
Chapter: TEXT OPERATORS (MySQL)

 MySQL with Example, Query, Data & O/O

◆ 1. LIKE Operator

- Use:** Pattern match karta hai kisi string ke sath
-

Table: students

id	name
1	Amit
2	Ramesh
3	Rahul
4	Anjali
5	Aman

Query:

```
SELECT * FROM students WHERE name LIKE 'A%';
```

- Output (O/O):**

id	name
1	Amit
4	Anjali
5	Aman

 Name jo 'A' se start ho rahe hain wo aaye.

◆ 2. NOT LIKE

- Use:** Jo pattern match **nahi** karta wo data hai

❖ Query:

```
SELECT * FROM students WHERE name NOT LIKE 'A%';
```

□ O/O:

id name

2	Ramesh
3	Rahul

◆ 3. INSTR()

- **Use:** Ek word ke andar dusra word kis position pe hai, ye batata hai
-

❖ Query:

```
SELECT INSTR('Hello World', 'World') AS Position;
```

□ O/O:

Position

7

❖ 'World' start ho raha hai 7th position se.

◆ 4. CONCAT()

- **Use:** Strings ko jodne ke liye use hota hai
-

■ Table: users

id fname lname

1	Raj	Kumar
2	Neha	Singh

❖ Query:

```
SELECT CONCAT(fname, ' ', lname) AS full_name FROM users;
```

□ O/O:

full_name

Raj Kumar

Neha Singh

◆ 5. SUBSTRING()

□ Use: String ka kuch hissa nikalne ke liye

❖ Query:

```
SELECT SUBSTRING('OpenAI Rocks', 1, 6) AS result;
```

□ O/O:

result

OpenAI

◆ 6. REPLACE()

□ Use: String me word ko replace karta hai

❖ Query:

```
SELECT REPLACE('Coding is fun', 'fun', 'awesome') AS updated;
```

□ O/O:

updated

Coding is awesome

◆ 7. TRIM()

- Use:** Extra space hatata hai string ke aage/peeche se
-

❖ Query:

```
SELECT TRIM('    Hello Bhai    ') AS clean;
```

- O/O:**

clean
Hello Bhai

◆ 8. UPPER()

- Use:** String ko CAPITAL me convert karta hai
-

❖ Query:

```
SELECT UPPER('hello world') AS upper_text;
```

- O/O:**

upper_text
HELLO WORLD

◆ 9. LOWER()

- Use:** String ko small letters me convert karta hai
-

❖ Query:

```
SELECT LOWER('HELLO WORLD') AS lower_text;
```

- O/O:**

lower_text
hello world



❖ Database create

1. ➡ Table create
 2. ➕ Data insert
 3. ↗ Text Operators ke Queries
 4. ↙ Output (O/O)
-

❑ Chapter: TEXT OPERATORS in MySQL

(With Database, Insert, Queries, Output —)

◆ Step 1: DATABASE Creation

```
CREATE DATABASE myschool;
USE myschool;
```

↗ Database myschool create karke use kar liya.

◆ Step 2: TABLE Create

```
CREATE TABLE students (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50)
);
```

↗ Table students ban gaya with id & name.

◆ Step 3: DATA Insert

```
INSERT INTO students (name) VALUES
('Amit'),
('Ramesh'),
('Rahul'),
('Anjali'),
```

```
('Aman'),  
('Ritika'),  
('Mohit'),  
('Ankita'),  
('Rohit');
```

↙ 9 rows insert ho gayi students table me.

◆ Step 4: TEXT OPERATORS Examples

↙ 1. LIKE Operator

□ **Kaam:** Names jo 'A' se start ho rahe hain.

```
SELECT * FROM students WHERE name LIKE 'A%';
```

□ **O/O:**

id name

```
1 Amit  
4 Anjali  
5 Aman  
8 Ankita
```

↙ 2. NOT LIKE

□ **Kaam:** Names jo 'R' se start nahi ho rahe

```
SELECT * FROM students WHERE name NOT LIKE 'R%';
```

□ **O/O:**

id name

```
1 Amit  
4 Anjali  
5 Aman  
7 Mohit  
8 Ankita
```

↙ 3. INSTR()

- Kaam:** Name me 'it' kahaan aa raha hai?

```
SELECT name, INSTR(name, 'it') AS position FROM students;
```

- O/p:**

name position

Amit 3

Mohit 4

Ankita 0

Ritika 3

Rohit 4

 *Jahan o hai, wahan 'it' nahi mila.*

 **4. CONCAT()**

- Kaam:** Naam ke aage "Mr./Ms." lagana

```
SELECT CONCAT('Mr./Ms. ', name) AS full_name FROM students;
```

- O/O:**

full_name

Mr./Ms. Amit

Mr./Ms. Ramesh

Mr./Ms. Rahul

Mr./Ms. Anjali

...

 **5. SUBSTRING()**

- Kaam:** First 3 letters nikalna name me se

```
SELECT name, SUBSTRING(name, 1, 3) AS short_name FROM students;
```

- O/O:**

name short_name

Amit Ami

Ramesh Ram

Anjali Anj

Mohit Moh

✓ 6. REPLACE()

- **Kaam:** Name me 'it' ko 'xx' se replace karna

```
SELECT name, REPLACE(name, 'it', 'xx') AS updated_name FROM students;
```

- **O/O:**

name updated_name

Amit Amxx

Mohit Mohxx

Ritika Rixxka

Rohit Rohxx

✓ 7. TRIM()

- **Kaam:** Spaces hataana (example ke liye string use kiya hai)

```
SELECT TRIM('    Hello Bro    ') AS clean_text;
```

- **O/O:**

clean_text

Hello Bro

✓ 8. UPPER() / LOWER()

```
SELECT name, UPPER(name) AS capital_name, LOWER(name) AS small_name FROM students;
```

- **O/O:**

name capital_name small_name

Amit AMIT amit

Anjali ANJALI anjali

✓ Chapter Done Bhai