

# Use of Wild Cards in SQL

Wildcards in SQL (`%` and `\_`) are extremely powerful when performing pattern matching in strings using the `LIKE` operator. They allow for flexible querying of data where the exact match is not known, making them useful in various industries to extract insights from large datasets. Below is an explanation of the two most common wildcards and their practical applications in real-life industries.

## 1. Percent (`%`) Wildcard

The `%` wildcard is used to match any sequence of characters (including zero characters). It can be placed before, after, or in the middle of a pattern.

### Advantages:

1. Enables flexible searches where the full string is unknown or may vary.
2. Useful for matching multiple variations of a word, partial strings, or incomplete information.
3. Supports efficient querying without needing the exact values.

### Use Cases by Industry:

1. Retail: Search for product names containing specific keywords.

- Example: Find all products that contain the word "shoes" in the name, regardless of what comes before or after.

```
SELECT product_name
FROM Products
WHERE product_name LIKE '%shoes%';
```

- Use: This helps to filter out relevant products when running promotions or analyzing sales trends for specific items.

2. Healthcare: Find patient records where part of the diagnosis is known, but the full term is not.

- Example: Search for all patients diagnosed with a condition starting with "diabetes."

```
SELECT patient_id, diagnosis
FROM Patient_Records
WHERE diagnosis LIKE 'diabetes%';
```

- Use: This is particularly useful for analyzing patients with similar conditions when their exact diagnosis may vary slightly.

3. Finance: Search for all transactions made by clients whose names start with a certain letter or substring.

- Example: Find all clients whose names start with "John."

```
SELECT client_name
FROM Transactions
WHERE client_name LIKE 'John%';
```

- Use: This helps customer service and reporting teams to retrieve specific customer transaction data quickly.

4. Real Estate: Look for properties in cities that contain a certain keyword.

- Example: Find properties located in any city containing "Ville."

```
SELECT property_id, city
FROM Properties
WHERE city LIKE '%Ville%';
```

- Use: This helps agents find properties in areas with similar names, even when city names have variations.

## 2. Underscore (`\_`) Wildcard

The `\_` wildcard is used to match a single character. It is ideal when searching for strings with a known structure but one or more unknown characters.

### Advantages:

1. Allows for precise searches where only one character is uncertain.
2. Efficient when dealing with fixed-length codes or IDs with predictable patterns.
3. Minimizes errors by allowing limited variation in search patterns.

### Use Cases by Industry:

1. Logistics: Find shipment tracking numbers that differ only by one character (e.g., check for errors in tracking numbers).

- Example: Search for shipments where the third character could be any value.

```
SELECT tracking_number
FROM Shipments
WHERE tracking_number LIKE 'AB_12345';
```

- Use: This helps identify mistyped tracking numbers or potential duplicates in logistics systems.

2. Education: Search for student IDs where one digit might be uncertain due to input errors.

- Example: Find student IDs where the second character is unknown.

```
SELECT student_id
FROM Students
WHERE student_id LIKE 'S_123';
```

- Use: This allows administrators to fix input errors in student data efficiently.

3. Telecom: Identify customer phone numbers where one digit might be mistyped or unknown.

- Example: Search for all phone numbers that start with '234' and have one unknown digit.

```
SELECT phone_number
FROM Customers
WHERE phone_number LIKE '234_567890';
```

- Use: Helps telecom providers quickly identify and resolve issues with incorrect or incomplete phone numbers.

4. Manufacturing: Search for product codes where one specific character might vary (e.g., product model numbers).

- Example: Retrieve all products with a model number where the last character is uncertain.

```
SELECT product_code  
FROM Inventory  
WHERE product_code LIKE 'MFG_1234_';
```

- Use: This is particularly useful for inventory management, where different versions of a product might have very similar codes except for one character.

### **Advantages of Using Wildcards:**

1. **Efficient Searching:** Wildcards allow for searching data even when the exact match is not known, making them ideal for incomplete or messy data.
2. **Flexibility:** They enable pattern matching, which helps in querying large datasets with diverse information.
3. **Error Correction:** They can be used to find errors in data entry, such as mistyped names, product codes, or customer IDs.
4. **Versatility:** Wildcards can be applied across industries to perform searches, filter results, and extract valuable insights without requiring exact matches.

### **Conclusion**

Wildcards in SQL (`%` and `\_`) are critical tools for data analysts, allowing them to perform flexible and efficient searches. Whether it's finding similar products in retail, identifying errors in logistics, or searching patient records in healthcare, wildcards help industries make sense of large datasets even when some data is missing or incomplete. They offer flexibility, precision, and efficiency in querying real-world databases.