**PROMT**- ***what is update in postgresql , provide complete explanation about why it is use, how it work, common clauses and also provide examples with query***

Here's a complete explanation of the **UPDATE statement in PostgreSQL**, which is used to modify existing records in a table:

**What is UPDATE in PostgreSQL?**

The UPDATE statement in PostgreSQL is used to **modify existing records** in a table. It changes the values of one or more columns for one or multiple rows.

**Why is UPDATE Used?**

* To correct incorrect data (e.g., typo in name, wrong amount)
* To reflect changes (e.g., updated address, salary revision)
* To implement business logic (e.g., deactivate an account)
* To bulk modify rows based on conditions

**How UPDATE Works:**

1. PostgreSQL searches for rows that match the condition in the WHERE clause.
2. It applies the new values using the SET clause.
3. If no WHERE clause is used, **all rows in the table are updated**.
4. You can also return affected rows using the RETURNING clause.

**Basic Syntax:**

UPDATE table\_name

SET column1 = value1,

column2 = value2, ...

WHERE condition;

**Common Clauses in UPDATE:**

| **Clause** | **Purpose** |
| --- | --- |
| SET | Specifies the column(s) and new value(s) |
| WHERE | Filters rows to update (highly recommended to avoid updating all rows) |
| FROM | Used when updating with data from another table |
| RETURNING | Returns updated rows, useful for verification |

**💡 Examples:**

**1. Update a Single Column in One Row**

UPDATE employees

SET salary = 50000

WHERE employee\_id = 101;

Changes the salary of employee with ID 101 to 50000.

**2. Update Multiple Columns**

UPDATE employees

SET salary = 60000,

department = 'Finance'

WHERE employee\_id = 102;

Updates salary and department for employee 102.

**3. Update All Rows (⚠ Dangerous)**

UPDATE employees

SET status = 'Active';

Sets status to 'Active' for **all employees**.

**4. Using RETURNING to View Changes**

UPDATE employees

SET bonus = 1000

WHERE department = 'Sales'

RETURNING employee\_id, bonus;

Shows which employees received the bonus.

**5. Update Using Data from Another Table (FROM Clause)**

Suppose you have:

* employees(employee\_id, salary)
* salary\_updates(employee\_id, new\_salary)

UPDATE employees

SET salary = su.new\_salary

FROM salary\_updates su

WHERE employees.employee\_id = su.employee\_id;

Updates employee salaries based on salary\_updates table.

**Best Practices:**

* Always use WHERE clause unless you want to update all records.
* Test with SELECT first to make sure your condition is correct.
* Use RETURNING to verify the changes.
* Consider transactions if updating critical data.

**Summary:**

| **Feature** | **Description** |
| --- | --- |
| Statement | UPDATE |
| Purpose | Modify existing data in a table |
| Important clauses | SET, WHERE, RETURNING, FROM |
| Usage scenarios | Fixing data, applying changes, business logic |