

Funciones matemáticas

Las funciones matemáticas en MySQL son un conjunto de herramientas que permiten realizar cálculos matemáticos en las consultas SQL. Estas funciones son útiles para realizar operaciones como cálculos de valores absolutos, funciones trigonométricas, logaritmos, exponenciación y más, directamente en las consultas de la base de datos.

El buen conocimiento de las mismas te permitirá cubrir 2 objetivos:

- **Exploración de Funciones Matemáticas Básicas y Avanzadas:** Te sumergirás en el repertorio de funciones matemáticas proporcionadas por MySQL, logrando comprender de manera detallada cómo aplicar estas funciones para realizar cálculos precisos.
- **Aplicación de Funciones Matemáticas en Consultas:** Aprenderás a integrar funciones matemáticas directamente en tus consultas SQL, permitiéndote realizar cálculos específicos y avanzados en el contexto de tu base de datos.

Aquí tienes una tabla con todas las funciones y una breve descripción:

Función	Descripción
ABS()	Devuelve el valor absoluto
ACOS()	Devuelve el arcocoseno
ASIN()	Devuelve el arcoseno
ATAN()	Devuelve el arcotangente
ATAN2(), ATAN()	Devuelve el arcotangente de los dos argumentos

CEIL()	Devuelve el valor entero más pequeño no menor que el argumento
CEILING()	Devuelve el valor entero más pequeño no menor que el argumento
CONV()	Convierte números entre diferentes bases numéricas
COS()	Devuelve el coseno
COT()	Devuelve la cotangente
CRC32()	Calcula un valor de comprobación de redundancia cíclica
DEGREES()	Convierte radianes a grados
EXP()	Eleva a la potencia de
FLOOR()	Devuelve el valor entero más grande no mayor que el argumento
LN()	Devuelve el logaritmo natural del argumento
LOG()	Devuelve el logaritmo natural del primer argumento
LOG10()	Devuelve el logaritmo en base 10 del argumento
LOG2()	Devuelve el logaritmo en base 2 del argumento
MOD()	Devuelve el residuo
PI()	Devuelve el valor de pi
POW()	Devuelve el argumento elevado a la potencia especificada
POWER()	Devuelve el argumento elevado a la potencia especificada
RADIANS()	Devuelve el argumento convertido a radianes

RAND()	Devuelve un valor de punto flotante aleatorio
ROUND()	Redondea el argumento
SIGN()	Devuelve el signo del argumento
SIN()	Devuelve el seno del argumento
SQRT()	Devuelve la raíz cuadrada del argumento
TAN()	Devuelve la tangente del argumento
TRUNCATE()	Trunca a un número especificado de decimales

📌 Todas las funciones matemáticas retornan NULL en caso de error.

ABS(X)

Devuelve el valor absoluto de X, o NULL si X es NULL.

El tipo de resultado se deriva del tipo del argumento. Una implicación de esto es que `ABS(-9223372036854775808)` produce un error porque el resultado no puede almacenarse en un valor BIGINT con signo.

ACOS(X)

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

ASIN(X)

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

ATAN(X)

Devuelve el arcotangente de X, es decir, el valor cuya tangente es X. Retorna NULL si X es NULL.

ATAN(Y, X), ATAN2(Y, X)

Devuelve el arcotangente de los dos valores X e Y. Es similar al cálculo del arcotangente de Y / X , excepto que se utilizan los signos de ambos argumentos para determinar el cuadrante del resultado. Retorna NULL si X o Y son NULL.

CEIL(X) y CEILING(X)

Devuelve el valor entero redondeado X hacia abajo. Retorna NULL si X es NULL.

CONV(N, from_base, to_base)

Convierte números entre diferentes bases numéricas. Retorna una representación de cadena del número N, convertido de la base from_base a la base to_base. Retorna NULL si alguno de los argumentos es NULL. El argumento N se interpreta como un entero, pero puede especificarse como un entero o una cadena. La base mínima es 2 y la base máxima es 36. Si from_base es un número negativo, N se considera como un número con signo. De lo contrario, N se trata como sin signo. CONV() trabaja con precisión de 64 bits.

CONV() retorna NULL si alguno de sus argumentos es NULL.

Por ejemplo, convertir el número decimal 10 a binario (base 10 a base 2):

```
SELECT CONV(10, 10, 2) AS binario;  
# Resultado: '1010'
```

Convertir el número binario '1101' a decimal (base 2 a base 10):

```
SELECT CONV('1101', 2, 10) AS decimal;  
# Resultado: 13
```

COS(X)

Devuelve el coseno de X, donde X se proporciona en radianes. Retorna NULL si X es NULL.

COT(X)

Devuelve la cotangente de X. Retorna NULL si X es NULL.

CRC32(expr)

Calcula un valor de comprobación de redundancia cíclica y retorna un valor sin signo de 32 bits. El resultado es NULL si el argumento es NULL. Se espera que el argumento sea una cadena y (si es posible) se trata como tal si no lo es.

Ejemplos:

```
SELECT CRC32('MySQL') AS crc_valor;
-- Resultado: 3259397556

SELECT CRC32('Ejemplo de cadena') AS crc_valor;
-- Resultado: 3618981295

SELECT CRC32('') AS crc_valor;
-- Resultado: 0
```

💡 El "valor de comprobación de redundancia cíclica" (CRC) es un valor numérico que se calcula a partir de una secuencia de datos, generalmente una cadena de bits o caracteres, utilizando un algoritmo matemático. El propósito principal de un CRC es detectar cambios accidentales o errores en la secuencia de datos durante la transmisión o el almacenamiento. Cuando se envían datos, se adjunta su valor CRC correspondiente. El receptor puede calcular su propio valor CRC a partir de los datos recibidos y compararlo con el valor CRC recibido. Si los valores no coinciden, es una indicación de que los datos podrían haber sido alterados durante la transmisión o el almacenamiento, y se requieren medidas para corregir o solicitar nuevamente los datos.

