**📝 25 Practice Questions on DML in MySQL**

**🔹 Section 1: INSERT Statements**

1. Insert one new employee into the employees table with your own details.
2. Insert 3 new employees in a single query.
3. Insert a student record into a students table with values for name, age, grade, and admission\_date.
4. Insert a department into departments with dept\_name = "Human Resources".
5. Insert a row into employees without providing salary (check how NULL/default values behave).

**🔹 Section 2: UPDATE Statements**

1. Update the salary of employee with emp\_id = 2 to 90000.
2. Update the email of an employee with emp\_id = 5.
3. Increase salary of all employees by 10%.
4. Change the dept\_name of departments where dept\_id = 3 to "Finance".
5. Update multiple fields (salary and age) for employee emp\_id = 4.

**🔹 Section 3: DELETE Statements**

1. Delete an employee whose emp\_id = 6.
2. Delete all employees with salary < 50000.
3. Delete all students from students whose age > 25.
4. Remove all rows from departments table but keep the table structure (compare DELETE vs TRUNCATE).
5. Delete all employees who joined before 2020-01-01.

**🔹 Section 4: LOCK Statements**

1. Apply a READ LOCK on the employees table and then try selecting data.
2. Apply a WRITE LOCK on the employees table and attempt an UPDATE.
3. Lock the departments table for write operations, insert a new department, and then unlock it.
4. Demonstrate what happens if you try to insert into a locked table from another session.
5. Unlock all tables using UNLOCK TABLES.

**🔹 Section 5: CALL (Stored Procedures)**

1. Create a procedure add\_department that inserts a new department into departments and call it.
2. Create a procedure get\_employee\_by\_id that selects details of an employee given their emp\_id and call it.
3. Create a procedure increase\_salary that increases salary of all employees by 5000 and call it.
4. Create a procedure delete\_student that deletes a student from students table using student\_id and call it.
5. Create a procedure update\_email that takes emp\_id and new email as parameters and updates the record.

✅ **How to practice effectively**:

* Use the **same tables** (employees, departments, students) so your practice stays consistent.
* Start with **INSERT → UPDATE → DELETE** before moving to **LOCK & CALL** (since procedures need more setup).
* Try running queries **with and without WHERE clauses** to see the difference.