

Funciones más importantes en SQL para MySQL

El cuadro siguiente presenta las funciones más importantes de SQL que se utilizan en MySQL clasificadas por categorías (la mayoría se usan también en otras variedades de SQL).

<i>Categoría</i>	<i>Explicación y tipos</i>	<i>Ejemplo</i>
<i>Matemáticas</i>	<p>Realizan un cálculo sobre valores de entrada proporcionados como parámetros de la función y devuelven un valor numérico.</p> <p>ABS(x). Valor absoluto de la expresión numérica x. ACOS(x). Arco coseno del real x (en radianes). ASIN(x). Arco seno del real x (en radianes). ATAN(x). Arco tangente del real x (en radianes). ATN2(x,y). Arco tangente de un valor entre x e y. SIN(x). Seno del ángulo x en radianes. COS(x). Coseno del ángulo x en radianes. TAN(x). Tangente del ángulo x en radianes. COT(x). Cotangente del ángulo x en radianes. DEGREES(x). Pasa x radianes a grados. RADIANS(x). Pasa x grados a radianes. EXP(x). Da e^x. PI(). Da el número $\pi = 3,14159...$ LOG(x). Logaritmo neperiano de x. LOG10(x). Logaritmo decimal de x. POWER(x,y). Da x^y. RAND(). Número aleatorio entre 0 y 1. RAND(x). Número aleatorio entre 0 y 1 y semilla x. SQRT(x). Raíz cuadrada de x. TRUNCATE(x,d). Da x truncado a d decimales. CEILING(x). Menor entero mayor o igual que x. FLOOR(x). Mayor entero menor o igual que x. ROUND(x,l). Redondeo de x a la longitud l. SIGN(x). Signo de x (+, - ó 0).</p>	<p>ABS(-1.0) ACOS(-1) ASIN(-1) ATAN(-45.01) ATAN2(35,1.129,3) SIN(45.01) COS(45.01) TAN(45.01) COT(45.01) DEGREES(5.01) RADIANS(-45.01) EXP(1) PI() LOG(12) LOG10(12) POWER(5,25) RAND() RAND(2) SQRT(12) TRUNCATE(1.22234,3) CEILING(45.543) FLOOR(45.01) ROUND(45.01) SIGN(-45.01)</p>
<i>Categoría</i>	<i>Explicación y tipos</i>	<i>Ejemplo</i>
<i>Fecha y hora</i>	<p>Realizan una operación sobre un valor de entrada de fecha u hora y devuelven un valor de cadena, numérico o de fecha y hora.</p> <p>DAYNAME(fecha). Devuelve una cadena con el día de la semana relativo a la fecha.</p> <p>DAYOFMONTH(fecha). Devuelve un número entre 1 y 31 relativo al día del mes para la fecha.</p> <p>DAYOFYEAR(fecha). Devuelve un número entre 1 y 366 relativo al día del año para la fecha.</p> <p>DAYOFWEEK(fecha). Devuelve un número entre 1 y 7 relativo al día de la semana para la fecha.</p>	<p>DAYNAME("1999-12-01") DAYOFMONTH("1999-12-01") DAYOFYEAR("1999-12-01") DAYOFWEEK("1999-12-01")</p>

<i>Categoría</i>	<i>Explicación y tipos</i>	<i>Ejemplo</i>
<i>Fecha y hora</i>	HOUR(hora) . Devuelve un entero entre 0 y 23 que representa la parte entera de la hora especificada.	HOUR ("12:31:58")
	MINUTE(hora) . Devuelve un entero entre 0 y 59 que representa los minutos de la hora especificada.	MINUTE ("12:31:58")
	SECOND(hora) . Devuelve un entero entre 0 y 59 que representa los segundos de la hora especificada.	SEOND ("12:31:58")
