

## SQL String/Character Manipulation

টেবিল ডেটা:

Students Table:

	student_id	student_name	email	phone
1		alice johnson	alice@email.com	+880-1711-123456
2		BOB SMITH	bob@EMAIL.COM	+880-1812-234567
3		Charlie Brown	charlie@email.com	+880-1911-345678
4		David Lee	david@email.com	+880-1611-456789

Products Table:

	product_id	product_name	product_code	description
101		Laptop	LAP-2024-001	High performance laptop
102		Mouse	MOU-2024-002	Wireless mouse
103		Keyboard	KEY-2024-003	Mechanical keyboard
104		Monitor	MON-2024-004	24 inch LED monitor

### 1. UPPER() - সব অক্ষর বড় হাতের করা

কোড:

```
SELECT student_name, UPPER(student_name) AS uppercase_name  
FROM students;
```

ব্যাখ্যা:

- UPPER() function সব lowercase অক্ষর uppercase-এ convert করে
- Already uppercase থাকলে সেগুলো unchanged থাকে
- Case-insensitive comparison করতে কাজে লাগে
- WHERE clause-এও ব্যবহার করা যায়

## আউটপুট:

**student\_name uppercase\_name**

alice johnson ALICE JOHNSON

BOB SMITH BOB SMITH

Charlie Brown CHARLIE BROWN

David Lee DAVID LEE

**মনে রাখার কথা:** সব অক্ষর বড় হাতের (UPPERCASE)

---

## 2. LOWER() - সব অক্ষরচোট হাতের করা

### কোড:

```
SELECT student_name, LOWER(student_name) AS lowercase_name  
FROM students;
```

### ব্যাখ্যা:

- LOWER() function সব uppercase অক্ষর lowercase-এ convert করে
- Email address normalize করতে ব্যবহার হয়
- Case-insensitive search করতে কাজে লাগে
- Data consistency maintain করতে সাহায্য করে

## আউটপুট:

**student\_name lowercase\_name**

alice johnson alice johnson

BOB SMITH bob smith

Charlie Brown charlie brown

David Lee david lee

**মনে রাখার কথা:** সব অক্ষর ছোট হাতের (lowercase)

---

### 3. LENGTH() / CHAR\_LENGTH() - অক্ষর সংখ্যা গণনা

**কোড:**

```
SELECT student_name, LENGTH(student_name) AS name_length  
FROM students;
```

**ব্যাখ্যা:**

- LENGTH() string-এর মোট characters count করে
- Space এবং special characters-ও count হয়
- CHAR\_LENGTH() একই কাজ করে ( MySQL-এ )
- Validation check করতে ব্যবহার হয় ( password length check ইত্যাদি )

**আউটপুট:**

**student\_name name\_length**

alice johnson 13

BOB SMITH 9

Charlie Brown 13

David Lee 9

**মনে রাখার কথা:** মোট character সংখ্যা ( space সহ )

---

### 4. CONCAT() - String জোড়া লাগানো

**কোড:**

```
SELECT student_id, CONCAT('Student: ', student_name, ' (ID: ', student_id, ')') AS full_info  
FROM students;
```

**ব্যাখ্যা:**

- CONCAT() একাধিক strings একসাথে জোড়া লাগায়
- যেকোনো সংখ্যক arguments নিতে পারে
- NULL value থাকলে পুরো result NULL হয় ( MySQL-এ )
- Column values-এর সাথে text combine করতে ব্যবহার হয়

**আউটপুট:**

**student\_id full\_info**

- 1        Student: alice johnson (ID: 1)
- 2        Student: BOB SMITH (ID: 2)
- 3        Student: Charlie Brown (ID: 3)
- 4        Student: David Lee (ID: 4)

**মনে রাখার কথা:** একাধিক string একসাথে জোড়া

---

## 5. CONCAT\_WS() - Separator দিয়ে জোড়া লাগানো

**কোড:**

```
SELECT CONCAT_WS(' - ', student_id, student_name, email) AS student_details
FROM students;
```

**ব্যাখ্যা:**

- CONCAT\_WS() মানে CONCAT With Separator
- প্রথম argument separator, বাকিগুলো values
- প্রতিটি value-র মাঝে separator automatically add হয়
- NULL values skip করা হয় ( empty string হয় না )

**আউটপুট:**

### **student\_details**

1 - alice johnson - alice@email.com

2 - BOB SMITH - bob@EMAIL.COM

3 - Charlie Brown - charlie@email.com

4 - David Lee - david@email.com

**মনে রাখার কথা:** Separator সহ জোড়া (প্রথমটা separator)

---

### **6. SUBSTRING() / SUBSTR() - অংশ কেটে নেওয়া**

**কোড:**

```
SELECT product_code,  
       SUBSTRING(product_code, 1, 3) AS category,  
       SUBSTRING(product_code, 5, 4) AS year,  
       SUBSTRING(product_code, 10) AS serial  
FROM products;
```