DEGREES(X)

Retorna el argumento X convertido de radianes a grados. Retorna NULL si X es NULL.

```
SELECT DEGREES(1.0472) AS angulo_en_grados;
-- Resultado: 60.00000000114981
```

EXP(X)

Retorna el valor de e (la base de los logaritmos naturales) elevado a la potencia de X. La función inversa de esta función es LOG() (usando un solo argumento) o LN(). Si X es NULL, esta función retorna NULL.

FLOOR(X)

Devuelve el valor entero de X redondeando hacia arriba. Retorna NULL si X es NULL.

FORMAT(X, D)

Formatea el número X en un formato como '#,###,###.##', redondeado a D decimales, y retorna el resultado como una cadena.

HEX(N_or_S)

Esta función se puede usar para obtener una representación hexadecimal de un número decimal o una cadena; la manera en que lo hace varía según el tipo del argumento.

LN(X)

Retorna el logaritmo natural de X; es decir, el logaritmo en base-e de X. Si X es menor o igual a 0.0E0, la función retorna NULL y genera una advertencia de "Argumento no válido para el logaritmo". Retorna NULL si X es NULL. Esta función es sinónimo de LOG(X). La función inversa de esta función es la función EXP().

LOG(X), LOG(B, X)

Si se llama con un solo parámetro, esta función retorna el logaritmo natural de X. Si X es menor o igual a 0.0E0, la función retorna NULL y genera una advertencia de "Argumento no válido para el logaritmo". Retorna NULL si X o B son NULL. La función inversa de esta función (cuando se llama con un solo argumento) es la función EXP().

Si se llama con dos parámetros, esta función retorna el logaritmo de X en la base B. Si X es menor o igual a 0, o si B es menor o igual a 1, entonces se retorna NULL. LOG(B, X) es equivalente a LOG(X) / LOG(B).

LOG2(X)

Retorna el logaritmo en base 2 de X. Si X es menor o igual a 0.0E0, la función retorna NULL y genera una advertencia de "Argumento no válido para el logaritmo". Retorna NULL si X es NULL. Esta función es útil para determinar cuántos

bits requiere un número para su almacenamiento. La función es equivalente a la expresión $\text{LOG}(X) / \text{LOG}(2)$.

LOG10(X)

Retorna el logaritmo en base 10 de X. Si X es menor o igual a 0.0E0, la función retorna NULL y genera una advertencia de "Argumento no válido para el logaritmo". Retorna NULL si X es NULL. LOG10(X) es equivalente a LOG(10, X).

MOD(N, M), N % M, N MOD M

Operación de módulo. Retorna el residuo de N dividido por M. Retorna NULL si M o N son NULL. MOD(N, 0) retorna NULL. Esta función es segura para su uso con valores BIGINT.

PI()

Retorna el valor de π (pi). El número predeterminado de lugares decimales mostrados es siete, pero MySQL utiliza el valor de doble precisión completo internamente.

POW(X, Y) y POWER(X, Y)

Retorna el valor de X elevado a la potencia de Y. Retorna NULL si X o Y son NULL.

RADIANS(X)

Retorna el argumento X, convertido de grados a radianes. (Tenga en cuenta que π radianes equivalen a 180 grados.) Retorna NULL si X es NULL.

RAND([N])

Retorna un valor de punto flotante aleatorio v en el rango $0 \leq v < 1.0$. Para obtener un número entero aleatorio R en el rango $i \leq R < j$, se utiliza la expresión $\text{FLOOR}(i + \text{RAND}() * (j - i))$. Si se especifica un argumento entero N, se usa como valor inicial.

ROUND(X), ROUND(X, D)

Redondea el argumento X a D lugares decimales. El algoritmo de redondeo depende del tipo de datos de X. D se establece en 0 si no se especifica. D puede ser negativo para hacer que D dígitos a la izquierda del punto decimal del valor X se vuelvan cero. El valor absoluto máximo de D es 30; se truncan los dígitos en exceso de 30 (o -30). Si X o D son NULL, la función retorna NULL.

SIGN(X)

Retorna el signo del argumento como -1, 0 o 1, dependiendo de si X es negativo, cero o positivo. Retorna NULL si X es NULL.

SIN(X)

Retorna el seno de X, donde X se proporciona en radianes. Retorna NULL si X es NULL.

SQRT(X)

Retorna la raíz cuadrada de un número no negativo X. Si X es NULL, la función retorna NULL.

TAN(X)

Retorna la tangente de X, donde X se proporciona en radianes. Retorna NULL si X es NULL.

TRUNCATE(X, D)

Retorna el número X truncado a D lugares decimales. Si D es 0, el resultado no tiene punto decimal ni parte fraccional. D puede ser negativo para hacer que D dígitos a la izquierda del punto decimal del valor X se vuelvan cero. Si X o D son NULL, la función retorna NULL. Todos los números se redondean hacia cero. En MySQL 8.0.21 y posteriores, el tipo de datos devuelto por TRUNCATE() sigue las mismas reglas que determinan el tipo de retorno de la función ROUND().
