**1. Creating a Sample Table Schema**

Let’s assume you want to manage information about employees. We can create a table called employees that will have several character-based columns:

sql

CopyEdit

CREATE TABLE employees (

id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL,

bio TEXT

);

*Explanation:*

* **id:** Unique identifier for each employee.
* **first\_name and last\_name:** Store names of the employees.
* **email:** Stores the employee’s email address.
* **bio:** A text field that might contain a description or bio (allowing longer character strings).

**2. Inserting Sample Data**

To work with our character functions, let’s insert some sample data into the employees table:

sql

CopyEdit

INSERT INTO employees (first\_name, last\_name, email, bio) VALUES

('Alice', 'Smith', 'alice.smith@example.com', 'Alice has been a valued team member since 2010.'),

('Bob', 'Johnson', 'bob.johnson@example.com', 'Bob is an experienced developer with a knack for creative solutions.'),

('Carol', 'Williams', 'carol.williams@example.com', 'Carol specializes in database administration and optimization.');

**3. Using Character Functions in Queries**

Now, let’s see how you can utilize various MySQL character functions on our employees table.

**a. Concatenating Names**

To combine the first and last names into a full name:

sql

CopyEdit

SELECT id, CONCAT(first\_name, ' ', last\_name) AS full\_name

FROM employees;

*This uses CONCAT() to join the first name and last name with a space in between.*

**b. Changing Case**

To display names in uppercase or lowercase:

sql

CopyEdit

-- Display first names in uppercase

SELECT id, UPPER(first\_name) AS first\_name\_upper

FROM employees;

-- Display last names in lowercase

SELECT id, LOWER(last\_name) AS last\_name\_lower

FROM employees;

*UPPER() and LOWER() transform the text accordingly.*

**c. Extracting a Substring**

Suppose you want to extract a portion of the email address (for example, the first 5 characters):

sql

CopyEdit

SELECT id, email, SUBSTRING(email, 1, 5) AS email\_prefix

FROM employees;

*SUBSTRING() extracts a specified portion of the string starting at a given position.*

**d. Measuring String Length**

To know the length of the bio field:

sql

CopyEdit

SELECT id, bio, CHAR\_LENGTH(bio) AS bio\_length

FROM employees;

*CHAR\_LENGTH() returns the number of characters in a string (as opposed to LENGTH() which returns the byte count, which might differ for multibyte characters).*

**e. Replacing Substrings**

If you need to update email domains on the fly, you can replace parts of the email string:

sql

CopyEdit

SELECT id, email, REPLACE(email, '@example.com', '@company.com') AS updated\_email

FROM employees;

*REPLACE() finds occurrences of a substring and replaces them with another string.*

**f. Trimming Extra Spaces**

In case your data has extra spaces that you want to remove, use the TRIM() function. For example, if the bio has unwanted spaces:

sql

CopyEdit

SELECT id, bio, TRIM(both ' ' FROM bio) AS trimmed\_bio

FROM employees;

*This removes leading and trailing spaces from the bio.*

**4. Putting It All Together**

Below is a summary script combining table creation, data insertion, and demonstration queries:

sql

CopyEdit

-- 1. Create the table

CREATE TABLE employees (

id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL,

bio TEXT

);

-- 2. Insert sample data

INSERT INTO employees (first\_name, last\_name, email, bio) VALUES

('Alice', 'Smith', 'alice.smith@example.com', 'Alice has been a valued team member since 2010.'),

('Bob', 'Johnson', 'bob.johnson@example.com', 'Bob is an experienced developer with a knack for creative solutions.'),

('Carol', 'Williams', 'carol.williams@example.com', 'Carol specializes in database administration and optimization.');

-- 3a. Concatenate first and last names

SELECT id, CONCAT(first\_name, ' ', last\_name) AS full\_name

FROM employees;

-- 3b. Convert names to uppercase/lowercase

SELECT id, UPPER(first\_name) AS first\_name\_upper,

LOWER(last\_name) AS last\_name\_lower

FROM employees;

-- 3c. Extract a substring from email addresses

SELECT id, email, SUBSTRING(email, 1, 5) AS email\_prefix

FROM employees;

-- 3d. Find the character length of the bio

SELECT id, bio, CHAR\_LENGTH(bio) AS bio\_length

FROM employees;

-- 3e. Replace email domain

SELECT id, email, REPLACE(email, '@example.com', '@company.com') AS updated\_email

FROM employees;

-- 3f. Trim spaces from bio (if necessary)

SELECT id, bio, TRIM(both ' ' FROM bio) AS trimmed\_bio

FROM employees;

**Conclusion**

MySQL's character functions are powerful tools that simplify text manipulation directly in your SQL queries. By using functions like CONCAT(), UPPER(), LOWER(), SUBSTRING(), CHAR\_LENGTH(), REPLACE(), and TRIM(), you can efficiently handle and process string data. The provided table schema and queries illustrate a practical application that you can modify to suit your needs.

