

## Funciones Matemáticas

[ABS](#) Devuelve el valor absoluto

[CRC32](#) Cálculo de comprobación de redundancia cíclica

[MOD %](#) Resto de una división entera

[PI](#) Valor del número  $\pi$

[POW / POWER](#) Valor de potencias

[DEGREES](#) Conversión de radianes a grados

[RAND](#) Valores aleatorios

[SIGN](#) Devuelve el signo

[SQRT](#) Cálculo de la raíz cuadrada

[RADIANS](#) Conversión de grados a radianes

## Funciones Trigonométricas

[ACOS](#) Devuelve el arcocoseno

[ASIN](#) Devuelve el arcoseno

[ATAN / ATAN2](#): Devuelven el arcotangente

[COS](#) Coseno de un ángulo

[COT](#) Cotangente de un ángulo

[SIN](#) Cálculo del seno de un ángulo

[TAN](#) Cálculo de la tangente de un ángulo

## Funciones “Redondeo”

[CEILING / CEIL](#) Redondeo hacia arriba

[FLOOR](#) Redondeo hacia abajo

[ROUND / ROUND\(X\)](#) Cálculo de redondeos

[TRUNCATE](#) Elimina decimales

## Funciones Logarítmicas”

[LN](#) Logaritmo natural

[LOG](#): Logaritmo en base arbitraria

[LOG10](#) Logaritmo en base 10

[LOG2](#) Logaritmo en base dos

[EXP](#) Cálculo de potencias de e

## Funciones matemáticas

### ABS

ABS(X)

Devuelve el valor absoluto de X:

```
SELECT ABS (2) , ABS (-32) ;
```

	ABS(2)	ABS(-32)
▶	2	32

Esta función es segura con valores *BIGINT*.

### SIGN

SIGN(X)

Devuelve el signo del argumento como -1, 0 ó 1, dependiendo de si X es negativo, cero o positivo:

```
SELECT SIGN (-32) , SIGN (0) , SIGN (234) ;
```

	SIGN(-32)	SIGN(0)	SIGN(234)
▶	-1	0	1

### MOD

MOD(N,M)

%

Módulo (como el operador % en C). Devuelve el resto de la división de N entre M:

```
SELECT MOD (234 , 10) , 253 % 7 , MOD (29,9) ;
```

	MOD(234,10)	253 % 7	MOD(29,9)
▶	4	1	2

Esta función es segura con valores *BIGINT*.

### POW / POWER

POW(X,Y)

POWER(X,Y)

Devuelve el valor de X elevado a la potencia Y:

```
SELECT POW (2 ,2) , POW (2 , -2) , POWER (5 ,3) ;
```

	POW(2,2)	POW(2,-2)	POWER(5,3)
▶	4	0.25	125

## RAND

RAND()

RAND(N)

Devuelve un valor aleatorio en punto flotante, en el rango 0 a 1.0. Si se especifica un argumento entero N, se usa como valor de semilla (produciendo una secuencia repetible):

**SELECT RAND() , RAND(20) , RAND(25) ;**

	RAND()	RAND(20)	RAND(25)
▶	0.6516950378...	0.1588826125...	0.4097981596...

**SELECT RAND() , RAND(20) , RAND(25) ;**

	RAND()	RAND(20)	RAND(25)
▶	0.9841801822...	0.1588826125...	0.4097981596...

No es posible usar una columna con valores RAND() en una cláusula ORDER BY, porque ORDER BY puede evaluar la columna varias veces. Desde la versión 3.23 se puede hacer: SELECT \* FROM table\_name ORDER BY RAND(). Lo siguiente es corriente para obtener una muestra aleatoria de un conjunto SELECT \* FROM table1,table2 WHERE a=b AND c<d ORDER BY RAND() LIMIT 1000. Hay que tener en cuenta que RAND() en una cláusula WHERE será evaluada cada vez que el WHERE sea ejecutado. RAND() no está diseñado para ser un generador aleatorio perfecto, pero sin embargo es un modo rápido de generar números aleatorios que pueden ser portados entre plataformas por la misma versión de MySQL.