	MONTH(fecha) . Devuelve un entero entre 1 y 12 que representa el mes de la fecha especificada.	MONTH ("1999-12-01")
	MONTHNAME(fecha) . Devuelve una cadena con el nombre del mes de la fecha especificada.	MONTHNAME ("1999-12-01")
	QUARTER(fecha) . Devuelve un entero entre 1 y 4 que representa el trimestre de la fecha dada.	QUARTER ("1999-12-01")
	WEEK(fecha) . Devuelve un entero entre 0 y 52 que representa la semana de la fecha dada (las semanas empiezan en domingo).	WEEK ("1999-12-01")
	WEEK(fecha,n) . Devuelve un entero entre 0 y 52 que representa la semana de la fecha dada (las semanas empiezan en domingo si n=0 y empiezan en lunes si n=1).	WEEK ("1999-12-01",1)
	WEEKDAY(fecha) . Devuelve un entero entre 0 (lunes) y 6 (domingo) que representa el día de la semana de la fecha dada.	WEEKDAY ("1999-12-01")
	YEAR(fecha) . Devuelve un entero entre 1000 y 9999 que representa el año de la fecha dada.	YEAR ("1999-12-01")
	TO_DAYS(fecha) . Devuelve un entero que da el número de días entre el año cero y la fecha dada.	TO_DAYS ("1999-12-01")
	NOW() o SYSDATE() . Da una cadena con la fecha y la hora actual en formato AAAAMMDDhhmmss.	
	CURDATE() o CURRENT_DATE() . Da una cadena con la fecha actual en formato AAAA-MM-DD.	
	CURTIME() o CURRENT_TIME() . Da una cadena con la hora actual en formato hhmmss.	
	DATE_FORMAT(fecha,formato) . Devuelve la fecha en el formato dado. Los formatos posibles son. %S, %s, %H, %h, %i, %l, %k, %l, %T, %r, %p, %W, %a, %d, %D, %e, %w, %j, %U, %u, %M, %m, %b, %c, %y, %Y, %%. FROM_DAYS(n) . Devuelve la fecha relativa al entero n (número de días desde el año 0).	DATE_FORMAT ("1999-12-01") FROM_DAYS (730454 + 3)

<i>Categoría</i>	<i>Explicación y tipos</i>	<i>Ejemplo</i>
<i>Fecha y hora</i>	<p>TO_DAYS(<i>fecha</i>). Devuelve el entero <i>n</i> (número de días desde el año 0) correspondiente a la fecha.</p> <p>DATE_ADD(<i>fecha</i>, INTERVAL <i>n tiempo</i>) o ADDDATE(<i>fecha</i>, INTERVAL <i>n tiempo</i>). Añade <i>n</i> unidades de tiempo a la fecha dada. Los posibles valores de <i>tiempo</i> son, SECOND, MINUTE, HOUR, DAY, MONTH, YEAR, MINUTE_SECOND, HOUR_MINUTE, HOUR_SECOND, DAY_HOUR, DAY_MINUTE, DAY_SECOND, YEAR_MONTH.</p> <p>DATE_SUB(<i>fecha</i>, INTERVAL <i>n tiempo</i>) o SUBDATE(<i>fecha</i>, INTERVAL <i>n tiempo</i>). Resta <i>n</i> unidades de tiempo a la fecha dada. Los posibles valores de <i>tiempo</i> son los anteriores.</p> <p>EXTRACT(<i>tiempo FROM fechahora</i>). Da la parte de la <i>fecha hora</i> especificada en <i>tiempo</i>. Los posibles valores de <i>tiempo</i> son los anteriores.</p> <p>PERIOD_ADD(<i>periodo</i>, <i>n</i>). Añade <i>n</i> meses al valor <i>periodo</i>.</p> <p>PERIOD_DIFF(<i>periodo1</i>, <i>periodo2</i>). Halla la diferencia entre los dos periodos.