**Prompt: Explain SQL UPDATE statement in detail with syntax and 2 step by step examples**.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The SQL **UPDATE** statement is used to modify existing records in a table. It lets you change the values of one or more columns for a specific set of rows.

**Syntax**

The basic syntax for the UPDATE statement is:

SQL

UPDATE table\_name

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

WHERE condition;

* **UPDATE table\_name**: This specifies the table you want to modify.
* **SET column1 = value1, ...**: This clause indicates which columns you want to change and the new values you want to assign to them. You can update multiple columns at once by separating them with a comma.
* **WHERE condition**: This is a crucial part of the statement. It filters the rows and specifies which ones to update. If you **omit** the WHERE clause, the UPDATE statement will modify **all** rows in the table, which is usually not what you want and can be very dangerous.

**Example 1: Updating a Single Record**

Let's assume you have a users table with the following data:

| user\_id | username | email | status |
| --- | --- | --- | --- |
| 1 | johndoe | john.doe@example.com | active |
| 2 | janedoe | jane.doe@example.com | inactive |

You want to update the email address for johndoe.

**Step 1: Write the UPDATE statement.**

You need to specify the table, the column to change, the new value, and a condition to identify the specific row. The condition will be WHERE user\_id = 1.

SQL

UPDATE users

SET email = 'john.doe.new@example.com'

WHERE user\_id = 1;

**Step 2: Run the query and verify.**

After running the command, the table will be updated. You can verify this by running a SELECT query.

SQL

SELECT \* FROM users;

The result will show the updated email for johndoe:

| user\_id | username | email | status |
| --- | --- | --- | --- |
| 1 | johndoe | john.doe.new@example.com | active |
| 2 | janedoe | jane.doe@example.com | inactive |

**Example 2: Updating Multiple Records**

Now, let's say you have a products table and you want to increase the price of all products that are currently 'on sale'.

| product\_id | product\_name | price | on\_sale |
| --- | --- | --- | --- |
| 101 | Laptop | 1200.00 | true |
| 102 | Monitor | 250.00 | false |
| 103 | Keyboard | 75.00 | true |

You want to increase the price of all products where on\_sale is true by 10%.

**Step 1: Construct the UPDATE statement.**

The SET clause can use an expression based on the current value of a column. We'll set the new price to price \* 1.10. The condition will be WHERE on\_sale = true.

SQL

UPDATE products

SET price = price \* 1.10

WHERE on\_sale = true;

**Step 2: Run the query and check the results.**

Running the UPDATE statement will modify both the Laptop and Keyboard records.

SQL

SELECT \* FROM products;

The updated table will look like this:

| product\_id | product\_name | price | on\_sale |
| --- | --- | --- | --- |
| 101 | Laptop | 1320.00 | true |
| 102 | Monitor | 250.00 | false |
| 103 | Keyboard | 82.50 | true |