



Tema 1. Funciones en MySQL

Notas

- En 0xE282AC, 0x quiere decir que el número que viene a continuación E282AC está en hexadecimal
- En la mayoría de consultas se ofrece uno o varios ejemplos de uso de la función y algún ejemplo de uso con la base de datos con las que trabajamos en clase: *mundo*.
- *cad* se refiere a una cadena de caracteres, *num* a un número entero
- Las consultas son consultas de ejemplo y no consultas finales por lo que la mayoría no tienen *alias de columna*, indentación correcta, etc.
- Usamos mucho el carácter '€' como ejemplo de carácter multibyte, ya que este carácter se codifica como tres bytes 0xE282AC
- En el resumen no aparecen todas las funciones explicadas, sólo las que usaremos más
- El primer carácter de una cadena es el número 1, el siguiente el 2 y así
- Algunas funciones tienen *sinónimos*. Se suele preferir el nombre original, pero también se pueden usar los sinónimos, por ejemplo: *SUBSTR* es un sinónimo de *SUBSTRING*
- Para una referencia completa de las funciones se debe acudir al manual

Funciones de cadenas de caracteres

Resumen

ORD(cad)	código de un carácter multibyte de la forma: 1 ^{er} byte + 2 ^o byte*256 + 3 ^{er} byte*256 ² + ...
CHAR(num,...)	convierte los números en su carácter correspondiente
CHAR_LENGTH(cad)	devuelve la longitud en caracteres de una cadena
CONCAT(cad1, cad2, ...)	concatena las cadenas
CONCAT_WS(separador, cad1, cad2, ...)	concatena con un separador
HEX(cad)	devuelve la representación en hexadecimal de la cadena
UNHEX(cad)	convierte una representación en hexadecimal en cadena
INSERT(cad, pos, long, subcad)	inserta la cadena subcad en la cadena cad a partir de la posición pos eliminando long caracteres de la cadena cad
LEFT(cad, num)	devuelve los num caracteres más a la izquierda de cad
RIGHT(cad, num)	devuelve los num caracteres más a la derecha de cad
LOWER(cad)	convierte todos los caracteres de cad a minúsculas
UPPER(cad)	convierte todos los caracteres de cad a mayúsculas
LOCATE(subcad, cad)	devuelve la posición de la primera ocurrencia de subcad en cad o 0 si la cadena no es encontrada
LOCATE(subcad, cad, num)	devuelve la posición de la primera ocurrencia de subcad en cad a partir de la posición num o 0 si la cadena no es encontrada
REPLACE(cad, subcad1, subcad2)	devuelve cad con todas las ocurrencias de subcad1 reemplazadas por subcad2
REVERSE(cad)	devuelve cad invertida
SUBSTRING(cad, num)	devuelve una subcadena de cad empezando en la posición num
SUBSTRING(cad, num, long)	devuelve una subcadena de cad de long caracteres empezando en la posición



	num
--	-----

ASCII(cad): devuelve le valor ASCII del carácter más a la izquierda (sólo para caracteres de 1 byte)

```
SELECT ASCII('2'); -- Da 50
SELECT ASCII(LEFT(Codigo2,1)) FROM Pais;
```

ORD(cad): código de un carácter multibyte de la forma: 1^{er} byte + 2^o byte*256 + 3^{er} byte*256² + ...

```
SELECT ORD('€'); -- Da 14844588 '€' son tres bytes: 0xE282AC
SELECT ASCII('€'); -- Da 226 sólo da el primer byte: 0xE2
SELECT ORD('2'); -- Da 50
```

CHAR(num, ...): convierte los números en su carácter correspondiente

Si el N > 256 nos da el carácter multibyte correspondiente

```
SELECT CHAR(77,121,83,81,'76'); -- Da 'MySQL'
SELECT CHAR(50, 14844588); -- Da '2€'
```

LENGTH(cad): devuelve la longitud en bytes de una cadena

```
SELECT LENGTH('MySQL'); -- Da 5
SELECT LENGTH('2€'); -- Da 4 porque '€' es un carácter que ocupa 3 bytes
```

CHAR_LENGTH(cad): devuelve la longitud en caracteres de una cadena

```
SELECT CHAR_LENGTH('MySQL'); -- Da 5
SELECT CHAR_LENGTH('2€'); -- Da 2 aunque € es un caracter multibyte que ocupa 3 bytes
```

CONCAT(cadena1, cadena2, ...): concatena las cadenas

```
SELECT CONCAT('My', 'SQL'); -- da 'MySQL'
SELECT CONCAT(Continente, ' | ', Nombre) FROM Pais;
```

