



Arquitectura y Administración de bases de datos con SQL 2021

Héctor Manuel Garduño Castañeda

Diciembre, 2021



Contenido

Length

+ Las funciones de caracter se refiere a qué podemos hacer con los datos de tipo caracter (char(n), varchar, text)

Upper y Lower

Replace

TRIM, LTRIM y RTRIM

Concatenación

Subcadenas

Concatenación con limitadores



+ Esta es una función vectorizadas

+ Toma en cuenta los espacios

La función **LENGTH** devuelve la longitud ~~del valor~~ de caracter que se le ingresa, expresada como el número de caracteres que lo forman. Su sintaxis es **LENGTH(texto)**.

```
SELECT LENGTH('hola');
```

```
SELECT customer_name, LENGTH(customer_name) AS caracteres
FROM customer
WHERE age > 30;
```

```
SELECT customer_name as "nombres largos"
FROM customer
WHERE LENGTH(customer_name) > 10;
```

+ Una FUNCIÓN PUNUTAL permite un solo parámetro, que se aplique a un solo objeto

+ Una FUNCION VECTORIZADA permite varios parámetros, se puede aplicar a cada uno de los elementos de una columna



La funciones **UPPER/LOWER** convierten todos los caracteres a mayúsculas o minúsculas. La sintaxis es **UPPER**(texto) y **LOWER**(texto).

```
SELECT UPPER('Yo soy Popeye');
```

```
SELECT customer_name,
UPPER(customer_name) AS May, LOWER(customer_name) AS Min
FROM customer
WHERE age > 30;
```



La función **REPLACE** reemplaza todas las apariciones de un texto específico. La sintaxis es **REPLACE**(texto donde se va a reemplazar, texto que se va a reemplazar, texto con el que se va a reemplazar). Es sensible a mayúsculas y minúsculas.

```
SELECT REPLACE('Yo soy Popeye','soy','quiero ser');
```

```
SELECT customer_name, country,  
REPLACE(country,'United States','US') AS "country new"  
FROM customer;
```

```
SELECT customer_name, country,  
REPLACE(LOWER(country),'united states','US') AS "country new"  
FROM customer;
```



+ Cuando dice que elimina la "cadena máxima" quiere decir que borrará todos aquellos que encuentre (elementos consecutivos) hasta que se encuentre con otro caracter distinto

La función **TRIM** elimina la cadena máxima de todos los caracteres especificados de un texto específico. La función **LTRIM** elimina todos los caracteres especificados de un texto específico desde la izquierda. La función **RTRIM** elimina todos los caracteres especificados de un texto específico desde la derecha.

La sintaxis es

```
TRIM(LEADING '<<texto a quitar>>' FROM <<texto de donde se va a
quitar>>)
TRIM(TRAILING '<<texto a quitar>>' FROM <<texto de donde se va a
quitar>>)
TRIM(BOTH '<<texto a quitar>>' FROM <<texto de donde se va a
quitar>>)
LTRIM(<<texto de donde se va a quitar>>,'<<texto que se va a quitar>>')
RTRIM(<<texto de donde se va a quitar>>,'<<texto que se va a quitar>>')
```



+ Los tres sirven para eliminar str dentro de los textos
(con los ejemplos queda mas claro)

+ LEADING es para que busque despues la izq
 +TRAILING es para que busque desde la derecha

+ Esto es útil para borrar espacios. LEADING y TRAILING borrarán espacios del lado izq y derecho (respectivamente) pero cuando se topen con un caracter distinto al espacio, tendrán el borrado

```
SELECT TRIM(LEADING 'A' FROM 'AAA Yo soy Popeye BB');
SELECT TRIM(TRAILING 'B' FROM 'AAA Yo soy Popeye BB');
SELECT TRIM(BOTH ' ' FROM ' Yo soy Popeye BB ');
SELECT LTRIM(' Yo soy Popeye ',' ')
SELECT RTRIM(' Yo soy Popeye ',' ')
```

-- LEADING es para decir que quiero que busque desde izq a derecha
 -- Luego se pone qué quiero que busque
 -- FROM y el texto en donde quiero que busque
 -- y todo esto lo va a hacer para eliminarla (por eso es TRIM, esto hace que lo borre)

-- Como dice que "elimina la cadena máxima" entonces va a eliminar las 3 "A", no elimina de uno en uno
 -- No mata a los espacios
 ESTO ES DEL PRIMER EJEMPLO



debajo de la tecla "esc"

El operador || nos permite concatenar textos. Su sintaxis es <<texto>> || texto <<texto>> || <<texto>>...

SELECT customer_name, city || ', ' || state || ', ' || country **AS** direccion
FROM customer;

+ Muestra dos columnas, una con el nombre del cliente y una columna llamada direccion que consta de la concatenación que hacemos.



La función **SUBSTRING** nos permite extraer un subtexto de un texto. Su sintaxis es **SUBSTRING**(⟨⟨texto original⟩⟩ **FROM** ⟨⟨posición inicial⟩⟩ **FOR** ⟨⟨posición final⟩⟩).

```
SELECT customer_id,customer_name,
      SUBSTRING(customer_id FOR 2) AS cust_grupo
FROM customer
WHERE SUBSTRING(customer_id FOR 2) = 'AB';
```

```
SELECT customer_id,customer_name,
      SUBSTRING(customer_id FROM 4 FOR 5) AS cust_numero
FROM customer
WHERE SUBSTRING(customer_id FOR 2) = 'AB';
```

```
SELECT SUBSTRING( 'Este es un texto' FROM 1 TO 3)
-- Devuelve Est
```

+ **FROM** es el inicio y **FOR** es la cantidad de caracteres que queremos

+ Cuando omitimos el **FROM**, lo toma desde 1



+ En este concatenamos elementos de una misma columna, a diferencia del otro que podíamos concatenar elementos de varias columnas

La función **STRING_AGG** concatena valores de texto dentro de una lista mediante un limitador. Por lo tanto es una función vectorizada. Su sintaxis es **STRING_AGG**(⟨⟨expresión⟩⟩, '⟨delimitador⟩').

```
SELECT STRING_AGG(product_id, ', ' from sales;
```

Cuando se le usa con **GROUP BY** se puede ver su importancia:

```
SELECT order_id, STRING_AGG(product_id, ', ')
```

```
FROM sales
```

```
GROUP BY order_id;
```

+ Por cada uno de los order_id nos va a mostrar los product_id que tienen ese mismo valor de order_id pero los va a mostrar concatenados por una coma y espacio

```
SELECT state, STRING_AGG(customer_id, ', ')
```

```
FROM customer
```

```
GROUP BY state;
```

+ El primer ejemplo concatena a todos los elementos de product_id con una coma y un espacio en un solo registro (un renglón)

+ El delimitador es arbitrario, el que necesitemos