**ব্যাখ্যা:**

- SUBSTRING(string, start, length) নির্দিষ্ট অংশ কেটে নেয়
- start position 1 থেকে শুরু হয় (0 নয়)
- length না দিলে শেষ পর্যন্ত নেয়
- SUBSTR() হলো shorthand, same কাজ করে

**আউটপুট:**

**product\_code category year serial**

LAP-2024-001 LAP 2024 001

MOU-2024-002 MOU 2024 002

**product\_code category year serial**

KEY-2024-003 KEY 2024 003

MON-2024-004 MON 2024 004

**মনে রাখার কথা:** নির্দিষ্ট position থেকে নির্দিষ্ট length পর্যন্ত

---

## 7. LEFT() - বাম দিক থেকে নেওয়া

**কোড:**

```
SELECT phone, LEFT(phone, 4) AS country_code  
FROM students;
```

**ব্যাখ্যা:**

- LEFT(string, length) বাম দিক থেকে নির্দিষ্ট সংখ্যক characters নেয়
- SUBSTRING(string, 1, length) এর shorthand
- Phone number, postal code ইত্যাদি parse করতে ব্যবহার হয়
- সবসময় শুরু থেকে count করে

**আউটপুট:**

phone	country_code
-------	--------------

+880-1711-123456 +880

+880-1812-234567 +880

+880-1911-345678 +880

+880-1611-456789 +880

**মনে রাখার কথা:** শুরু থেকে নির্দিষ্ট সংখ্যক character

---

## 8. RIGHT() - ডান দিক থেকে নেওয়া

**কোড:**

```
SELECT product_code, RIGHT(product_code, 3) AS serial_number  
FROM products;
```

#### ব্যাখ্যা:

- RIGHT(string, length) ডান দিক থেকে নির্দিষ্ট সংখ্যক characters নেয়
- শেষের অংশ extract করতে সুবিধা
- File extension, serial number ইত্যাদি বের করতে ব্যবহার হয়
- সবসময় শেষ থেকে count করে

#### আউটপুট:

**product\_code serial\_number**

LAP-2024-001 001

MOU-2024-002 002

KEY-2024-003 003

MON-2024-004 004

মনে রাখার কথা: শেষ থেকে নির্দিষ্ট সংখ্যক character

---

## 9. TRIM() - দুই পাশের space সরানো

#### কোড:

```
SELECT CONCAT('[', ' hello world ', ']') AS before_trim,  
       CONCAT('[', TRIM(' hello world '), ']') AS after_trim;
```

#### ব্যাখ্যা:

- TRIM() string-এর শুরু এবং শেষের spaces সরায়
- মাঝের spaces অঙ্গত থাকে
- User input clean করতে ব্যবহার হয়
- Leading এবং trailing whitespace remove করে

## আউটপুট:

**before\_trim after\_trim**

[ hello world ] [hello world]

মনে রাখার কথা: দুই পাশের space সরায়, মাঝেরটা নয়

---

## 10. LTRIM() - বাম পাশের space সরানো

### কোড:

```
SELECT CONCAT('[', ' hello world ', ']') AS original,  
       CONCAT('[', LTRIM(' hello world '), ']') AS left_trimmed;
```

### ব্যাখ্যা:

- LTRIM() শব্দু বাম পাশের (leading) spaces সরায়
- ডান পাশের spaces থাকে
- L মানে Left
- Specific formatting-এ কাজে লাগে

## আউটপুট:

**original left\_trimmed**

[ hello world ] [hello world ]

মনে রাখার কথা: শব্দু বাম পাশের space সরায়

---

## 11. RTRIM() - ডান পাশের space সরানো

### কোড:

```
SELECT CONCAT('[', ' hello world ', ']') AS original,  
       CONCAT('[', RTRIM(' hello world '), ']') AS right_trimmed;
```

### ব্যাখ্যা:

- RTRIM() শুধু ডান পাশের (trailing) spaces সরায়
- বাম পাশের spaces থাকে
- R মানে Right
- Output formatting-এ ব্যবহার হয়

**আউটপুট:**

**original      right\_trimmed**

[ hello world ] [ hello world]

**মনে রাখার কথা:** শুধু ডান পাশের space সরায়

---

## 12. REPLACE() - অংশ পরিবর্তন করা

**কোড:**

```
SELECT phone,
       REPLACE(phone, '-', '.') AS dots_format,
       REPLACE(phone, '+880', '0') AS local_format
FROM students;
```