</p> <p>SEC_TO_TIME(<i>segundos</i>). Devuelve la hora relativa al número de segundos dado.</p> <p>TIME_TO_SEC(<i>hora</i>). Devuelve el número de segundos relativo a la hora dada.</p> <p>TIME_FORMAT(<i>hora</i>, <i>formato</i>). Devuelve la hora en el formato dado. Los formatos posibles son los mismos que para DATE_FORMAT.</p>	<p>TO_DAYS ("1999-12-01")</p> <p>DATE_ADD ("1999-12-31", INTERVAL 2 MONTH)</p> <p>DATE_SUB ("1999-12-31", INTERVAL 2 MONTH)</p> <p>EXTRACT (YEAR FROM "1999-12-01 13:42:19")</p> <p>PERIOD_ADD (9902, -3)</p> <p>PERIOD_DIFF (199811, 9902)</p> <p>SEC_TO_TIME (29834)</p> <p>TYME_TO_SEC ("12:31:58")</p> <p>TYME_FORMAT ("12:31:58", "%H %i")</p>
<i>Categoría</i>	<i>Explicación y tipos</i>	<i>Ejemplo</i>
<i>Cadena</i>	<p>ASCII(<i>cadena</i>). Devuelve el código ASCII del carácter más a la izquierda de la cadena.</p> <p>CHAR(<i>n</i>). Devuelve el carácter cuyo código ASCII es <i>n</i>.</p> <p>CHAR(<i>n,m...</i>). Devuelve y concatena los caracteres de códigos ASCII <i>n, m...</i></p> <p>BIN(<i>n</i>). Devuelve <i>n</i> en forma binaria.</p> <p>COALESCE(<i>exp1,exp2...</i>). Devuelve el primer elemento no nulo de la lista de expresiones.</p> <p>CONCAT(<i>cadena1,cadena2...</i>). Concatena las cadenas dadas.</p>	<p>ASCII ("abc")</p> <p>CHAR (97)</p> <p>CHAR (89, 105, 107, 101)</p> <p>BIN (97)</p> <p>COALESCE (NULL, 1/0, 2, "a")</p> <p>CONCAT ("abc", "def")</p>

<i>Categoría</i>	<i>Explicación y tipos</i>	<i>Ejemplo</i>
Cadena	<p>ELT(<i>n</i>,<i>cadena1</i>,<i>cadena2</i>...). Devuelve la cadena <i>enésima</i> de la lista.</p> <p>FIELD(<i>cad</i>,<i>cadena1</i>,<i>cadena2</i>...). Busca la cadena <i>cad</i> en la lista de cadenas y devuelve la posición de su primer elemento.</p> <p>CONV(<i>n</i>,<i>base1</i>,<i>base2</i>). Convierte el número <i>n</i> de <i>base1</i> a <i>base2</i>.</p> <p>FORMAT(<i>n</i>,<i>d</i>). Formatea el número <i>n</i> en una cadena de <i>d</i> posiciones decimales tipo <i>mm,mmn.nnn</i>.</p> <p>HEX(<i>n</i>). Da el número <i>n</i> en hexadecimal.</p> <p>OCT(<i>n</i>). Da el número <i>n</i> en octal.</p> <p>INSERT(<i>cadena</i>,<i>posición</i>,<i>l</i>,<i>subcadena</i>). Se inserta en la <i>cadena</i>, a partir de la <i>posición</i> dada, los primeros <i>l</i> caracteres de la <i>subcadena</i>.</p> <p>LCASE(<i>cadena</i>) o LOWER(<i>cadena</i>). Da la cadena en minúsculas.</p> <p>UCASE(<i>cadena</i>) o UPPER(<i>cadena</i>). Da la cadena en mayúsculas.</p> <p>LEFT(<i>cadena</i>,<i>n</i>). Da los <i>n</i> elementos más a la izquierda de la <i>cadena</i>.</p> <p>RIGHT(<i>cadena</i>,<i>n</i>). Da los <i>n</i> elementos más a la derecha de la <i>cadena</i>.