This is where **most beginners get confused**, but I'll make it **super simple** for you.

NOT Operator — When to Use Which Keyword

You actually use **NOT** with only 5 types of conditions: This covers **100% of all NOT uses in SQL.**

✔ 1. NOT =

where salary <> 50000
-- same as
where NOT (salary = 50000)
Use when you want **anything except** one value.

✔ 2. NOT IN

where city NOT IN ('Delhi', 'Mumbai')
Use when you want to **exclude multiple values**.

✔ 3. NOT BETWEEN

where age NOT BETWEEN 25 AND 35
Use when you want values **outside a range.**

✔ 4. NOT LIKE

where name NOT LIKE '%a%'
Use when you want to **exclude patterns**.

✔ 5. IS NOT NULL

Where email IS NOT NULL
Use when you want to **remove empty or missing data**.

🎯 When to use which?

Use Case	Best NOT Condition	Example
Excluding **one value**	!= or <>	salary != 50000
Excluding **multiple values**	NOT IN	city NOT IN (...)
Excluding a **range**	NOT BETWEEN	age NOT BETWEEN 20 AND 30
Excluding a **pattern**	NOT LIKE	name NOT LIKE '%e%'
Excluding **null values**	IS NOT NULL	email IS NOT NULL

💡 SUPER EASY MEMORY TRICK

👉 **If it's one value → use** `!=`
👉 **If it's a list → use** `NOT IN`
👉 **If it's a range → use** `NOT BETWEEN`
👉 **If it's a pattern → use** `NOT LIKE`
👉 **If it's empty check → use** `IS NOT NULL`

Important: [!= or <>] instead of this, if we will use only NOT every time - so anything is goes wrong on syntax?
Answer: Using **NOT** is correct SQL — BUT it **cannot replace** != or <> in all situations.

✔ Use <> or != for **simple NOT EQUAL**
✔ Use **NOT** only when reversing logical conditions.

🔄 What is "Reversing Logical Condition"?

Reversing logical condition = opposite output of an existing condition.

It means:
👉 You take a TRUE condition
👉 And convert it into FALSE
OR
👉 You take a FALSE condition
👉 And convert it into TRUE
You do this using **NOT**.

Example 1 — Reversing a simple condition

Condition:
`department = 'IT'`
This returns TRUE for IT employees.
Now apply **NOT** → reverse it:
`NOT (department = 'IT')`
This now returns TRUE for all departments **except IT**.
So "reversing" means:

- TRUE becomes FALSE
- FALSE becomes TRUE

Example 2 — NOT LIKE

Normal condition:

```
name LIKE 'A%'
```

Meaning: names starting with A.

Reverse it:

```
name NOT LIKE 'A%'
```

Meaning: names NOT starting with A.