

---

## MySQL – STRING FUNCTIONS (Book Format Notes)

---

### □ Step 1: Create Sample Table

```
CREATE TABLE users (
    id INT,
    name VARCHAR(50),
    email VARCHAR(100)
);
```

---

### □ Step 2: Insert Sample Data

```
INSERT INTO users (id, name, email) VALUES
(1, 'Ankit', 'ankit@example.com'),
(2, 'Riya', 'RIYA@EXAMPLE.COM'),
(3, 'Aman', 'aman@sql.com'),
(4, 'Sneha', 'sneha@Example.com'),
(5, 'rohit', 'rohit@data.com');
```

---

### ■ Data in `users` Table:

	<b>id</b>	<b>name</b>	<b>email</b>
1	Ankit	<a href="mailto:ankit@example.com">ankit@example.com</a>	
2	Riya	<a href="mailto:RIYA@EXAMPLE.COM">RIYA@EXAMPLE.COM</a>	
3	Aman	<a href="mailto:aman@sql.com">aman@sql.com</a>	
4	Sneha	<a href="mailto:sneha@Example.com">sneha@Example.com</a>	
5	rohit	<a href="mailto:rohit@data.com">rohit@data.com</a>	

---

### ❖ 1. `LENGTH()` – Length of string

```
SELECT name, LENGTH(name) AS name_length FROM users;
```

- ☞ Name ke characters count karega, space bhi count hoti hai.
- ☞ Har name ka length milega.

↑ Output:

**name name\_length**

Ankit 5

Riya 4

Aman 4

Sneha 5

rohit 5

-----&gt;

**❖ 2. LOWER() – Convert to lowercase**

```
SELECT email, LOWER(email) AS lower_email FROM users;
```

☞ Email ko lowercase me convert karta hai.

☞ Useful in case-insensitive comparison ke liye.

↑ Output:

email	lower_email
<a href="mailto:ankit@example.com">ankit@example.com</a>	<a href="mailto:ankit@example.com">ankit@example.com</a>
<a href="mailto:RIYA@EXAMPLE.COM">RIYA@EXAMPLE.COM</a>	<a href="mailto:riya@example.com">riya@example.com</a>
<a href="mailto:aman@sql.com">aman@sql.com</a>	<a href="mailto:aman@sql.com">aman@sql.com</a>
<a href="mailto:sneha@Example.com">sneha@Example.com</a>	<a href="mailto:sneha@example.com">sneha@example.com</a>
<a href="mailto:rohit@data.com">rohit@data.com</a>	<a href="mailto:rohit@data.com">rohit@data.com</a>

-----&gt;

**❖ 3. UPPER() – Convert to uppercase**

```
SELECT name, UPPER(name) AS upper_name FROM users;
```

☞ Sabhi characters ko capital letters me convert karega.

☞ Mostly formatting ke kaam aata hai.

↑ Output:

**name upper\_name**

Ankit ANKIT

Riya RIYA

Aman AMAN

Sneha SNEHA

**name upper\_name**

rohit ROHIT

----->

---

## ❖ 4. SUBSTRING() – Part of a string

```
SELECT email, SUBSTRING(email, 1, 5) AS short_email FROM users;
```

☞ Email ka pehla 5 characters return karega.

☞ Syntax: SUBSTRING(str, start\_pos, length)

↑ Output:

email	short_email
<a href="mailto:ankit@example.com">ankit@example.com</a>	ankit
<a href="mailto:RIYA@EXAMPLE.COM">RIYA@EXAMPLE.COM</a>	RIYA@
<a href="mailto:aman@sql.com">aman@sql.com</a>	aman@
<a href="mailto:sneha@Example.com">sneha@Example.com</a>	sneha
<a href="mailto:rohit@data.com">rohit@data.com</a>	rohit

----->

---

## ❖ 5. CONCAT() – Join strings

```
SELECT CONCAT(name, ' - ', email) AS info FROM users;
```

☞ Name aur email ko join karta hai – ke sath.

☞ Multiple strings ko combine karne ke liye use hota hai.

↑ Output:

info
Ankit - <a href="mailto:ankit@example.com">ankit@example.com</a>
Riya - <a href="mailto:RIYA@EXAMPLE.COM">RIYA@EXAMPLE.COM</a>
Aman - <a href="mailto:aman@sql.com">aman@sql.com</a>
Sneha - <a href="mailto:sneha@Example.com">sneha@Example.com</a>
rohit - <a href="mailto:rohit@data.com">rohit@data.com</a>

----->

---

## ❖ 6. REPLACE() – Replace part of string

```
SELECT email, REPLACE(email, 'example', 'gmail') AS new_email FROM users;
```

- ☞ example ko gmail se replace karega email column me.
- ☞ Word ya pattern ko change karne ke liye kaam aata hai.

▲ Output:

email	new_email
<u>ankit@example.com</u>	<u>ankit@gmail.com</u>
<u>RIYA@EXAMPLE.COM</u>	<u>RIYA@EXAMPLE.COM</u>
<u>aman@sql.com</u>	<u>aman@sql.com</u>
<u>sneha@Example.com</u>	<u>sneha@Example.com</u>
<u>rohit@data.com</u>	<u>rohit@data.com</u>

⚠ Replace case-sensitive hota hai. Isliye kuch rows unchanged hain.

----->

---

## ❖ 7. TRIM() – Remove spaces

```
SELECT TRIM('    Hello SQL      ') AS trimmed;
```

- ☞ String ke aage/peechhe ke extra spaces hata deta hai.
- ☞ Formatting me useful hota hai.

▲ Output:

```
trimmed  
Hello SQL
```

----->

---

## ❖ 8. INSTR() – Position of substring

```
SELECT email, INSTR(email, '@') AS at_pos FROM users;
```

- ☞ @ symbol kis position pe hai string me, wo batata hai.
- ☞ Agar symbol nahi mila to 0 return karta hai.

↑ Output:

email	at_pos
<a href="mailto:ankit@example.com">ankit@example.com</a>	6
<a href="mailto:RIYA@EXAMPLE.COM">RIYA@EXAMPLE.COM</a>	5
<a href="mailto:aman@sql.com">aman@sql.com</a>	5
<a href="mailto:sneha@Example.com">sneha@Example.com</a>	6
<a href="mailto:rohit@data.com">rohit@data.com</a>	6

----->

---

## ❖ 9. REVERSE() – Reverse string

```
SELECT name, REVERSE(name) AS rev_name FROM users;
```

❖ Name ko ulta (reverse) kar ke dikhata hai.

❖ Fun/logic based query ke kaam aata hai.

↑ Output:

name	rev_name
Ankit	tiknA
Riya	ayiR
Aman	namA
Sneha	ahenS
rohit	tihor

----->

---

## ❖ Summary – Useful String Functions in MySQL

Function	Use
LENGTH()	Count characters in string
LOWER()	Convert to lowercase
UPPER()	Convert to uppercase
SUBSTRING()	Extract part of string
CONCAT()	Join multiple strings
REPLACE()	Replace part of string
TRIM()	Remove leading/trailing spaces
INSTR()	Find position of substring

<b>Function</b>	<b>Use</b>
-----------------	------------

| REVERSE() | Reverse string |

---

chapter complete! 100