

¡Genial! Las **String Functions** en SQL son funciones que nos permiten manipular y analizar datos de tipo texto (VARCHAR, CHAR, TEXT). Son muy útiles en análisis de datos para limpiar, transformar y extraer información de cadenas de texto.

◆ Principales String Functions en SQL

Voy a explicarte las funciones más usadas con ejemplos prácticos.

1. UPPER() y LOWER() - Convertir texto a mayúsculas o minúsculas

Sirven para estandarizar datos de texto.

```
SELECT UPPER('hola mundo'); -- Resultado: 'HOLA MUNDO'
```

```
SELECT LOWER('HOLA MUNDO'); -- Resultado: 'hola mundo'
```

📌 Ejemplo práctico:

Si en una base de datos los nombres de clientes están escritos de manera inconsistente, podemos normalizarlos.

```
SELECT Nombre, UPPER(Nombre) AS Nombre_Normalizado FROM Clientes;
```

2. LENGTH() - Contar caracteres en una cadena

Nos dice cuántos caracteres tiene una cadena (incluye espacios).

```
SELECT LENGTH('Hola Mundo'); -- Resultado: 10
```

📌 Ejemplo práctico:

Encontrar clientes con nombres muy cortos o largos:

```
SELECT Nombre FROM Clientes WHERE LENGTH(Nombre) > 15;
```

3. TRIM(), LTRIM(), RTRIM() - Eliminar espacios en blanco

- TRIM() quita los espacios al inicio y al final.
- LTRIM() elimina espacios **a la izquierda**.
- RTRIM() elimina espacios **a la derecha**.

```
SELECT TRIM(' Hola Mundo '); -- Resultado: 'Hola Mundo'
```

```
SELECT LTRIM(' Hola Mundo'); -- Resultado: 'Hola Mundo'
```

```
SELECT RTRIM('Hola Mundo '); -- Resultado: 'Hola Mundo'
```

📌 Ejemplo práctico:

Eliminar espacios en nombres ingresados incorrectamente:

```
SELECT TRIM(Nombre) FROM Clientes;
```

4. SUBSTRING() - Extraer parte de una cadena

Nos permite obtener una porción de texto.

```
SELECT SUBSTRING('Hola Mundo', 1, 4); -- Resultado: 'Hola'
```

(Desde la posición 1, toma 4 caracteres)

📌 Ejemplo práctico:

Si tenemos códigos de productos tipo "PROD-12345" y queremos extraer solo el número:

```
SELECT SUBSTRING(Codigo_Producto, 6, 5) FROM Productos;
```

5. REPLACE() - Reemplazar texto

Sustituye una parte de una cadena por otra.

```
SELECT REPLACE('Hola Mundo', 'Mundo', 'SQL'); -- Resultado: 'Hola SQL'
```

📌 Ejemplo práctico:

Si en una base de datos de emails todos los correos son "@gmail.com" y queremos cambiarlos a "@empresa.com":

```
SELECT REPLACE(Email, '@gmail.com', '@empresa.com') FROM Clientes;
```

6. CONCAT() - Unir texto

Concatena (une) cadenas de texto.

```
SELECT CONCAT('Hola', ' ', 'Mundo'); -- Resultado: 'Hola Mundo'
```

📌 Ejemplo práctico:

Unir nombre y apellido en una consulta:

```
SELECT CONCAT(Nombre, ' ', Apellido) AS Nombre_Completo FROM Clientes;
```

7. LOCATE() - Buscar una palabra dentro de un texto

Devuelve la posición donde aparece una palabra dentro de una cadena.

```
SELECT LOCATE('Mundo', 'Hola Mundo'); -- Resultado: 6
```

(Mundo empieza en la posición 6)

📌 Ejemplo práctico:

Buscar clientes que tengan "Inc." en el nombre de su empresa:

```
SELECT Nombre FROM Empresas WHERE LOCATE('Inc.', Nombre) > 0;
```

Proyecto práctico usando String Functions

Escenario:

Tienes una base de datos de clientes y quieres limpiar y formatear la información de los nombres y correos electrónicos.

Tablas disponibles:

Cientes

| ClienteID | Nombre | Apellido | Email |
|-----------|--------|----------|--|
| 1 | Ana | Pérez | ana123 @gmail.com |
| 2 | Luis | Gómez | luis.gomez@gmail.com |
| 3 | Marta | López | marta@outlook.com |

Objetivo:

1. Normalizar los nombres y apellidos (primera letra mayúscula).
2. Limpiar espacios en los nombres.
3. Unir nombre y apellido en una sola columna.
4. Formatear los correos electrónicos (eliminar espacios y cambiar a minúsculas).

Consulta SQL usando String Functions

SELECT

 ClienteID,

 CONCAT(

 UPPER(LEFT(TRIM(Nombre), 1)), LOWER(SUBSTRING(TRIM(Nombre), 2))

) AS Nombre_Limpio,

 CONCAT(

 UPPER(LEFT(TRIM(Apellido), 1)), LOWER(SUBSTRING(TRIM(Apellido), 2))

) AS Apellido_Limpio,

 CONCAT(

 UPPER(LEFT(TRIM(Nombre), 1)), LOWER(SUBSTRING(TRIM(Nombre), 2)), ' ',

 UPPER(LEFT(TRIM(Apellido), 1)), LOWER(SUBSTRING(TRIM(Apellido), 2))

) AS Nombre_Completo,

 LOWER(TRIM(REPLACE>Email, ' ', ''))) AS Email_Limpio

FROM Clientes;

Explicación del código:

- ✓ **Limpia espacios en los nombres:** TRIM(Nombre).
- ✓ **Convierte la primera letra en mayúscula y el resto en minúsculas:**

UPPER(LEFT(Nombre, 1)) || LOWER(SUBSTRING(Nombre, 2))

- ✓ **Concatena nombre y apellido en una sola columna:** CONCAT(Nombre, ' ', Apellido).
- ✓ **Corrige los correos electrónicos eliminando espacios y convirtiéndolos a minúsculas:** LOWER(TRIM(REPLACE>Email, ' ', ''))).

Resultado esperado después del procesamiento

| ClienteID | Nombre_Limpio | Apellido_Limpio | Nombre_Completo | Email_Limpio |
|-----------|---------------|-----------------|-----------------|--|
| 1 | Ana | Pérez | Ana Pérez | ana123@gmail.com |
| 2 | Luis | Gómez | Luis Gómez | luis.gomez@gmail.com |
| 3 | Marta | López | Marta López | marta@outlook.com |

Conclusión

Las **String Functions en SQL** son muy útiles para limpiar y transformar datos de texto. En este proyecto, usamos varias funciones para:

- **Estandarizar nombres**
- **Eliminar espacios en blanco**
- **Formatear correos electrónicos**
- **Combinar información de diferentes columnas**

🔥 ¡Así es como los analistas de datos trabajan con información desordenada y la convierten en datos útiles!

♦ **¿Te gustaría agregar más análisis a este proyecto?** Por ejemplo, contar cuántos clientes usan Gmail, Outlook, etc. 🚀