*Trasformando un string a minúsculas*

La función LOWER() en SQLite se utiliza para convertir todos los caracteres de un texto a minúsculas. Por ejemplo, si tenemos una tabla 'usuarios' con un campo 'nombre' que contiene nombres en mayúsculas, podemos convertir todos los nombres a minúsculas utilizando la siguiente consulta:

SELECT LOWER(nombre) AS nombre\_minusculas FROM productos;

Esto nos devolverá una lista de todos los nombres en la tabla 'usuarios', pero en minúsculas. La función LOWER() no modifica los datos en la tabla, sólo los transforma para los resultados de la consulta.

Ejercicio

Se tiene una **tabla** de **usuarios** con los campos id, nombre, e email. El campo email es de tipo texto y contiene algunas mayúsculas, lo que puede ocasionar errores en la base de datos.

Selecciona los emails de la tabla usuarios con el alias email\_lower. Todos los emails deben ser mostrados en minúsculas.

select lower(email) as email\_lower from usuarios

*Quitando espacios en blanco de un string*

En SQLite la función TRIM() se utiliza para eliminar los espacios en blanco iniciales y finales de un string.

Por ejemplo, si tenemos una tabla de productos con una columna 'nombre' que contiene espacios en blanco al inicio y final de cada nombre, podemos utilizar la siguiente consulta para quitar esos espacios:

SELECT TRIM(nombre) FROM productos;

Esto nos devolverá los nombres de los productos sin los espacios en blanco al inicio y final.

Ejercicio

Se tiene una **tabla** de **usuarios** con las columnas nombre, apellido, email y teléfono. Los nombres y correos poseen espacios en blanco tanto al inicio como al final de su valor. Utiliza la función TRIM() para seleccionar los nombres e emails y quitar los espacios en blanco.

select trim(nombre), trim(email) from usuarios

*Combinando funciones*

En SQL podemos combinar funciones. Veamos un ejemplo combinando LOWER y TRIM :

SELECT LOWER(TRIM(email)) as email\_limpios from usuarios;

Esta consulta selecciona los correos electrónicos de la tabla "usuarios", los convierte a minúsculas y elimina cualquier espacio en blanco adicional alrededor de ellos. El resultado será una lista de correos electrónicos "limpios" y en minúsculas.

Ejercicio

Se tiene una **tabla** de **usuarios** con las columnas nombre, apellido, email y teléfono. Los nombres, apellidos y correos poseen espacios en blanco tanto al inicio como al final y algunos de ellos tienen mayúsculas.

Utiliza lo aprendido para seleccionar los nombres, emails y apellidos, limpiando cada uno de estos campos. Para que el resultado sea correcto debes ocupar los alias nombre\_limpio, apellido\_limpio e email\_limpio respectivamente.

SELECT LOWER

(TRIM(nombre)) AS nombre\_limpio,

LOWER(TRIM(apellido)) AS apellido\_limpio,

LOWER(TRIM(email)) AS email\_limpio

fROM usuarios;

*Obteniendo el largo de un string*

En SQL, podemos utilizar la función LENGTH() para obtener la longitud de una cadena de caracteres. Por ejemplo, si queremos obtener la longitud del nombre de todos los usuarios en la tabla 'usuarios', podríamos utilizar la siguiente consulta:

SELECT nombre, LENGTH(nombre) FROM usuarios;

Esto nos devolverá una lista de nombres junto con su longitud respectiva.

Ejercicio

Selecciona el largo del apellido de todos los usuarios en la **tabla usuarios**.

SELECT length(apellido) from usuarios

*Obteniendo el nombre mas largo de la tabla*

Ya vimos en el ejercicio anterior que, para calcular el largo de una cadena de caracteres, debemos utilizar la función LENGTH(). Si queremos obtener la cadena más corta de la columna, debemos combinar la función LENGTH() con ORDER BY y LIMIT.

Por ejemplo, si queremos seleccionar el largo del nombre más corto de la tabla usuarios, la consulta quedaría así:

SELECT LENGTH(nombre) as largo\_nombre FROM usuarios ORDER BY LENGTH(nombre) LIMIT 1 ;

Por otro lado, si queremos obtener el largo del nombre más largo, invertiremos el orden de la selección.

SELECT LENGTH(nombre) as largo\_nombre FROM usuarios ORDER BY LENGTH(nombre) DESC LIMIT 1 ;

Esto nos devolverá la longitud del nombre más largo en la tabla 'usuarios'.

Ejercicios

Se tiene una **tabla usuarios** con las columnas nombre, apellido, email y teléfono.

