

Lab #3: Implementation of SQL Data Manipulation Language (DML) Operations

1. Objectives

- To understand the purpose and usage of Data Manipulation Language (DML) in SQL.
- To insert, update, delete, and retrieve data from database tables using SQL commands.
- To practice querying data using SELECT.

2. Theory

Data Manipulation Language (DML) is the subset of SQL used to work with data stored inside tables. Unlike DDL (which defines structure), DML changes or retrieves the actual records. DML statements are often used together with conditions (WHERE) and may be controlled using transactions.

Key DML Commands:

- INSERT INTO – Adds new rows to a table.
- UPDATE – Modifies existing rows in a table.
- DELETE – Removes rows from a table.
- SELECT – Retrieves rows from a table.

3. Problem Description

Use the previously created database “collegedb” and its three tables:

1. student
2. course
3. enrollment

Perform DML operations to populate data, modify records, remove records, and query information required for a college course registration system.

4. Logical Design (Conceptual)

Tables (same as Lab #2):

- student table

- student_id (int)
- name (varchar)
- email (varchar)
- department (varchar)
- phone (varchar, optional if added in Lab #2)
- course table
- course_code (varchar)
- course_name (varchar)
- credit (int)
- enrollment table
- enrollment_id (int)
- student_id (int)
- course_code (varchar)
- sem (varchar)

5. SQL DML Commands

i. Insert sample data

```
INSERT INTO student (student_id, name, email, department, phone) VALUES
(1, 'Anita Karki', 'anita@gces.edu.np', 'Computer', '9800000031'),
(2, 'Ram Thapa', 'ram@gces.edu.np', 'Software', '9800000008'),
(3, 'Bikash Thapa', 'bikash@gces.edu.np', 'Software', '9800000012'),
(4, 'Sita Gurung', 'sita@gces.edu.np', 'Computer', '9800000003');
```

```
INSERT INTO course (course_code, course_name, credit) VALUES
('CMP222', 'Database Management Systems', 3),
```

```
('MA101', 'Engineering Mathematics I', 3),  
( 'EE201', 'Digital Logic', 2);
```

```
INSERT INTO enrollment (enrollment_id, student_id, course_code, sem) VALUES  
(101, 1, 'CMP222', 'Fall'),  
(102, 1, 'MA101', 'Fall'),  
(103, 2, 'EE201', 'Fall'),  
(104, 3, 'CMP222', 'Fall');
```

ii. Select queries:

```
SELECT * FROM student;  
  
-- view Computer department students only  
SELECT student_id, name, email  
FROM student  
WHERE department = 'Computer';  
  
-- list courses with credit >= 3  
SELECT course_code, course_name, credit  
FROM course  
WHERE credit >= 3;
```

iii. select ,group, aggregate queries

```
-- sort students by name (A-Z)  
SELECT * FROM student
```

ORDER BY name ASC;

-- count students per department

```
SELECT department, COUNT(*) AS total_students
FROM student
GROUP BY department;
```

-- average credit of courses

```
SELECT AVG(credit) AS avg_credit
FROM course
```

iv. Update operations

-- update a student's phone number

```
UPDATE student
SET phone = '9800000099'
WHERE student_id = 2;
```

-- change course credit

```
UPDATE course
SET credit = 4
WHERE course_code = 'CMP222';
```

-- update multiple rows (set semester to 'Spring' for CMP222 enrollments)

```
UPDATE enrollment
SET sem = 'Spring'
WHERE course_code = 'CMP222'
```

v. Delete operations

-- delete one enrollment record

```
DELETE FROM enrollment
```

```
WHERE enrollment_id = 103;
```

-- delete all enrollments of a student

```
DELETE FROM enrollment
```

```
WHERE student_id = 1;
```

6. Tasks

I. Insert at least 4 students, 4 courses, and 6 enrollments into respective tables.

II. Write SELECT queries to:

a) Display all students.

b) Display students of a specific department.

c) Display courses with credit ≥ 3 .

e) Count number of students per department.

III. Update:

a) A student's phone number.

b) Credit value of a course.

IV. Delete:

a) One enrollment by enrollment_id.

b) All enrollments of one student.

7. Observation Table

Students should record the following in their lab notebook/report:

I. ER Diagram

II. SQL commands executed for each task.

If you insert duplicate 'student_id' in 'student' table , what will happen? Is there a better way to handle this?

III. Output rows from SELECT queries.

IV. Any errors encountered and how they were fixed.

Task / Command	Output / Result	Remarks

8. Conclusion

In this lab, we practiced core SQL DML operations to manipulate and retrieve data in the collegedb database. We inserted records, queried data using SELECT (with conditions, aggregates), updated existing rows, and deleted rows. These skills are essential for building real-world applications that interact with databases.