Feel free to adjust the schema and queries as necessary to match your use case or data model!

2/2

**You said:**

also give me output

**ChatGPT said:**

Sure! Below is the **sample output** you would get when you run each query on the table employees created earlier with the sample data.

**✅ 1. Table Contents**

After inserting the sample data:

sql

CopyEdit

SELECT \* FROM employees;

**Output:**

| **id** | **first\_name** | **last\_name** | **email** | **bio** |
| --- | --- | --- | --- | --- |
| 1 | Alice | Smith | alice.smith@example.com | Alice has been a valued team member since 2010. |
| 2 | Bob | Johnson | bob.johnson@example.com | Bob is an experienced developer with a knack for creative solutions. |
| 3 | Carol | Williams | carol.williams@example.com | Carol specializes in database administration and optimization. |

**✅ 2. Concatenating First and Last Names**

sql

CopyEdit

SELECT id, CONCAT(first\_name, ' ', last\_name) AS full\_name FROM employees;

**Output:**

| **id** | **full\_name** |
| --- | --- |
| 1 | Alice Smith |
| 2 | Bob Johnson |
| 3 | Carol Williams |

**✅ 3. UPPER() and LOWER() Functions**

sql

CopyEdit

SELECT id, UPPER(first\_name) AS first\_name\_upper, LOWER(last\_name) AS last\_name\_lower FROM employees;

**Output:**

| **id** | **first\_name\_upper** | **last\_name\_lower** |
| --- | --- | --- |
| 1 | ALICE | smith |
| 2 | BOB | johnson |
| 3 | CAROL | williams |

**✅ 4. SUBSTRING() on Email**

sql

CopyEdit

SELECT id, email, SUBSTRING(email, 1, 5) AS email\_prefix FROM employees;

**Output:**

| **id** | **email** | **email\_prefix** |
| --- | --- | --- |
| 1 | alice.smith@example.com | alice |
| 2 | bob.johnson@example.com | bob.j |
| 3 | carol.williams@example.com | carol |

**✅ 5. CHAR\_LENGTH() of Bio**

sql

CopyEdit

SELECT id, bio, CHAR\_LENGTH(bio) AS bio\_length FROM employees;

**Output:**

| **id** | **bio** | **bio\_length** |
| --- | --- | --- |
| 1 | Alice has been a valued team member since 2010. | 48 |
| 2 | Bob is an experienced developer with a knack for creative solutions. | 74 |
| 3 | Carol specializes in database administration and optimization. | 66 |

**✅ 6. REPLACE() in Email**

sql

CopyEdit

SELECT id, email, REPLACE(email, '@example.com', '@company.com') AS updated\_email FROM employees;

**Output:**

| **id** | **email** | **updated\_email** |
| --- | --- | --- |
| 1 | alice.smith@example.com | alice.smith@company.com |
| 2 | bob.johnson@example.com | bob.johnson@company.com |
| 3 | carol.williams@example.com | carol.williams@company.com |

**✅ 7. TRIM() Function (if bios had spaces)**

Let’s assume Carol’s bio had extra spaces at the start and end. If we run:

sql

CopyEdit

SELECT id, bio, TRIM(BOTH ' ' FROM bio) AS trimmed\_bio FROM employees;

**Output (if bios had leading/trailing spaces):**

| **id** | **bio** | **trimmed\_bio** |
| --- | --- | --- |
| 1 | Alice has been a valued team member since 2010. | Alice has been a valued team member since 2010. |
| 2 | Bob is an experienced developer with a knack for creative solutions. | Bob is an experienced developer with a knack for creative solutions. |
| 3 | Carol specializes in database administration and optimization. | Carol specializes in database administration and optimization. |

*(Only Carol's bio had spaces removed)*