## SQRT

SQRT(X)

Devuelve la raíz cuadrada no negativa de X:

**SELECT SQRT(4) , SQRT(20) , SQRT(-5) ;**

	SQRT(4)	SQRT(20)	SQRT(-5)
▶	2	4.4721359549...	NULL

## CRC32

### CRC32(expr)

Calcula el valor de comprobación de redundancia cíclica y devuelve un valor entero sin signo de 32 bits. El resultado es NULL si el argumento es NULL. El argumento esperado es una cadena y será tratado como tal si no lo es.

```
SELECT CRC32 ('MySQL' ) ;
```

CRC32('MySQL')	CRC32(125)
3259397556	1630258919

## DEGREES

### DEGREES(X)

Devuelve el argumento X, convertido de radianes a grados:

```
SELECT DEGREES (PI ( ) ) ;
```

DEGREES(PI())
180

## PI

### PI()

Devuelve el valor de  $\pi$ . Por defecto se obtienen 5 decimales, pero MySQL usa internamente un valor completo de doble precisión.

```
SELECT PI ( ) , PI ( ) +0.000000000000000000 ;
```

PI()	PI()+0.000000000000000000
3.141593	3.141592653589793000

## RADIANS

### RADIANS(X)

Devuelve el argumento X, convertido de grados a radianes:

```
SELECT RADIANS (90) ;
```

RADIANS(90)
1.5707963267948966

## Funciones "REDONDEO"

### CEILING / CEIL

CEILING(X)

CEIL(X)

Devuelve el entero más pequeño cuyo valor es mayor que X:

```
SELECT CEILING(1.23) , CEIL(-1.23) , CEIL(1.99) ;
```

	CEILING(1.23)	CEIL(-1.23)	CEIL(1.99)
▶	2	-1	2

El alias CEIL() fue añadido en la versión 4.0.6. El valor de retorno se convierte a BIGINT!

### FLOOR

FLOOR(X)

Devuelve el entero más grande inferior o igual a X:

```
SELECT FLOOR(1.23) , FLOOR(-1.23) , FLOOR(1.99) ;
```

	FLOOR(1.23)	FLOOR(-1.23)	FLOOR(1.99)
▶	1	-2	1

El valor de retorno se convierte a BIGINT.

### ROUND / ROUND(X)

ROUND(X,D)

Devuelve el argumento X, redondeado al entero más cercano. Con dos argumentos redondea a un número con D decimales.

```
SELECT ROUND(-1.23) , ROUND(-1.58) , (1.58) , ROUND(1.298, 1) ,  
ROUND(1.298, 0) , ROUND(23.298, -1) ;
```

	ROUND(-1.23)	ROUND(-1.58)	1.58	ROUND(1.298,...	ROUND(1.298,...	ROUND(23.29...
▶	-1	-2	1.58	1.3	1	20

El comportamiento de ROUND() cuando el argumento está justo en la mitad de dos enteros depende de la implementación de la librería C. Algunos redondean al número impar más cercano, otros hacia arriba, hacia abajo, o hacia cero. Si se necesita un tipo de redondeo, se puede usar la función TRUNCATE() o FLOOR() en su lugar.

### TRUNCATE

TRUNCATE(X,D)

Devuelve el número X, truncado a D decimales. Si D es 0, el resultado no tendrá punto decimal o parte fraccionaria:

```
SELECT TRUNCATE(1.223,1) , TRUNCATE(1.999,1) , TRUNCATE(1.999,0) ,
TRUNCATE(-1.999,1) ;
```

	TRUNCATE(1....	TRUNCATE(1....	TRUNCATE(1....	TRUNCATE(-1....
►	1.2	1.9	1	-1.9

Desde MySQL 3.23.51, todos los números se redondean hacia cero. Si D es negativo, entonces la parte correspondiente del número es asignada a cero:

```
SELECT TRUNCATE(122,-2) ;
```

	TRUNCATE(12...
►	100

Como los números decimales normalmente no se almacenan como números exactos en los ordenadores, sino como valores de doble precisión, el siguiente resultado puede parecer sorprendente:

```
SELECT TRUNCATE(10.28*100,0) ;
```

	TRUNCATE(10...
►	1028

Lo anterior sucede porque 10.28 se almacena como algo parecido a 10.2799999999999999.