CONCAT_WS(separador, cad1, cad2, ...): concatena con un separador

```
SELECT CONCAT_WS(' <> ', 'Hola', 'mundo', 'cruel'); -- da 'Hola <> mundo <> cruel'
SELECT CONCAT_WS(' | ', Continente, Region, Nombre) FROM Pais;
```

HEX(cad): devuelve la representación en hexadecimal de la cadena

```
SELECT HEX('2€'); -- da 0x32E282AC donde 0x32 es el carácter '2' y 0xE282AC es el carácter '€'
```

UNHEX(cad): convierte una representación en hexadecimal en cadena

```
SELECT UNHEX('32E282AC'); -- da '2€'
```

INSERT(cad, pos, long, subcad): inserta la cadena *subcad* en la cadena *cad* a partir de la posición *pos* eliminando *long* caracteres de la cadena *cad*

```
SELECT INSERT('Hola mundo cruel',6,5,'MUNDO');
SELECT INSERT('Hola cruel',6,0,'mundo ');
```

LEFT(cad, num): devuelve los *num* caracteres más a la izquierda de *cad*

```
SELECT LEFT ('Hola mundo',3); -- da 'Hol'
SELECT CONCAT(LEFT(Codigo2,1),Codigo) FROM Pais;
```

RIGHT(cad, num): devuelve los *num* caracteres más a la derecha de *cad*

```
SELECT RIGHT ('Hola mundo',3); -- da 'ndo'
SELECT CONCAT(LEFT(Codigo2,1),Codigo, RIGHT(Codigo2,1)) FROM Pais;
```



LOWER(cad): convierte todos los caracteres de *cad* a minúsculas

```
SELECT LOWER('Hola mundo'); -- da 'hola mundo'
```

UPPER(cad): convierte todos los caracteres de *cad* a mayúsculas

```
SELECT UPPER('Hola mundo'); -- da 'HOLA MUNDO'
```

LOCATE(subcad, cad): devuelve la posición de la primera ocurrencia de *subcad* en *cad* o 0 si la cadena no es encontrada

```
SELECT LOCATE('mundo', 'Hola mundo cruel');-- da 6
SELECT Nombre,Codigo,Codigo2, LOCATE(Codigo2,Codigo) FROM Pais;
```

LOCATE(subcad, cad, num): devuelve la posición de la primera ocurrencia de *subcad* en *cad* a partir de la posición *num* o 0 si la cadena no es encontrada

```
SELECT LOCATE('u', 'Hola mundo cruel', 10);-- da 14
SELECT Nombre,Codigo,Codigo2, LOCATE(RIGHT(Codigo2,1),Codigo,2) FROM Pais; -- busca la posición
del segundo carácter del Codigo2 en código a partir de su segundo carácter
```

REPLACE(cad, subcad1, subcad2): devuelve *cad* con todas las ocurrencias de *subcad1* reemplazadas por *subcad2*

```
SELECT REPLACE('Hola mundo cruel', 'cruel', 'dulce'); -- da 'Hola mundo dulce'
SELECT REPLACE(Codigo, LEFT(Codigo2,1), '1') FROM Pais; -- sustituye todas las ocurrencias del
primer carácter del Codigo2 de cada país en el Codigo del país por el carácter 1
SELECT REPLACE(REPLACE(Codigo, LEFT(Codigo2,1), '1'), RIGHT(Codigo2,1), '2') FROM Pais; -- sustituye
todas las ocurrencias del primer carácter del Codigo2 de cada país en el Codigo del país por el
carácter 1 y las del segundo carácter del Codigo2 por '2'
```

REVERSE(cad): devuelve *cad* invertida

```
SELECT REVERSE('MySQL'); -- da 'LQSyM'
SELECT REVERSE(Nombre) FROM Pais;
```

SUBSTRING(cad, num): devuelve una subcadena de *cad* empezando en la posición *num*

```
SELECT SUBSTRING('Hola mundo', 6); -- da 'mundo'
SELECT SUBSTRING(Codigo,2) FROM Pais; -- da el segundo y tercer carácter del código del país. Como
el campo Codigo tiene siempre 3 caracteres es equivalente a RIGHT(Codigo,2)
```

SUBSTRING(cad, num, long): devuelve una subcadena de *cad* de long caracteres empezando en la posición *num*

```
SELECT SUBSTRING('Hola mundo', 6, 3); -- da 'mun'
SELECT SUBSTRING(Codigo,2,1) FROM Pais; -- da el segundo carácter del código del país
```

