



Funciones en MySQL



Funciones

- **MySQL puede hacer mucho más que simplemente almacenar y recuperar datos .**
- Puede **realizar manipulaciones en los datos** antes de recuperarlos o guardarlos.
- Las funciones son simplemente piezas de código que realizan algunas operaciones y luego devuelven un resultado. Algunas funciones aceptan parámetros mientras que otras funciones no aceptan parámetros.



Ventajas de usar funciones en MySQL

- Reduce el re-trabajo en la lógica comercial y reduce las inconsistencias de datos.
- puede ayudar a reducir el tráfico de red en las aplicaciones cliente / servidor

Tipos de Funciones

- **Funciones integradas** son funciones que ya vienen implementadas en el servidor MySQL.

Estas funciones permiten realizar diferentes tipos de manipulaciones en los datos. Las funciones integradas se pueden clasificar básicamente .

- **Funciones de cadenas** – operan en tipos de datos de cadena
- **Funciones numéricas** : opere en tipos de datos numéricos
- **Funciones de fecha** : operan en tipos de datos de fecha
- **Funciones agregadas** : opere en todos los tipos de datos anteriores y produzca conjuntos de resultados resumidos.
- **Funciones de control de flujo** : se emplea como un Switch o Case para el retorno de datos de acuerdo a un condicional.

Funciones de cadena

- **ASCII(str)** Retorna el valor numérico del carácter más a la izquierda de la cadena de caracteres str. Retorna 0 si str es la cadena vacía. Retorna NULL si str es NULL. ASCII() funciona para caracteres con valores numéricos de 0 a 255.

```
SELECT ASCII('2');
```

-> 50

- **BIN(N)** Retorna una representación de cadena de caracteres del valor binario de N, donde N es un número muy largo (BIGINT) . Esto es equivalente a CONV(N,10,2). Retorna NULL si N es NULL.

```
SELECT BIN(12);
```

-> '1100'

- **CHAR(N,...)** CHAR() interpreta los argumentos como enteros y retorna la cadena de caracteres que consiste en los caracteres dados por los códigos de tales enteros. Los valores NULL no se tienen en cuenta.

```
SELECT CHAR(77,121,83,81,'76');
```

-> 'MySQL'

- **CONCAT(str1,str2,...)** Retorna la cadena resultado de concatenar los argumentos. Retorna NULL si alguna argumento es NULL. Puede tener uno o más argumentos. Si todos los argumentos son cadenas no binarias, el resultado es una cadena no binaria. Si los argumentos incluyen cualquier cadena binaria, el resultado es una cadena binaria. Un argumento numérico se convierte a su forma de cadena binaria equivalente; si quiere evitarlo puede usar conversión de tipos explícita, como en este ejemplo: SELECT CONCAT(CAST(int_col AS CHAR), char_col)

```
SELECT CONCAT('My', 'S', 'QL');
```

-> 'MySQL'

Funciones de cadena

- **CONCAT_WS(separator, str1, str2, ...)** CONCAT_WS() significa CONCAT With Separator (CONCAT con separador) y es una forma especial de CONCAT(). El primer argumento es el separador para el resto de argumentos. El separador se añade entre las cadenas a concatenar. El separador puede ser una cadena como el resto de argumentos. Si el separador es NULL, el resultado es NULL. La función evita valores NULL tras el argumento separador.

```
SELECT CONCAT_WS(',', 'First name', 'Second name', 'Last Name');
```

```
-> 'First name,Second name,Last Name'
```

Funciones de cadena

- **INSERT(str,pos,len,newstr)** Retorna la cadena str, con la subcadena comenzando en la posición pos y len caracteres reemplazados por la cadena newstr. Retorna la cadena original si pos no está entre la longitud de la cadena. Replaza el resto de la cadena a partir de la posición pos si len no está dentro de la longitud del resto de la cadena. Retorna NULL si cualquier argum

SELECT INSERT('Quadratic', 3, 4, 'What'); -> QuWhattic

Funciones de cadena

- **INSTR(str,substr)** Retorna la posición de la primera ocurrencia de la subcadena substr en la cadena str. Es lo mismo que la forma de dos argumentos de LOCATE(), excepto que el orden de los argumentos es inverso.

```
SELECT INSTR('foobarbar', 'bar');
```

-> 4

Funciones de cadena

- **LCASE(str)** LCASE() es sinónimo de LOWER().
- **LOWER(str)** Retorna la cadena str con todos los caracteres cambiados a minúsculas según el mapeo del conjunto de caracteres actual (por defecto es ISO-8859-1 Latin1).

```
SELECT LOWER('QUADRATICALLY'); -> 'quadratically'
```
- **LENGTH(str)** Retorna la longitud de la cadena str, medida en bytes. Un carácter multi-byte cuenta como múltiples bytes. Esto significa que para cadenas que contengan cinco caracteres de dos bytes, LENGTH() retorna 10, mientras que CHAR_LENGTH() retorna 5.

```
SELECT LENGTH('text');
```

-> 4
- **LEFT(str,len)** Retorna los len caracteres empezando por la izquierda de la cadena str.

```
SELECT LEFT('foobarbar', 5);
```

-> 'fooba'

Funciones de cadena

- **LOAD_FILE(file_name)** Lee el fichero y retorna el contenido como cadena de caracteres. El fichero debe estar localizado en el servidor, debe especificar la ruta completa al fichero, y debe tener el privilegio FILE . El fichero debe ser legible por todo el mundo y su tamaño menor a max_allowed_packet bytes. Si el fichero no existe o no puede ser leído debido a que una de las anteriores condiciones no se cumple, la función retorna NULL.

```
UPDATE tbl_name SET blob_column=LOAD_FILE('/tmp/picture') WHERE id=1;
```

Funciones de cadena

- **LTRIM(str)** Retorna la cadena str con los caracteres en blanco iniciales eliminados.

```
SELECT LTRIM(' barbar');
```

-> 'barbar'

- **MID(str,pos,len)** MID(str,pos,len) es sinónimo de SUBSTRING(str,pos,len).

- **REPEAT(str,count)** Retorna una cadena consistente de la cadena str repetida count veces. Si count <= 0, retorna una cadena vacía. Retorna NULL si str o count son NULL.

```
SELECT REPEAT('MySQL', 3); -> 'MySQLMySQLMySQL'
```

Funciones de cadena

- **REPLACE(str,from_str,to_str)** Retorna la cadena str con todas las ocurrencias de la cadena from_str reemplazadas con la cadena to_str.

```
SELECT REPLACE('www.mysql.com', 'w', 'Ww'); -> 'WwWwWw.mysql.com'
```

Funciones de cadena

- **RIGHT(str,len)** Retorna los len caracteres de la derecha de la cadena str
`SELECT RIGHT('foobarbar', 4); -> 'rbar'`
- **RTRIM(str)** Retorna la cadena str con los espacios precedentes eliminados.
`SELECT RTRIM('barbar '); -> 'barbar'`
- **SPACE(N)** Retorna la cadena consistente en N caracteres blancos.
`SELECT SPACE(6); -> ' '`

Funciones de cadena

- **TRIM([{BOTH | LEADING | TRAILING} [remstr] FROM] str), TRIM(remstr FROM] str)** Retorna la cadena str con todos los prefijos y/o sufijos remstr eliminados. Si ninguno de los especificadores BOTH, LEADING, o se da TRAILING, BOTH se asumen. Si remstr es opcional y no se especifica, los espacios se eliminan.

```
SELECT TRIM(' bar '); -> 'bar'
```