Learning Material: Data Manipulation (DML)

Introduction to Data Manipulation

1. What is Data Manipulation?

Data manipulation refers to the process of **adding**, **viewing**, **updating**, and **removing** data stored inside database tables. These operations allow applications—such as enrollment systems, POS systems, inventory apps, and HR systems—to work with real, changing data.

SQL uses **DML (Data Manipulation Language)** commands to perform these tasks.

Main DML Commands

1. **INSERT** — Add new data
2. **SELECT** — Retrieve or view data
3. **UPDATE** — Modify existing data
4. **DELETE** — Remove data

These operations help maintain accurate and updated records in database systems.

2. INSERT: Adding Records

Syntax

```
INSERT INTO table_name (column1, column2)
VALUES (value1, value2);
```

Example

```
INSERT INTO students (id, firstname, course)
VALUES (101, 'Anna', 'BSIT');
```

Notes:

- The order of values must match the column order.

- Multiple rows can be inserted in one command.

SELECT and UPDATE

3. SELECT: Retrieving Data

The `SELECT` command is used to **view** data in tables.

Basic Syntax

```
SELECT column1, column2 FROM table_name;
```

Examples

Get all records:

```
SELECT * FROM students;
```

Filter data:

```
SELECT firstname, course
FROM students
WHERE course = 'BSIT';
```

Sort results:

```
SELECT * FROM students ORDER BY lastname ASC;
```

4. UPDATE: Modifying Records

Syntax

```
UPDATE table_name
SET column = new_value
WHERE condition;
```

Example

```
UPDATE students
SET course = 'BSCS'
WHERE id = 101;
```

Important Reminder:
Never update without a **WHERE** clause.

```
UPDATE students SET course = 'BSCS';
```

This changes **ALL** rows, which may damage the data.

DELETE, Conditions, Practice

5. DELETE: Removing Records

Syntax

```
DELETE FROM table_name WHERE condition;
```

Example

```
DELETE FROM students
WHERE id = 101;
```

⚠ Without WHERE, this happens:

```
DELETE FROM students;
```

— deletes **every record** in the table.

---

6. Using Conditions in Data Manipulation

Conditions help target specific records.

Common WHERE Operators

- `=` equal
- `>` `<` greater/less
- `<>` or `!=` not equal
- `BETWEEN` — range
- `IN` — list of values
- `LIKE` — pattern matching

Example Using LIKE

```
SELECT * FROM students
WHERE firstname LIKE 'A%';
```

---

7. Quick Real-World Applications

Enrollment System

- INSERT new student
- UPDATE course/section
- SELECT student list
- DELETE dropped students

Inventory System

- INSERT new stock
- UPDATE quantity
- SELECT items low in stock
- DELETE expired items