    email VARCHAR(50) UNIQUE,
    age NUMBER CHECK (age >= 18)
);


-- enrollment table
```

```
CREATE TABLE enrollment (
    enroll_id INT PRIMARY KEY,
    student_id INT,
    course_id INT,
    CONSTRAINT fk_student
        FOREIGN KEY (student_id)
        REFERENCES student(student_id),
    CONSTRAINT fk_course
        FOREIGN KEY (course_id)
        REFERENCES course(course_id)
);

-- data insertion
INSERT INTO course VALUES (1, 'Artificial Intelligence');
INSERT INTO course VALUES (2, 'Data Science');
INSERT INTO student VALUES (101, 'Ajay', 22);
INSERT INTO student VALUES (102, 'Ravi', 23);
INSERT INTO enrollment VALUES (1, 101, 1);
INSERT INTO enrollment VALUES (2, 102, 2);

--data updatate
UPDATE student
SET age = 23
WHERE student_id = 101;
```

```
--data delete

DELETE FROM enrollment

WHERE enroll_id = 2;


-- new role created

CREATE ROLE report_user LOGIN PASSWORD 'report123';


-- grant

GRANT SELECT ON course TO report_user;

GRANT SELECT ON student TO report_user;

GRANT SELECT ON enrollment TO report_user;

--add phone number

ALTER TABLE student

ADD COLUMN phone_no VARCHAR(15);


-- revovke

REVOKE ALL ON course FROM report_user;

REVOKE ALL ON student FROM report_user;

REVOKE ALL ON enrollment FROM report_user;

--table drop

DROP TABLE enrollment;
```

I/O Analysis (Input / Output)

Input

- Course, Student, and Enrollment table creation queries
- Records inserted into all tables using INSERT commands
- UPDATE query to modify student details
- DELETE query to remove enrollment records
- Role creation and privilege assignment queries
- ALTER TABLE command to add a new column
- REVOKE and DROP TABLE commands

Output

- Course, Student, and Enrollment tables created successfully
- Records inserted, updated, and deleted correctly
- Referential integrity maintained between Student and Course tables
- Data retrieved and displayed correctly using SELECT queries
- Role-based access verified using GRANT and REVOKE operations
- Table structure modified successfully and Enrollment table dropped

OUTPUT:

Access granted to new role:

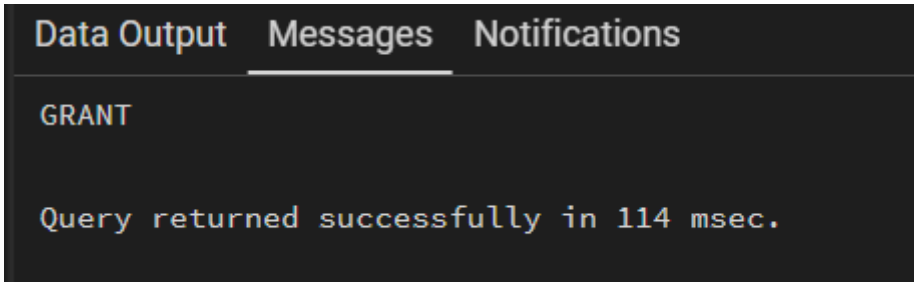
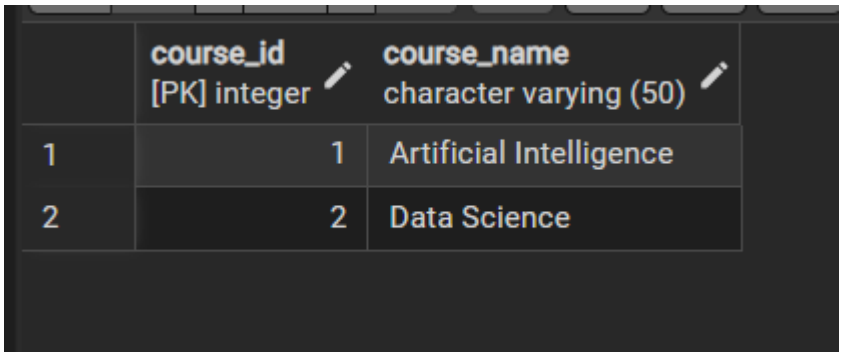


Table Accesed by new role:



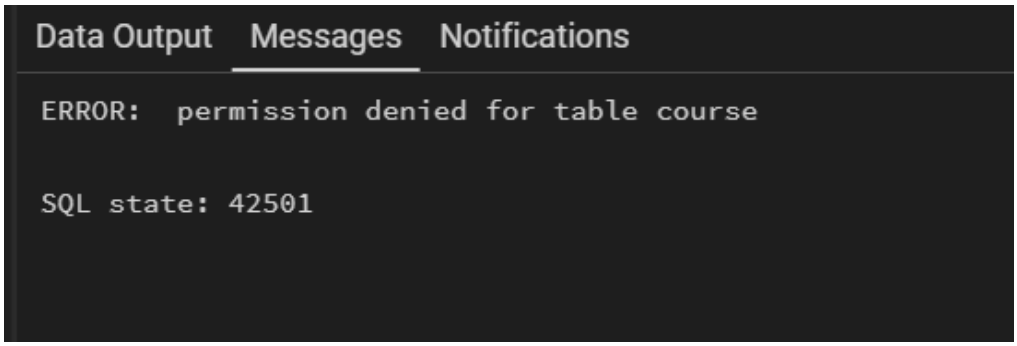
	course_id [PK] integer	course_name character varying (50)
1	1	Artificial Intelligence
2	2	Data Science

Access revoked:

```
REVOKE

Query returned successfully in 74 msec.
```

Accessing after permission revoked:



Learning Outcome:

- Learned to create and manage tables using DDL commands in PostgreSQL.
- Understood the use of constraints to maintain data integrity.
- Performed INSERT, UPDATE, and DELETE operations using DML.
- Implemented role-based access control using DCL commands.
- Provided secure read-only access by granting only SELECT privilege.