**ব্যাখ্যা:**

- REPLACE(string, old\_text, new\_text) নির্দিষ্ট অংশ replace করে
- সব occurrences replace হয় (শুধু প্রথমটা নয়)
- Case-sensitive (MySQL-এ default)
- Data formatting এবং cleaning-এ ব্যবহার হয়

**আউটপুট:**

**phone      dots\_format      local\_format**

+880-1711-123456 +880.1711.123456 01711-123456

<b>phone</b>	<b>dots_format</b>	<b>local_format</b>
+880-1812-234567	+880.1812.234567	01812-234567
+880-1911-345678	+880.1911.345678	01911-345678
+880-1611-456789	+880.1611.456789	01611-456789

**মনে রাখার কথা:** পুরানো text নতুন text দিয়ে replace করে

---

### 13. REVERSE() - উল্টো করা

**কোড:**

```
SELECT student_name, REVERSE(student_name) AS reversed_name
FROM students;
```

**ব্যাখ্যা:**

- REVERSE() string-কে উল্টো করে দেয়
- শেষের character প্রথমে, প্রথমের শেষে
- Palindrome check করতে ব্যবহার হয়
- Algorithm এবং puzzle solve করতে কাজে লাগে

**আউটপুট:**

**student\_name reversed\_name**

alice johnson nosnhoj ecila

BOB SMITH HTIMS BOB

Charlie Brown nworB eilrahC

David Lee eeL divaD

**মনে রাখার কথা:** পুরো string উল্টো করে দেয়

---

### 14. LOCATE() / POSITION() / INSTR() - খুঁজে বের করা

**কোড:**

```
SELECT email,  
       LOCATE('@', email) AS at_position,  
       LOCATE('.', email) AS dot_position,  
       LOCATE('email', email) AS email_word_position  
  
FROM students;
```

**ব্যাখ্যা:**

- LOCATE(substring, string) substring-এর position খুঁজে দেয়
- প্রথম occurrence-এর position return করে
- না পেলে 0 return করে
- Position 1 থেকে শুরু হয় (0 নয়)
- POSITION() এবং INSTR() একই কাজ করে, syntax আলাদা

**আউটপুট:**

email	at_position	dot_position	email_word_position
alice@email.com	6	12	7
bob@EMAIL.COM	4	10	5
charlie@email.com	8	14	9
david@email.com	6	12	7

**মনে রাখার কথা:** substring-এর প্রথম position খুঁজে দেয়

---

## 15. LPAD() - বাম দিকে padding যোগ করা

**কোড:**

```
SELECT student_id,  
       LPAD(student_id, 5, '0') AS padded_id
```

```
FROM students;
```

#### ব্যাখ্যা:

- LPAD(string, total\_length, pad\_string) বাম দিকে characters add করে
- Total length পর্যন্ত pad করে
- ID formatting, zero-padding করতে ব্যবহার হয়
- যদি string already বড় হয় তাহলে truncate করে

#### আউটপুট:

**student\_id padded\_id**

1	00001
2	00002
3	00003
4	00004

**মনে রাখার কথা:** বাম দিকে character যোগ করে নির্দিষ্ট length বানায়

---

## 16. RPAD() - ডান দিকে padding যোগ করা

#### কোড:

```
SELECT product_name,  
       RPAD(product_name, 15, '.') AS formatted_name  
FROM products;
```

#### ব্যাখ্যা:

- RPAD(string, total\_length, pad\_string) ডান দিকে characters add করে
- Report formatting করতে কাজে লাগে
- Alignment তৈরি করতে ব্যবহার হয়
- Table-এর মতো output format করতে সুবিধা

## ଆଉଟପୁଟ:

**product\_name formatted\_name**

Laptop                  Laptop.....

Mouse.....

## Keyboard      Keyboard.....

**Monitor**      **Monitor.....**

**মনে বাথার কথা:** ডান দিকে character যোগ করে নির্দিষ্ট length বানায়

### 17. REPEAT() - বারবার পুনরাবৃত্তি করা

କାର୍ତ୍ତ

```
SELECT student_name,  
       REPEAT('*', LENGTH(student_name)) AS stars,  
       CONCAT(student_name, ' ', REPEAT('!', 3)) AS excited  
FROM students;
```

व्याख्या:

- REPEAT(string, count) নির্দিষ্ট string বারবার repeat করে
  - Pattern তৈরি করতে ব্যবহার হয়
  - Masking, decoration ইত্যাদিতে কাজে লাগে
  - Count 0 হলে empty string return করে

ଆউটপুট:

**student name stars**

alice johnson \*\*\*\*\* alice johnson !!!

