**SQL: Data Manipulation Commands (DML)**

**Introduction to Data Manipulation Commands (DML)**  
Data Manipulation Commands (DML) are used to retrieve, insert, update, and delete data in a database. These commands allow users to manipulate the data stored in tables.

DML operations are essential for interacting with the data in a relational database system. Unlike Data Definition Commands (DDC), which deal with the structure of the database, DML operations focus on the content of the tables.

**1. Adding Rows (INSERT)**

The INSERT command is used to add new rows (records) into a table.

**Basic Syntax:**

sql

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

VALUES (value1, value2, ...);

* The column names must correspond to the order of the values being inserted.
* If inserting into all columns, you can omit the column names in the INSERT statement.

**Example:**

sql

INSERT INTO Employees (EmployeeID, FirstName, LastName, Salary)

VALUES (1, 'John', 'Doe', 55000);

**Inserting Multiple Rows:**

You can insert multiple rows in a single INSERT statement.

sql

INSERT INTO Employees (EmployeeID, FirstName, LastName, Salary)

VALUES

(2, 'Jane', 'Smith', 60000),

(3, 'Alex', 'Johnson', 65000);

**INSERT with Subquery:**

You can insert data into a table using the result of a subquery.

sql

INSERT INTO Employees (EmployeeID, FirstName, LastName, Salary)

SELECT EmployeeID, FirstName, LastName, Salary

FROM TempEmployees

WHERE Salary > 40000;

**2. Updating Rows (UPDATE)**

The UPDATE command is used to modify the data in one or more columns of existing rows in a table.

**Basic Syntax:**

sql

UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

* The WHERE clause is essential to specify which rows to update. If omitted, all rows in the table will be updated.

**Example:**

sql

UPDATE Employees

SET Salary = 60000

WHERE EmployeeID = 1;

* This updates the salary of the employee with EmployeeID 1 to 60000.

**UPDATE with Subquery:**

You can update rows using the result of a subquery.

sql

UPDATE Employees

SET Salary = (SELECT AVG(Salary) FROM Employees)

WHERE DepartmentID = 1;

* This updates the salary of employees in Department 1 to the average salary of all employees.

**3. Deleting Rows (DELETE)**

The DELETE command is used to remove rows from a table.

**Basic Syntax:**

sql

DELETE FROM table\_name

WHERE condition;

* The WHERE clause specifies which rows to delete. Omitting it will delete all rows in the table.

**Example:**

sql

DELETE FROM Employees

WHERE EmployeeID = 2;

* This deletes the row with EmployeeID 2 from the Employees table.

**DELETE with Subquery:**

You can delete rows using the result of a subquery.

sql

DELETE FROM Employees

WHERE EmployeeID IN (SELECT EmployeeID FROM Employees WHERE Salary < 40000);

* This deletes employees whose salary is less than 40000.

**4. Inserting, Deleting, and Updating Rows with Subqueries**

Subqueries can be used in INSERT, UPDATE, and DELETE statements to provide data dynamically based on other queries.

**a. INSERT with Subquery:**

sql

INSERT INTO Employees (EmployeeID, FirstName, LastName, Salary)

SELECT EmployeeID, FirstName, LastName, Salary

FROM TempEmployees

WHERE Salary > 40000;

**b. UPDATE with Subquery:**

sql

UPDATE Employees

SET Salary = (SELECT AVG(Salary) FROM Employees WHERE DepartmentID = 1)

WHERE DepartmentID = 1;

**c. DELETE with Subquery:**

sql

DELETE FROM Employees

WHERE EmployeeID IN (SELECT EmployeeID FROM Employees WHERE DepartmentID = 2);

* The subquery here selects the EmployeeID values from employees who belong to Department 2, and the DELETE statement removes those rows from the Employees table.

**5. COMMIT, ROLLBACK, and SAVEPOINT**

These commands are used for transaction control, allowing users to manage database changes and ensure data integrity.

**a. COMMIT**

* The COMMIT command is used to permanently save all the changes made during the current transaction.
* Once a transaction is committed, the changes are final and cannot be undone.

sql

COMMIT;

* **Example**:

sql

UPDATE Employees

SET Salary = 70000

WHERE EmployeeID = 3;

COMMIT;

**b. ROLLBACK**

* The ROLLBACK command is used to undo all changes made in the current transaction.
* It restores the database to its state before the transaction began.

sql

ROLLBACK;

* **Example**:

sql

UPDATE Employees

SET Salary = 75000

WHERE EmployeeID = 4;

ROLLBACK; -- This will undo the update, leaving the salary unchanged.

**c. SAVEPOINT**

* A SAVEPOINT is used to set a point within a transaction to which you can later roll back.
* It allows partial rollback, rather than undoing the entire transaction.

sql

SAVEPOINT savepoint\_name;

* **Example**:

sql

UPDATE Employees

SET Salary = 80000

WHERE EmployeeID = 5;

SAVEPOINT after\_salary\_update; -- Mark the point after salary update

UPDATE Employees

SET Salary = 85000

WHERE EmployeeID = 6;

ROLLBACK TO SAVEPOINT after\_salary\_update; -- Roll back to the state after salary update

* In this example, the ROLLBACK TO SAVEPOINT command undoes the update on EmployeeID = 6 but keeps the changes made to EmployeeID = 5.

**6. Best Practices and Considerations**

* **Transactions**: Always use COMMIT to make sure your changes are saved permanently. If an error occurs, use ROLLBACK to restore the database to its previous state.
* **WHERE Clause**: Always include a WHERE clause when updating or deleting data to avoid modifying or deleting all rows in the table.
* **Subqueries**: When using subqueries, ensure they are efficient, as poorly optimized subqueries can impact performance.
* **Atomicity**: Ensure that DML operations are atomic. This means either all operations succeed (and are committed) or none of them do (and the changes are rolled back).

**7. Summary**

* **INSERT**: Adds new rows into a table. You can insert single or multiple rows, and use subqueries to insert data dynamically.
* **UPDATE**: Modifies existing data in one or more rows. Subqueries can be used to calculate values dynamically for updates.
* **DELETE**: Removes rows from a table. You can delete based on conditions or using subqueries.
* **Subqueries**: Subqueries can be used with INSERT, UPDATE, and DELETE commands to provide dynamic data and conditions.
* **Transaction Control**: COMMIT, ROLLBACK, and SAVEPOINT are used to manage changes in a transaction, providing control over data integrity and recovery.

DML commands are crucial for interacting with the data within a database, providing mechanisms for inserting, updating, deleting, and controlling changes to ensure data accuracy and integrity.