</p> <p>LOCATE(<i>subcadena</i>, <i>cadena</i>, <i>n</i>). Da la posición de la primera ocurrencia de la subcadena en la cadena empezando a buscar a partir de la posición <i>n</i> en la cadena. Si <i>n</i> no se especifica, se empieza a buscar en la primera posición de la cadena.</p> <p>LENGTH(<i>cadena</i>). Devuelve el número de caracteres en la cadena.</p> <p>LPAD(<i>cadena</i>,<i>l</i>,<i>cadena1</i>). Añade <i>cadena1</i> a la izquierda de la cadena las veces que sea necesario hasta que la cadena resultante tenga <i>l</i> caracteres.</p> <p>LTRIM(<i>cadena</i>). Devuelve la cadena después de quitar los espacios en blanco a la izquierda.</p> <p>MID(<i>cadena</i>,<i>posición</i>,<i>n</i>). Da la subcadena de la cadena que comienza en la posición dada y que tiene una longitud <i>n</i>.</p>	<p>ELT(3,"a","d","abc","de")</p> <p>FIELD("a","d","abc","a")</p> <p>CONV(11111111,2,8)</p> <p>FORMAT(1234.56789,3)</p> <p>HEX(1234561)</p> <p>OCT(1234561)</p> <p>INSERT("abcdefgh",3,4,"xyztsp")</p> <p>LCASE("abcdefgh")</p> <p>UCASE("ABCDEFGH")</p> <p>LEFT("abcdefgh",4)</p> <p>RIGHT("abcdefgh",4)</p> <p>LOCATE("d","adbcd",4)</p> <p>LENGTH("abcdefgh")</p> <p>LPAD("xy","adbcd",14)</p> <p>LTRIM(" abcdefgh")</p> <p>MID("abcdefgh",3,4)</p>

<i>Categoría</i>	<i>Explicación y tipos</i>	<i>Ejemplo</i>
<i>Cadena</i>	<p>POSITION(subcadena, cadena). Da la posición de la primera ocurrencia de la subcadena en la cadena empezando a buscar a partir de la primera posición de la cadena. Es equivalente a LOCATE(subcadena, cadena).</p> <p>REPEAT(cadena,n). Repite la cadena <i>n</i> veces.</p> <p>REPLACE(cad1, cad2, cad3). Reemplaza por la tercera cadena todas las apariciones de la segunda cadena en la primera cadena.</p> <p>REVERSE (cadena). Devuelve la cadena invertida.</p> <p>RIGHT(cadena, n). Devuelve los <i>n</i> caracteres más a la derecha de la cadena.</p> <p>RPAD(cadena,l,cadena1). Añade <i>cadena1</i> a la derecha de la cadena las veces que sea necesario hasta que la cadena resultante tenga <i>l</i> caracteres.</p> <p>RTRIM(cadena). Devuelve una cadena de caracteres después de trincar todos los espacios en blanco a la derecha.</p> <p>SOUNDEX(cadena). Devuelve la cadena tipo SOUNDEX relativa a la cadena dada.</p> <p>SPACE(n). Devuelve una cadena de espacios en blanco repetidos <i>n</i> veces.</p> <p>SUBSTRING(cadena, n). Devuelve la parte de la cadena que va desde la posición <i>n</i> hasta su final.</p> <p>SUBSTRING(cadena, n, m). Devuelve <i>m</i> caracteres de la cadena a partir del enésimo.</p>	<p>POSITION ("d", "adbcd") LOCATE ("d", "adbcd")</p> <p>REPEAT ("a", 10)</p> <p>REPLACE ("adbde", "d", "oh")</p> <p>REPVERSE ("adbcd")</p> <p>RIGHT ("adbcd")</p> <p>RPAD ("xy", "adbcd",14)</p> <p>RTRIM ("adbcd")</p> <p>SOUNDEX ("abcdefgh")</p> <p>SPACE (6)</p> <p>SUBSTRING ("abcdefgh",4)</p> <p>SUBSTRING ("abcdefgh",4,3)</p>