Funciones numéricas

Resumen

ABS(num)	devuelve el valor absoluto de num (lo pasa a positivo)
num1 DIV num2	devuelve la división entera. Es un operador, no una función
PI()	devuelve el número PI
SIN(num)	seno de un ángulo num en radianes
COS(num)	coseno de un ángulo num en radianes
DEGREES(num)	convierte num de radianes a grados
RADIANS(num)	convierte num de grados a radianes



MOD(num1, num2)	devuelve el resto de dividir num1 entre num2
POW(num1, num2)	devuelve el resultado de num1 elevado a num2
RAND()	devuelve un valor aleatorio mayor o igual que 0 y menor que 1
ROUND(num1, num2)	redondea num1 a num2 decimales. Si num2 no existe redondea al entero más cercano.
SQRT(num)	devuelve la raíz cuadrada de num
TRUNCATE(num1, num2)	trunca num1 a num2 decimales.
FLOOR(num)	trunca al entero menor que num
CEILING(num)	trunca al entero mayor que num

ABS(num): devuelve el valor absoluto de *num* (lo pasa a positivo)

```
SELECT ABS(-2); -- da 2
SELECT Nombre, ABS(AnyIndep) FROM Pais; -- Para China saldrá 1523 porque China se independizó en el -1523
```

num1 DIV num2: devuelve la división entera. Es un operador, no una función

```
SELECT 9 DIV 2; -- da 4
SELECT Nombre, PNB*1000000 DIV Poblacion FROM Pais; -- reta per cápita sin decimales
```

PI(): devuelve el número PI

```
SELECT PI(); -- da 3,141593 aunque internamente usa más decimales
```

SIN(num): seno de un ángulo *num* en radianes

```
SELECT SIN(PI()/2); -- da 1. PI()/2 se corresponde con un ángulo de 90 grados
```

COS(num): coseno de un ángulo *num* en radianes

```
SELECT COS(PI()/2); -- da un número que prácticamente es 0: 6,1*10^-17. No da 0 porque el resultado es un número real de doble precisión. PI()/2 se corresponde con un ángulo de 90 grados
```

DEGREES(num): convierte *num* de radianes a grados

```
SELECT DEGREES(PI()/2); -- da 90 (grados)
```

RADIANS(num): convierte *num* de grados a radianes

```
SELECT RADIANS(90); -- da 1.5707963267948966, que es PI()/2
```

MOD(num1, num2): devuelve el resto de dividir *num1* entre *num2*

```
SELECT MOD(10, 3); -- da 1
```

POW(num1, num2): devuelve el resultado de *num1* elevado a *num2*

```
SELECT POW(2,8); -- da 256
```

RAND(): devuelve un valor aleatorio mayor o igual que 0 y menor que 1

```
SELECT RAND(); -- cada vez que ejecutamos este comando nos da un número difenete >= 0 y < 1
SELECT Nombre FROM Pais ORDER BY RAND(); da todos los países cada vez con un orden diferente
SELECT Nombre FROM Pais ORDER BY RAND() LIMIT 5; da 5 países aleatorios
```

ROUND(num1, num2): redondea *num1* a *num2* decimales. Si *num2* no existe redondea al entero más cercano.

```
SELECT ROUND(1.3); -- da 1
SELECT ROUND(1.33,1); -- da 1.3
SELECT ROUND(-1.3); -- da -1
```

Si *mun2* es negativo redondea a las decenas, centenas, etc.



```
SELECT ROUND(1256, -2); -- da 1300
```

SQRT(num): devuelve la raíz cuadrada de *num*

```
SELECT SQRT(81); -- da 9
```

TRUNCATE(num1, num2): trunca *num1* a *num2* decimales.

```
SELECT TRUNCATE(1.7,0); -- da 1
SELECT TRUNCATE(1.3,0); -- da 1
SELECT TRUNCATE(-1.3,0); -- da -1
SELECT TRUNCATE(-1.7,0); -- da -1
```

Valores negativos de *num2* truncan las decenas, centenas, etc.

```
SELECT TRUNCATE(123,-2); -- da 100
```

FLOOR(num): trunca al entero menor que *num*

```
SELECT FLOOR(1.7); -- da 1
SELECT FLOOR(1.3); -- da 1
SELECT FLOOR(-1.3); -- da -2
SELECT FLOOR(-1.7); -- da -2
```

CEILING(num): trunca al entero mayor que *num*

```
SELECT CEILING(1.7); -- da 2
SELECT CEILING(1.3); -- da 2
SELECT CEILING(-1.3); -- da -1
SELECT CEILING(-1.7); -- da -1
```