BOB SMITH \*\*\*\*\* BOB SMITH !!!

**student\_name stars**                   **excited**

Charlie Brown \*\*\*\*\* Charlie Brown !!!

David Lee \*\*\*\*\* David Lee !!!

**মনে রাখার কথা:** একই string বারবার repeat করে

---

## 18. SPACE() - নির্দিষ্ট সংখ্যক space তৈরি করা

**কোড:**

```
SELECT CONCAT(student_name, SPACE(5), email) AS formatted_output  
FROM students;
```

**ব্যাখ্যা:**

- SPACE(count) নির্দিষ্ট সংখ্যক spaces তৈরি করে
- REPEAT(' ', count) এর shorthand
- Formatting এবং alignment-এ ব্যবহার হয়
- Fixed-width output তৈরি করতে সুবিধা

**আউটপুট:**

**formatted\_output**

alice johnson alice@email.com

BOB SMITH bob@EMAIL.COM

Charlie Brown charlie@email.com

David Lee david@email.com

**মনে রাখার কথা:** নির্দিষ্ট সংখ্যক space তৈরি করে

---

## 19. ASCII() - Character-এর ASCII মান

**কোড:**

```
SELECT student_name,  
       LEFT(student_name, 1) AS first_char,  
       ASCII(LEFT(student_name, 1)) AS ascii_value  
FROM students;
```

#### ব্যাখ্যা:

- ASCII(character) প্রথম character-এর ASCII code return করে
- A-Z: 65-90, a-z: 97-122
- Sorting order বুঝতে এবং comparison করতে কাজে লাগে
- শুধু প্রথম character-এর মান দেয়

#### আউটপুট:

**student\_name first\_char ascii\_value**

alice johnson	a	97
BOB SMITH	B	66
Charlie Brown	C	67
David Lee	D	68

**মনে রাখার কথা:** প্রথম character-এর ASCII মান

---

## 20. CHAR() - ASCII থেকে Character

#### কোড:

```
SELECT CHAR(65, 66, 67, 68) AS characters,  
       CHAR(72, 101, 108, 108, 111) AS hello;
```

#### ব্যাখ্যা:

- CHAR(ascii\_values) ASCII codes থেকে characters তৈরি করে
- একাধিক ASCII values একসাথে দেওয়া যায়

- ASCII() এর opposite function
- Special characters generate করতে ব্যবহার হয়

**আউটপুট:**

**characters hello**

ABCD      Hello

**মনে রাখার কথা:** ASCII code থেকে character তৈরি করে

---

## **21. STRCMP() - String তুলনা করা**

**কোড:**

SELECT

```
    STRCMP('apple', 'banana') AS compare1,  
    STRCMP('hello', 'hello') AS compare2,  
    STRCMP('zebra', 'apple') AS compare3;
```

**ব্যাখ্যা:**

- STRCMP(str1, str2) দুটি string compare করে
- Return values: 0 (same), -1 (str1 < str2), 1 (str1 > str2)
- Case-sensitive comparison
- Alphabetical order অনুযায়ী compare করে

**আউটপুট:**

**compare1 compare2 compare3**

-1            0            1

**মনে রাখার কথা:** -1 (ছোট), 0 (সমান), 1 (বড়)

---

## **22. FORMAT() - Number formatting**

**কোড:**

```
SELECT student_id,  
       FORMAT(student_id * 1000, 0) AS formatted_no_decimal,  
       FORMAT(student_id * 1000.5678, 2) AS formatted_two_decimal  
FROM students;
```

**ব্যাখ্যা:**

- `FORMAT(number, decimal_places)` number-কে formatted string-এ convert করে
- Thousand separator (,) automatically add করে
- Decimal places specify করা যায়
- Currency, percentage display করতে ব্যবহার হয়

**আউটপুট:**

student_id	formatted_no_decimal	formatted_two_decimal
1	1,000	1,000.57
2	2,000	2,001.14
3	3,000	3,001.70
4	4,000	4,002.27

**মনে রাখার কথা:** Number-কে formatted string-এ convert করে

---

### 23. `INSERT()` - নির্দিষ্ট অংশ প্রতিস্থাপন

**কোড:**

```
SELECT phone,  
       INSERT(phone, 10, 4, 'XXXX') AS masked_phone  
FROM students;
```

**ব্যাখ্যা:**

- INSERT(string, position, length, new\_string) নির্দিষ্ট অংশ replace করে
- Position থেকে শুরু করে length পর্যন্ত remove করে new\_string বসায়
- Data masking করতে ব্যবহার হয়
- Sensitive information hide করতে কাজে লাগে