<i>Categoría</i>	<i>Explicación y tipos</i>	<i>Ejemplo</i>
<i>Funciones de agregado</i>	<p>Las funciones de agregado realizan un cálculo sobre un conjunto de valores y devuelven un solo valor. Con la excepción de COUNT, las funciones de agregado omiten los valores NULL. Las funciones de agregado se suelen utilizar con la cláusula GROUP BY de la instrucción SELECT. Las funciones de agregado sólo se aceptan como expresiones en la lista de selección de una instrucción SELECT (en una subconsulta o en la consulta externa), en cláusulas COMPUTE o COMPUTE BY y en cláusulas HAVING. El lenguaje de programación SQL proporciona funciones de agregado como las siguientes:</p>	

<i>Categoría</i>	<i>Explicación y tipos</i>	<i>Ejemplo</i>
<i>Funciones de agregado</i>	<p>AVG(<i>expresión</i>). Devuelve la media de los valores de un grupo definido por la expresión. Los valores NULL se pasan por alto.</p> <p>COUNT(<i>expresión</i>). Devuelve el número de elementos del grupo definido por la expresión. Los valores NULL se pasan por alto.</p> <p>MAX(<i>expresión</i>). Devuelve el valor máximo de la expresión. Los valores NULL se pasan por alto.</p> <p>MIN(<i>expresión</i>). Devuelve el valor mínimo de la expresión. Los valores NULL se pasan por alto.</p> <p>SUM(<i>expresión</i>). Devuelve la suma de todos los valores en la expresión especificada. SUM sólo puede utilizarse con columnas numéricas. Los valores NULL se pasan por alto.</p> <p>STDDEV(<i>expresión</i>) o STDDEV(<i>expresión</i>). Da la desviación típica estadística de todos los valores de la expresión especificada. Los valores NULL se pasan por alto.</p>	<p>SELECT AVG(advance)</p> <p>SELECT COUNT(city)</p> <p>SELECT MAX(ytd_sales)</p> <p>SELECT MIN(ytd_sales)</p> <p>SELECT type, SUM(price)</p> <p>SELECT STDDEV(royalty)</p>
<i>Categoría</i>	<i>Explicación y tipos</i>	<i>Ejemplo</i>
<i>Funciones de comparación</i>	<p>GREATEST(<i>exp1, exp2...</i>). Devuelve el argumento de mayor valor de la secuencia.</p> <p>LEAST(<i>exp1, exp2...</i>). Devuelve el argumento de menor valor de la secuencia.</p> <p>IF(<i>exp1, exp2, exp3</i>). Devuelve <i>exp2</i> si <i>exp1</i> es verdadero y da <i>exp3</i> si <i>exp1</i> es falso.</p> <p>IFNULL(<i>exp1, exp2</i>). Devuelve <i>exp2</i> si <i>exp1</i> es NULL y da <i>exp1</i> en caso contrario.</p> <p>INTERVAL(<i>n,n1, n2...</i>). Da 0 si $n < n1$, 1 si $n < n2$ y así sucesivamente. Da -1 si <i>n</i> es NULL.</p> <p>ISNULL(<i>exp</i>). Devuelve 1 si <i>exp1</i> es NULL y da 0 en caso contrario.</p> <p>STRCMP(<i>cadena1, cadena2</i>). Devuelve 1 si <i>cadena1</i> = <i>cadena2</i>, da 0 si son diferentes y NULL si cualquier elemento es NULL.</p>	<p>GREATEST(2,3,1)</p> <p>LEAST(2,3,1)</p> <p>IF(1.3, "non-zero", "zero")</p> <p>IF(NULL, "null")</p> <p>INTERVAL(1.1, 0,1,2)</p>