

Funciones incorporadas en MySQL

Funciones para manejo de cadenas

Cuando usamos funciones NO debe haber espacios entre un nombre de función y los paréntesis porque MySQL puede confundir una llamada a una función con una referencia a una tabla o campo que tenga el mismo nombre de una función.

MySQL tiene funciones para trabajar con cadenas de caracteres.

Vamos a mostrar algunas, recordamos que pueden variar según la versión del MySQL en la que estemos trabajando.

ord(caracter): Retorna el código ASCII para el caracter enviado como argumento. Ejemplo:

```
select ord('A');
```

retorna 65.

char(x,...): retorna una cadena con los caracteres en código ASCII de los enteros enviados como argumentos. Ejemplo:

```
select char(65,66,67);
```

retorna "ABC".

concat(cadena1,cadena2,...): devuelve la cadena resultado de concatenar los argumentos. Ejemplo:

```
select concat('Hola,', 'como esta?');
```

retorna "Hola, como esta?".

concat_ws(separador,cadena1,cadena2,...): "ws" son las iniciales de "with separator". El primer argumento especifica el separador que utiliza para los demás argumentos; el separador se agrega entre las cadenas a concatenar. Ejemplo:

```
select concat_ws('-', 'Juan', 'Pedro', 'Luis');
```

retorna "Juan-Pedro-Luis".

find_in_set(cadena,lista de cadenas): devuelve un valor entre de 0 a n (*correspondiente a la posición*), si la cadena enviada como primer argumento está presente en la lista de cadenas enviadas como segundo argumento. La lista de cadenas enviada como segundo argumento es una cadena formada por subcadenas separadas por comas. Devuelve 0 si no encuentra "cadena" en la "lista de cadenas". Ejemplo:

```
select find_in_set('hola','como esta,hola,buen dia');
```

retorna 2, porque la cadena "hola" se encuentra en la lista de cadenas, en la posición 2.

length(cadena): retorna la longitud de la cadena enviada como argumento. Ejemplo:

```
select length('Hola');
```

devuelve 4.

locate(subcadena,cadena): retorna la posición de la primera ocurrencia de la subcadena en la cadena enviadas como argumentos. Devuelve "0" si la subcadena no se encuentra en la cadena. Ejemplo:

```
select locate('o','como le va');
```

retorna 2.

position(subcadena in cadena): funciona como "locate()". Devuelve "0" si la subcadena no se encuentra en la cadena. Ejemplo:

```
select position('o' in 'como le va');
```

retorna 2.

locate(subcadena,cadena,posicioninicial): retorna la posición de la primera ocurrencia de la subcadena enviada como primer argumentos en la cadena enviada como segundo argumento, empezando en la posición enviada como tercer argumento. Devuelve "0" si la subcadena no se encuentra en la cadena. Ejemplos:

```
select locate('ar','Margarita',1);
```

retorna 1.

```
select locate('ar','Margarita',3);
```

retorna 5.

instr(cadena,subcadena): retorna la posición de la primera ocurrencia de la subcadena enviada como segundo argumento en la cadena enviada como primer argumento. Ejemplo:

```
select instr('como le va','om');
```

devuelve 2.

lpad(cadena,longitud,cadenarelleno): retorna la cadena enviada como primer argumento, rellenada por la izquierda con la cadena enviada como tercer argumento hasta que la cadena retornada tenga la longitud especificada como segundo argumento. Si la cadena es más larga, la corta. Ejemplo:

```
select lpad('hola',10,'0');
```

retorna "000000hola".

rpad(cadena,longitud,cadenarelleno): igual que "lpad" excepto que rellena por la derecha.

left(cadena,longitud): retorna la cantidad (longitud) de caracteres de la cadena comenzando desde la izquierda, primer caracter. Ejemplo:

```
select left('buenos dias',8);
```

retorna "buenos d".

right(cadena,longitud): retorna la cantidad (longitud) de caracteres de la cadena comenzando desde la derecha, último caracter. Ejemplo:

```
select right('buenos dias',8);
```

retorna "nos dias".

substring(cadena,posicion,longitud): retorna una subcadena de tantos caracteres de longitud como especifica en tercer argumento, de la cadena enviada como primer argumento, empezando desde la posición especificada en el segundo argumento. Ejemplo:

```
select substring('Buenas tardes',3,5);
```

retorna "enas".

