📘 Module 29: Pattern (String) Matching

🔍🧠 What is Pattern Matching?

Pattern matching is used to search, filter, or identify data that **matches a specific text format** using symbols and functions.

It is especially useful when:

- You don't know the full value
- You want to filter by **starting**, **ending**, or **partial text**

🔤 Pattern Matching Using `LIKE` and Wildcards

✅ Remember:

The `LIKE` **operator** is the most basic tool for pattern matching.

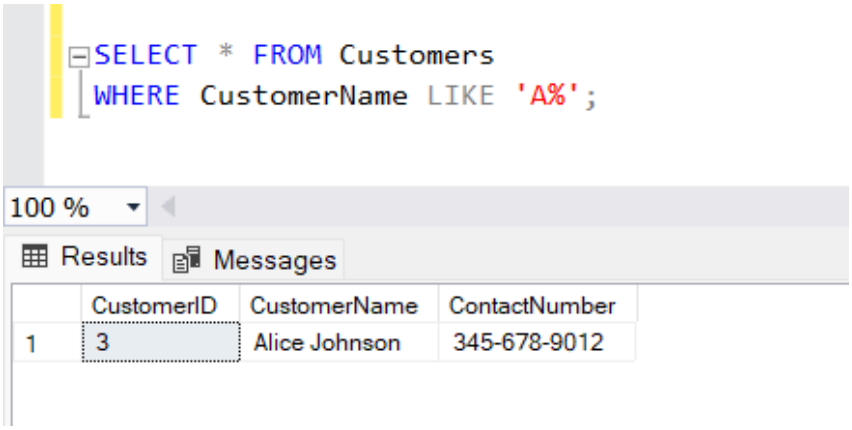
Use **wildcards** like `%`, `_`, and `[]` to define your pattern.

✅ Common Wildcards:

Wildcard	Meaning	Example	Matches
`%`	Any number of characters	`'A%'`	A, Alex, Apple
`_`	Exactly one character	`'J_n'`	Jan, Jon, Jim
`[]`	One character from a set/range	`'[A-C]%'`	Apple, Beta, Cat
`[^]`	Not in the given set	`'[^A-C]%'`	Mango, Zebra

📌 Example:

SELECT * FROM Customers
WHERE CustomerName LIKE 'A%';



🔍 `PATINDEX()` – Pattern Index

✅ Definition:

Returns the **starting position** of a pattern in a string using wildcards (`%`).

✅ Syntax:

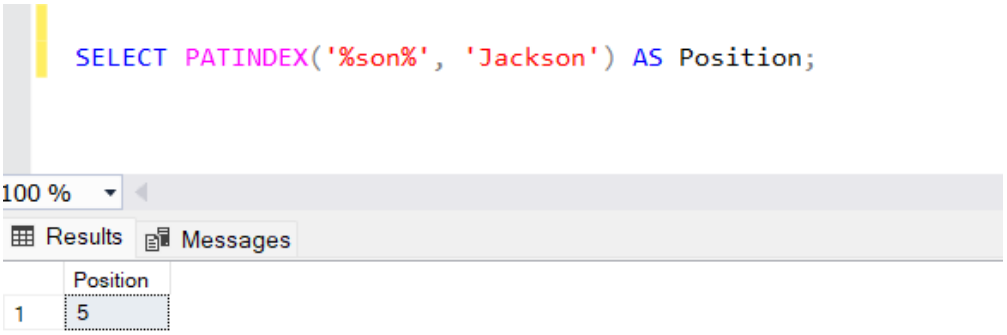
```
PATINDEX('%pattern%', string)
```

📌 Example:

```
SELECT PATINDEX('%son%', 'Jackson') AS Position;
```

✔️ Returns 0 if the pattern is not found.

✔️ Case-insensitive by default in SQL Server



🔎 `CHARINDEX()` – Character Index

✅ Definition:

Finds the **position of a substring** in a string.

Unlike `PATINDEX`, it doesn't support `%` wildcards.

✅ Syntax:

```
CHARINDEX('substring', string)
```

📌 Example:

```
SELECT CHARINDEX('a', 'Tarun') AS Position;
```

✔️ Returns position of first match

✔️ Returns 0 if substring not found

```
SELECT CHARINDEX('a', 'Tarun') AS Position;
```

100 %

Results | Messages

	Position
1	2

↔️ LEFT() and RIGHT() – Extracting Substrings

✅ Definitions:

- `LEFT()` extracts characters from the **start** of a string.
- `RIGHT()` extracts characters from the **end** of a string.

✅ Syntax:

LEFT(string, number_of_characters)
RIGHT(string, number_of_characters)

📌 Example:

```
SELECT
  LEFT('DataScience', 4) AS StartPart,
  RIGHT('DataScience', 7) AS EndPart;
```

✔️ Use to split or format columns like name, IDs, phone numbers, etc.

```
SELECT
    LEFT('DataScience', 4) AS StartPart,
    RIGHT('DataScience', 7) AS EndPart;
```

100 %

Results | Messages

	StartPart	EndPart
1	Data	Science

📝 Key Points / Important Notes

Concept	Tip / Usage
`LIKE`	Use `%`, `_` for flexible search
`PATINDEX()`	Supports wildcard `%`, returns position
`CHARINDEX()`	Exact substring match, no wildcards
`LEFT()/RIGHT()`	Use to trim/focus on part of string
Case-sensitivity	`LIKE` is **case-insensitive** in SQL Server
Combination	Combine functions (e.g., `LEFT(FullName, 1)`)

🧠 Bonus Tips:

- Use `ISNULL()` or `COALESCE()` when working with string functions on nullable fields
- Use `TRIM()` before pattern matching to clean extra spaces
- Combine `LIKE` + `LEFT/RIGHT` + `CHARINDEX()` for powerful custom filters