1. First thing to understand:

LIKE = “pattern must match”
NOT LIKE = “pattern must NOT match”
Very simple.
✔ LIKE (means YES, allow this pattern)
Example:
WHERE name LIKE 'A%'
→ Name must **start with A**.
✔ NOT LIKE (means NO, reject this pattern)
Example:
WHERE name NOT LIKE '%1%'
→ Name must **NOT contain 1**.

---

2. The MOST CONFUSING PART: What is [^] ?

Many people misunderstand this, so let’s make it SUPER SIMPLE.

📌 [A-Z] → allow these characters

Meaning: inside brackets = allowed characters
Examples:

- [A-Z] → uppercase letters
- [a-z] → lowercase
- [0-9] → digits
- [A-Za-z] → all letters

📌 [^A-Z] → NOT these characters

The **^ symbol inside brackets means “NOT these characters”**.
So:
[A-Z] = uppercase letters
[^A-Z] = anything except uppercase letters
(numbers, space, special chars, lowercase letters, etc.)
Simple!

First: What does [A-Z] normally mean?

Letters between A and Z
OK.

❓ What does [^A-Z] mean inside LIKE?

It means:

👉 **"First character must be *NOT* an uppercase letter"**
Not “reject”, not “ban” —
it literally means:
The first character is from the set “anything except A–Z”.

---

3. The MASTER RULE (very easy):

Use NOT LIKE + [^] when you want ONLY allowed characters.

Example:
"Name should contain only letters (A–Z, a–z)"

Correct:

```
WHERE name NOT LIKE '%[^A-Za-z]%'
```

Why?

- `[^A-Za-z]` means “any character that is NOT a letter”
- `NOT LIKE '%[^A-Za-z]%'` means
  → REJECT names that contain even **one** non-letter
  → So only letters remain

One query solves everything.

Why you got confused?

Because:

`[^A-Z]` means "not uppercase"

`NOT LIKE '[^A-Z]%'` means "NOT (not uppercase)"

→ which becomes

👉 **"uppercase"**

So it’s double negative.

Why does `NOT LIKE '[^A-Z]%'` become uppercase?

NOT means negative, and ^ also means negative —
so how does double negative become POSITIVE (uppercase)?”**
You were confused because:

- `[^A-Z]` means **NOT uppercase characters**
- `NOT LIKE` means **NOT this pattern**
  And you asked:

****“Why does NOT (NOT uppercase) = uppercase?**
Shouldn’t NOT always give a negative result?”**
So your original question is about:

🔍 ****Understanding why `NOT LIKE '[^A-Z]%'` results in uppercase names,**
even though both NOT and ^ mean negative.**

Your doubt is **100% logical**, and the confusion comes from mixing **two different "NOT" meanings**:

- **NOT inside the brackets** → [^A-Z]
- **NOT before LIKE** → NOT LIKE

Both are "NOT", but they work in **two different places** and for **two different jobs**.

2) Why double negative becomes positive — step by step

Write the predicate P:

- P = "first character is NOT uppercase"

Now evaluate NOT P:

- NOT P = "NOT (first character is NOT uppercase)"

In plain language NOT (not X) = X. So:

- NOT (first char is NOT uppercase) = "first char IS uppercase".

This is just **double negative** in logic.

3) Truth table (very small)

Let X = "first char is uppercase".

First char	X = uppercase?	P = NOT X? (LIKE '[^A-Z]%')	NOT P? (NOT LIKE '[^A-Z]%')
A	true	false (not uppercase? no)	true (not false = true)
a	false	true	false
1	false	true	false
@	false	true	false
(space)	false	true	false

1. What is a *character*?

A **character** is **any single symbol** you can type.

Characters include:

- Letters → A, b, C
- Digits → 0, 1, 2
- Symbols → @, #, %, -
- Space → ( )
- Punctuation → ! , . ?
- Emojis ☺
- Anything you type on the keyboard

✔ **Every letter is a character.**

⍰ But not every character is a letter.

🔍 2. What is a *letter*?

A **letter** is only the **A–Z alphabet** (English letters).

Letters:

- Uppercase → A, B, C, … Z
- Lowercase → a, b, c, … z

Letters do NOT include:

- Digits → 0, 1, 2
- Special characters → @, #, %, $
- Space → " "
- Emojis ☺

Why NOT use LIKE with [A-Z] for only-uppercase?

Example **WRONG**:

```
WHERE dept LIKE '%[A-Z]%'
```

This only checks:

- Does dept contain **at least 1 uppercase letter?**

So these will ALSO pass:

- HR1
- A!
- A-12@
- Aaaaa12###

Because they have **one uppercase letter**.

So LIKE + [A-Z] is NEVER the right solution when you want **ONLY uppercase**.

⊘ When NOT to use ^ ?

Very simple rule:

🔍 Do NOT use ^ when you only want to check presence

Example:

```
WHERE name LIKE '%[0-9]%'   -- check if digit exists
```

This is correct.

You don't use ^ here.

🎯 When TO use ^ ?

Also simple:

✔ Use [^] when you want to ban certain characters

Examples:

1. Only letters
WHERE name NOT LIKE '%[^A-Za-z]%'
2. Only lowercase
WHERE name NOT LIKE '%[^a-z]%'
3. Only uppercase
WHERE name NOT LIKE '%[^A-Z]%'
4. Only letters + hyphen
WHERE name NOT LIKE '%[^A-Za-z-]%'

🔥 SUPER SIMPLE ANALOGY (for non-technical students)

Think of:

[A-Z] = "Allowed people can enter the party."

[^A-Z] = "These people CANNOT enter."

Now:

LIKE '%[A-Z]%'

→ "At least one allowed person entered the party."

NOT LIKE '%[^A-Z]%'

→ "Reject the party if ANY bad person entered."

That's all.

🎯 Quick summary in 10 seconds

Pattern	Meaning
[A-Z]	Only uppercase letters
[^A-Z]	Anything except uppercase letters
LIKE '%[A-Z]%'	Check if uppercase exists
NOT LIKE '%[^A-Z]%'	Ensure ONLY uppercase letters present

Shortcut: To allow ONLY specific characters → ALWAYS use NOT LIKE + [^allowed].