substring(cadena from posicion for longitud): variante de **substring(cadena,posicion,longitud)**. Ejemplo:

```
select substring('Buenas tardes' from 3 for 5);
```

mid(cadena,posicion,longitud): igual que "substring(cadena,posicion,longitud)". Ejemplo:

```
select mid('Buenas tardes' from 3 for 5);
```

retorna "enas".

substring(cadena,posicion): retorna la subcadena de la cadena enviada como argumento, empezando desde la posición indicada por el segundo argumento. Ejemplo:

```
select substring('Margarita',4);
```

retorna "garita".

substring(cadena from posicion): variante de "substring(cadena,posicion)". Ejemplo:

```
select substring('Margarita' from 4);
```

retorna "garita".

substring_index(cadena,delimitador,ocurrencia): retorna la subcadena de la cadena enviada como argumento antes o después de la "ocurrencia" de la cadena enviada como delimitador. Si "ocurrencia" es positiva, retorna la subcadena anterior al delimitador (comienza desde la izquierda); si "ocurrencia" es negativa, retorna la subcadena posterior al delimitador (comienza desde la derecha). Ejemplo:

```
select substring_index( 'margarita','ar',2);
```

retorna "marg", todo lo anterior a la segunda ocurrencia de "ar".

```
select substring_index( 'margarita','ar',-2);
```

retorna "garita", todo lo posterior a la segunda ocurrencia de "ar".

ltrim(cadena): retorna la cadena con los espacios de la izquierda eliminados. Ejemplo:

```
select ltrim('  Hola  ');
```

retorna

"Hola "

rtrim(cadena): retorna la cadena con los espacios de la derecha eliminados. Ejemplo:

```
select rtrim('  Hola  ');
```

retorna

" Hola"

trim([both|leading|trailing] [subcadena] from) cadena): retorna una cadena igual a la enviada pero eliminando la subcadena prefijo y/o sufijo. Si no se indica ningún especificador (both, leading o trailing) se asume "both" (ambos). Si no se especifica prefijos o sufijos elimina los espacios. Ejemplos:

```
select trim('  Hola  ');
```

retorna 'Hola'.

```
select trim (leading '0' from '00hola00');
```

retorna "hola00".

```
select trim (trailing '0' from '00hola00');
```

retorna "00hola".

```
select trim (both '0' from '00hola00');
```

retorna "hola".

```
select trim ('0' from '00hola00');
```

retorna "hola".

```
select trim ('  hola  ');
```

retorna "hola".

replace(cadena,cadenareemplazo,cadenareemplazar): retorna la cadena con todas las ocurrencias de la subcadena reemplazo por la subcadena a reemplazar. Ejemplo:

```
select replace('xxx.mysql.com','x','w');
```

retorna "www.mysql.com".

repeat(cadena,cantidad): devuelve una cadena consistente en la cadena repetida la cantidad de veces especificada. Si "cantidad" es menor o igual a cero, retorna una cadena vacía. Ejemplo:

```
select repeat('hola',3);
```

retorna "holaholahola".

reverse(cadena): devuelve la cadena invirtiendo el order de los caracteres. Ejemplo:

```
select reverse('Hola');
```

retorna "aloH".

insert(cadena,posicion,longitud,nuevacadena): retorna la cadena con la nueva cadena colocándola en la posición indicada por "posicion" y elimina la cantidad de caracteres indicados por "longitud". Ejemplo:

```
select insert('buenas tardes',2,6,'xx');
```

retorna "xxxtardes".

lcase(cadena) y lower(cadena): retornan la cadena con todos los caracteres en minúsculas. Ejemplo:

```
select lower('HOLA ESTUDIAnte');
```

retorna "hola estudiante".

```
select lcase('HOLA ESTUDIAnte');
```

retorna "hola estudiante".

ucase(cadena) y upper(cadena): retornan la cadena con todos los caracteres en mayúsculas. Ejemplo:

```
select upper('HOLA ESTUDIAnte');
```

retorna "HOLA ESTUDIANTE".

```
select ucase('HOLA ESTUDIAnte');
```

retorna "HOLA ESTUDIANTE".

strcmp(cadena1,cadena2): retorna 0 si las cadenas son iguales, -1 si la primera es menor que la segunda y 1 si la primera es mayor que la segunda. Ejemplo:

```
select strcmp('Hola','Chau');
```

retorna 1.

Un listado completo de todas las funciones para el manejo de cadenas de caracteres de las diferentes versiones de MySQL las podemos consultar en la [documentación oficial](#).