Utiliza lo aprendido para seleccionar el largo de los 3 correos más largos de la tabla. La columna resultante debe mostrar sólo el largo (cantidad de caracteres) de estos correos.

select length (email) from usuarios order by length(email) desc 3;

*Ordenando todos los datos y la funcion*

En el ejercicio previo, habíamos optado por seleccionar únicamente la longitud (largo) de una columna. No obstante, en ciertos casos, puede ser necesario recuperar tanto la información de la columna en sí como su longitud. Por ejemplo, ¿qué ocurriría si se nos solicita el correo electrónico más largo en lugar de simplemente su longitud?

Para esto podemos hacer una consulta como la siguiente:

SELECT email FROM usuarios ORDER BY LENGTH(email) DESC LIMIT 1;

Además, es posible que se nos solicite seleccionar todos los campos del usuario cuyo correo sea el más largo. Para lograrlo, podemos utilizar la siguiente consulta:

SELECT \* FROM usuarios ORDER BY LENGTH(email) DESC LIMIT 1;

Esta consulta nos devolverá todas las filas de la tabla "usuarios" donde la longitud del correo electrónico sea igual a la longitud máxima encontrada en toda la tabla.

Asimismo, podrían requerir que seleccionemos todos los campos de la tabla y, además, incluir el largo del correo electrónico. La idea es similar, simplemente agregamos la función al SELECT:

SELECT \*, LENGTH(email) as largo\_email FROM usuarios ORDER BY LENGTH(email) DESC LIMIT 1;

Lo importante es entender qué nos están pidiendo para poder realizar las consultas correctas.

*Ejercicios*

Se tiene una **tabla usuarios** con las columnas nombre, apellido, email y teléfono.

Utiliza lo aprendido para seleccionar los 3 correos más largos de la tabla. El resultado debe mostrar dos columnas: una con los emails y otra con sus largos respectivos.

SELECT email,LENGTH(email) from usuarios order by LENGTH(email) desc limit 3

*Concatenar strings*

En este ejercicio aprenderemos a juntar textos. Por ejemplo, si tenemos una columna con un nombre y otra con un apellido, podemos generar una única columna con el nombre y apellido. A esto se le llama *concatenar* y utilizaremos el operador **||**

Un ejemplo de consulta con concatenación sería la siguiente:

SELECT nombre || ' ' || apellido AS nombre\_completo FROM empleados;

En esta consulta, estamos concatenando el nombre y el apellido de cada empleado, separados por un espacio, y utilizando el alias 'nombre\_completo' para la nueva columna creada.

Ejercicio

Supongamos que tienes una **tabla** llamada **productos** con los campos 'producto', 'marca' y 'precio'. Selecciona una lista de todos los productos con su nombre, seguido de un guion ("-"), y su marca. Asigna el alias 'marca\_producto' a la columna creada.

select producto || '-' || marca AS marca\_producto FROM productos

Seleccionando caracteres de un string con SUBSTR

La función SUBSTR() se utiliza para seleccionar una determinada cantidad de caracteres de un string:

SUBSTR( string, inicio, largo )

En la sintaxis podemos observar que la función tiene 3 argumentos. 1. String: el nombre de la columna o palabra que será utilizada 2. Inicio: un integer que especifica la posicion de inicio desde la cual se extraerán caracteres al string. 3. Largo: la cantidad de caracteres extraidos

Por ejemplo, si tenemos una tabla de productos con el campo 'nombre' y queremos seleccionar sólo la primera letra de cada nombre, podemos utilizar la siguiente consulta:

SELECT SUBSTR(nombre, 1, 1) AS primera\_letra FROM productos;

En este ejemplo estamos indicando que en la columna nombre, partiendo desde su primera letra, nos devuelva sólo 1 caracter que corresponderá a la primera letra de cada nombre.

Es importante recordar que algunas funciones tienen distintos nombres dependiendo del motor de base de datos. Por ejemplo, para lograr este mismo objetivo en PostgreSQL, deberíamos usar la función LEFT()

Ejercicio

Se tiene una **tabla usuarios** con las columnas id, nombre, apellido, email y teléfono. Utiliza la función SUBSTR para seleccionar las tres primeras letras del apellido de cada usuario en la tabla 'usuarios'. Asigna el nombre 'primeras\_letras' a la columna creada.

select substr (apellido, 1,3) as primeras\_letras from usuarios;

Seleccionando caracteres

Se tiene una **tabla** de **usuarios** con las columnas nombre y apellido. Utilizando la función SUBSTR(), selecciona 3 caracteres del apellido de María, partiendo desde el segundo caracter. Asigna el alias 'tres\_caracteres\_del\_apellido' a la columna creada.

select substr (apellido,2,3) as tres\_caracteres\_del\_apellido

from usuarios

where nombre = 'María';