## Funciones TRIGONOMÉTRICAS

### ACOS

#### ACOS(X)

Devuelve el arcocoseno de X, es decir, el valor del arco cuyo coseno es X. Devuelve NULL si X no está en el rango de -1 a 1:

```
SELECT ACOS(1) , ACOS(1.0001) , ACOS(0) ;
```

	ACOS(1)	ACOS(1.0001)	ACOS(0)
►	0	NULL	1.5707963267...

## ASIN

### ASIN(X)

Devuelve el arcoseno de X, es decir, el valor del arco cuyo seno es X. Devuelve NULL si X no está en el rango de -1 a 1:

```
SELECT ASIN(0.2) , ASIN('foo') ;
```

	ASIN(0.2)	ASIN('foo')
▶	0.2013579207...	0

## ATAN / ATAN2

### ATAN(X)

### ATAN(Y,X)

### ATAN2(Y,X)

Devuelve el arcotangente de X, es decir, el valor del arco cuya tangente es X:

```
SELECT ATAN(2) , ATAN(-2) ;
```

	ATAN(2)	ATAN(-2)
▶	1.1071487177...	-1.1071487177...

Con dos argumentos devuelve el arcotangente de las dos variables X e Y. Esto es similar a calcular el arcotangente de Y / X, excepto que los signos de ambos argumentos se tienen en cuenta para determinar el cuadrante del resultado:

```
SELECT ATAN(-2, 2) , ATAN2(PI(),-1) ;
```

	ATAN(-2, 2)	ATAN2(PI(),-1)
▶	-0.7853981633...	1.8789653979...

## COS

### COS(X)

Devuelve el coseno de X, donde X viene dado en radianes:

```
SELECT COS(PI()) ;
```

COS(PI())
-1

## COT

### COT(X)

Devuelve la cotangente de X:

```
SELECT COT(12) ;
```

COT(12)	COT(-1)
-1.5726734063...	-0.6420926159...

## SIN

### SIN(X)

Devuelve el seno de X, donde X viene dado en radianes:

```
SELECT SIN(PI()) ;
```

SIN(PI())
1.2246467991...

## TAN

### TAN(X)

Devuelve la tangente de X, donde X viene dado en radianes:

```
SELECT TAN(PI()+1) ;
```

TAN(PI()+1)
1.5574077246...

## Funciones LOGARÍTMICAS

## EXP

### EXP(X)

Devuelve el valor del número e (la base de los logaritmos naturales) elevado a la potencia X:

```
SELECT EXP(2) , EXP(-2) ;
```



	EXP(2)	EXP(-2)
▶	7.3890560989...	0.1353352832...

## LN

### LN(X)

Devuelve el logaritmo natural de X:

```
SELECT LN(2) , LN(-2) ;
```

	LN(2)	LN(-2)
▶	0.6931471805...	NULL

Esta función se añadió en MySQL 4.0.3. Es sinónimo de LOG(X) en MySQL.

## LOG

### LOG(X)

### LOG(B,X)

Si se llama con un parámetro, esta función devuelve el logaritmo natural de X:

```
SELECT LOG(2) , LOG(-2) ;
```

	LOG(2)	LOG(-2)
▶	0.6931471805...	NULL

Si se llama con dos parámetros, devuelve el logaritmo de X para una base arbitraria B:

```
SELECT LOG(2,65536) , LOG(1,100) ;
```

	LOG(2,65536)	LOG(1,100)
▶	16	NULL

La opción de base arbitraria se añadió en MySQL 4.0.3. LOG(B,X) equivale a LOG(X)/LOG(B).

## LOG10

### LOG10(X)

Devuelve el logaritmo en base 10 de X:

```
SELECT LOG10(2) , LOG10(100) , LOG10(-100) ;
```

	LOG10(2)	LOG10(100)	LOG10(-100)
▶	0.3010299956...	2	NULL

## LOG2

### LOG2(X)

Devuelve el logaritmo en base 2 de X:

```
SELECT LOG2 (65536) , LOG2 (-100) ;
```

	LOG2(65536)	LOG2(-100)
▶	16	NULL

**LOG2()** es corriente para calcular cuántos bits son necesarios para almacenar un número.