**আউটপুট:**

<b>phone</b>	<b>masked_phone</b>
--------------	---------------------

+880-1711-123456 +880-1711XXXX56

+880-1812-234567 +880-1812XXXX67

+880-1911-345678 +880-1911XXXX78

+880-1611-456789 +880-1611XXXX89

**মনে রাখার কথা:** নির্দিষ্ট position থেকে নতুন string insert করে

---

## 24. FIELD() - Value-এর position খুঁজা

**কোড:**

```
SELECT product_name,
       FIELD(product_name, 'Mouse', 'Keyboard', 'Monitor', 'Laptop') AS priority
  FROM products;
```

**ব্যাখ্যা:**

- FIELD(value, val1, val2, val3...) list-এ value-এর position return করে
- না পেলে 0 return করে
- Custom sorting order তৈরি করতে ব্যবহার হয়
- ORDER BY-তে FIELD() দিয়ে custom order করা যায়

**আউটপুট:**

### **product\_name priority**

Laptop	4
Mouse	1
Keyboard	2
Monitor	3

**মনে রাখার কথা:** List-এ value-এর position বলে দেয়

---

### **25. ELT() - Position থেকে value পাওয়া**

**কোড:**

```
SELECT student_id,  
       ELT(student_id, 'First', 'Second', 'Third', 'Fourth', 'Fifth') AS rank_name  
FROM students;
```

**ব্যাখ্যা:**

- ELT(index, str1, str2, str3...) নির্দিষ্ট index-এর string return করে
- Index 1 থেকে শুরু হয়
- FIELD() এর opposite function
- Index out of range হলে NULL return করে

**আউটপুট:**

### **student\_id rank\_name**

1	First
2	Second
3	Third
4	Fourth

**মনে রাখার কথা:** Position number দিয়ে value পাওয়া

---

## String Functions তুলনা চাট:

Function	কাজ	Use Case
UPPER/LOWER	Case conversion	Case-insensitive search
LENGTH	Character count	Validation
CONCAT	String জোড়া	Full name, address
SUBSTRING	অংশ কাটা	Parsing, extraction
LEFT/RIGHT	শুরু/শেষ থেকে নেওয়া	Prefix/suffix
TRIM	Space সরানো	Data cleaning
REPLACE	Text পরিবর্তন	Formatting
LOCATE	Position খুঁজা	Parsing, validation
LPAD/RPAD	Padding যোগ করা	Formatting, ID

---

## কখন কোন Function ব্যবহার করবেন?

### Data Cleaning:

- TRIM, LTRIM, RTRIM: Extra spaces সরাতে
- UPPER, LOWER: Case normalize করতে
- REPLACE: Unwanted characters সরাতে

### Data Formatting:

- CONCAT, CONCAT\_WS: Multiple columns একসাথে করতে
- LPAD, RPAD: Fixed-width output তৈরি করতে
- FORMAT: Number display করতে

### Data Parsing:

- SUBSTRING, LEFT, RIGHT: Specific অংশ extract করতে
- LOCATE, POSITION: Delimiter position খুঁজতে
- FIELD: Custom sorting করতে

#### **Data Validation:**

- LENGTH: Minimum/maximum length check
  - LOCATE: Required characters আছে কিনা check
  - ASCII: Character range validation
- 

#### **Performance Tips:**

#### **ভালো Practice:**

- WHERE clause-এ function avoid করুন (index ব্যবহার হয় না)
- একবারে multiple operations করুন (nested functions)
- Stored columns ব্যবহার করুন frequently used calculations-এর জন্য

#### **এড়িয়ে চলুন:**

- প্রতিটি row-এ complex string operations (slow performance)
  - Unnecessary case conversions
  - অতিরিক্ত CONCAT operations
- 

#### **Advanced Examples:**

##### **Email থেকে username এবং domain extract:**

SELECT email,

    SUBSTRING(email, 1, LOCATE('@', email) - 1) AS username,

    SUBSTRING(email, LOCATE('@', email) + 1) AS domain

FROM students;

##### **Phone number masking:**

```
SELECT phone,  
       CONCAT(LEFT(phone, 9), REPEAT('*', LENGTH(phone) - 13), RIGHT(phone, 4)) AS masked  
  FROM students;
```

**Initials তৈরি করা:**

```
SELECT student_name,  
       CONCAT(UPPER(LEFT(student_name, 1)),  
              UPPER(SUBSTRING(student_name, LOCATE(' ', student_name) + 1, 1))) AS initials  
  FROM students;
```