**📘 DML Statements in MySQL**

**🔹 What is DML?**

* **DML (Data Manipulation Language)** is a category of SQL commands used to **manipulate data stored inside tables**.
* While **DDL** defines the structure (tables, databases), **DML** works on the **actual records (rows)** inside those tables.

**🔹 Why are DML Statements Used?**

1. To **insert new data** into tables.
2. To **update existing records** when values change.
3. To **delete unnecessary or incorrect data**.
4. To **control concurrency (LOCK)** when multiple users work on the same data.
5. To **execute stored procedures (CALL)** that manipulate data logically.

👉 In short: **DML lets you add, change, and remove the information inside your database.**

**🔹 Main DML Statements**

**1. INSERT Statement**

* **Definition**: Used to add new rows (records) into a table.
* **Syntax**:

INSERT INTO table\_name (column1, column2, ...)

VALUES (value1, value2, ...);

* **Example**:

INSERT INTO employees (f\_name, last\_name, salary, joining\_date, email, age)

VALUES ('Rohit', 'Kumar', 65000.00, '2022-10-12', 'rohit.kumar@example.com', 29);

👉 Adds a new employee record.

**2. UPDATE Statement**

* **Definition**: Modifies existing data in a table.
* **Syntax**:

UPDATE table\_name

SET column1 = value1, column2 = value2

WHERE condition;

* **Example**:

UPDATE employees

SET salary = 70000.00, age = 30

WHERE emp\_id = 1;

👉 Updates salary and age of employee with ID 1.  
⚠️ Without WHERE, it updates **all rows**.

**3. DELETE Statement**

* **Definition**: Removes data (rows) from a table.
* **Syntax**:

DELETE FROM table\_name WHERE condition;

* **Example**:

DELETE FROM employees

WHERE emp\_id = 3;

👉 Deletes the employee with ID 3.  
⚠️ Without WHERE, it deletes **all rows**.

**4. LOCK Statement**

* **Definition**: Used to control concurrent access by locking tables.
* **Types**:
  + **READ LOCK** → Others can read, but not write.
  + **WRITE LOCK** → Others cannot read or write until unlocked.
* **Syntax**:

LOCK TABLES table\_name READ;

LOCK TABLES table\_name WRITE;

-- After work is done

UNLOCK TABLES;

* **Example**:

LOCK TABLES employees WRITE;

UPDATE employees SET salary = salary + 2000;

UNLOCK TABLES;

👉 Prevents other sessions from interfering while updating salaries.

**5. CALL Statement**

* **Definition**: Executes a stored procedure (a saved set of SQL statements).
* **Syntax**:

CALL procedure\_name(parameters);

* **Example**: Suppose we create a procedure to raise salary:

DELIMITER //

CREATE PROCEDURE increase\_salary(IN emp INT, IN amount DECIMAL(10,2))

BEGIN

UPDATE employees

SET salary = salary + amount

WHERE emp\_id = emp;

END //

DELIMITER ;

-- Call the procedure

CALL increase\_salary(2, 5000.00);

👉 Increases salary of employee with ID 2 by 5000.

**✅ Summary**

* **DML (INSERT, UPDATE, DELETE, LOCK, CALL)** works on **data inside tables**.
* **INSERT** → Add new rows.
* **UPDATE** → Modify existing rows.
* **DELETE** → Remove rows.
* **LOCK** → Prevent conflicts in multi-user environments.
* **CALL** → Execute stored procedures for reusable logic.