## Prompt: Tell Me In Brief About UPDATE Command In Postgresql, What It Do, How It Do, Where We Use It And What Is Structure Of It With 2 Examples, Explain Me Like A Teacher And I Am New In This Field Explain Step By Step In Ascending Order Of Difficulty Level

The **UPDATE** command in PostgreSQL is used to modify existing rows of data in a table. If INSERT adds new furniture, UPDATE is for changing something about it—like repainting a chair or changing its price tag. 🏷️

### What it does and Where we use it

Imagine you have a spreadsheet of your products. The UPDATE command is the action of finding a specific row and changing the value in one or more of its cells.

You use UPDATE whenever you need to **change data that already exists** in your database. This happens all the time:

* A user changes their email address.
* You need to correct a typo in a product's name.
* You increase the price of an item or put it on sale.

The basic structure is to specify the **table**, the **changes** you want to make using SET, and—most importantly—**which rows** to change using WHERE.

UPDATE table\_name SET column = new\_value WHERE condition;

**A Very Important Warning!** The WHERE clause is critical. If you forget it, the UPDATE command will change **every single row** in the table! Always double-check your WHERE clause before running an UPDATE.

Let's use our products table, which now has a few items in it, for our examples.

### Level 1: Updating a Single, Specific Row

This is the most common use case. You want to change a value in one specific row, and you know how to identify it uniquely (e.g., by its ID or name).

#### Structure

You use the WHERE clause to pinpoint the exact row you want to modify.

UPDATE table\_name  
SET column\_to\_change = new\_value  
WHERE identifying\_column = specific\_value;

#### Example

The 'Gaming Mouse' we added is now on sale. Let's change its price from 75.50 to 69.99.

1. **The Command**:  
     
   UPDATE products  
   SET price = 69.99  
   WHERE product\_name = 'Gaming Mouse';

1. **What Happens?**:
   * UPDATE products: We tell PostgreSQL we want to modify the products table.
   * SET price = 69.99: This is the change we want to make. We're setting the price column to a new value.
   * WHERE product\_name = 'Gaming Mouse': This is our safety check! The command will find the *one* row where the product\_name is 'Gaming Mouse' and apply the change only to that row.

### Level 2: Updating Multiple Rows at Once

Sometimes you need to apply the same change to a group of rows that share a common characteristic.

#### Structure

The WHERE clause is used to select a group of rows instead of just one. You can also make changes based on a column's existing value.

UPDATE table\_name  
SET column\_to\_change = new\_value\_or\_calculation  
WHERE condition\_that\_matches\_multiple\_rows;

#### Example

Let's apply a 10% discount to all products that cost more than $100.

1. **The Command**:  
     
   UPDATE products  
   SET price = price \* 0.90  
   WHERE price > 100;
2. **What Happens?**:
   * UPDATE products: Again, we target the products table.
   * WHERE price > 100: The command first finds all rows that satisfy this condition. In our list, this would be the 'Mechanical Keyboard' (priced at 120.00).
   * SET price = price \* 0.90: This is the clever part. For each row that the WHERE clause found, it takes the **current value** of the price column, multiplies it by 0.90 (giving a 10% discount), and updates the column with the result. The keyboard's new price would become 108.00.

This is a powerful way to make bulk changes safely and efficiently without having to update each item one by one.