

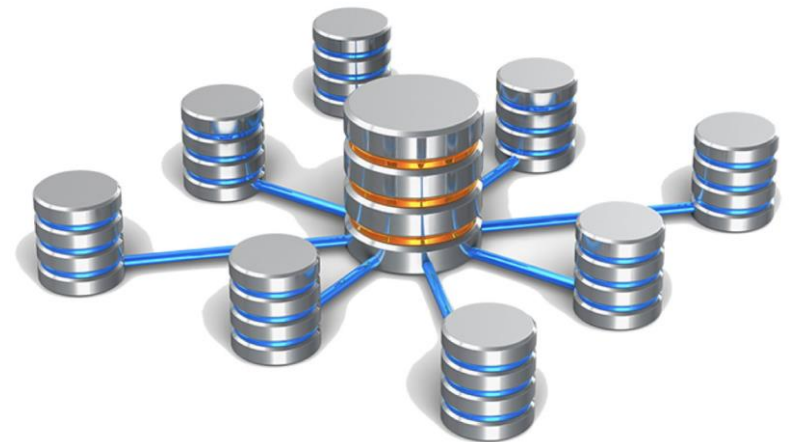
BBDD

Bases de Datos – Funciones en Consultas de Selección



Objetivos

- Utilizar funciones de MySQL en las consultas de información almacenada en una base de datos relacional.
- Saber cuando utilizar las funciones estudiadas.



Lenguaje SQL

El lenguaje de programación **SQL** es el **lenguaje fundamental de los SGBD** relacionales y los elementos que lo componen son:

- a) **DML (*Data Manipulation Language*)**: es el lenguaje que consulta o manipula los datos ya existentes de nuestra BD.
- b) **DDL (*Data Definition Language*)**: permite la **definición, modificación y eliminación de las estructuras básicas** (BD, tablas, etc.) en un SGBD.
- c) **DCL (*Data Control Language*)**: administra a los usuarios de la BD, concediendo o denegando los permisos oportunos.
- d) **TCL (*Transaction Control Language*)**: lenguaje que controla el procesamiento de las transacciones de la BD.

DML (Data Manipulation Language)

DML (Data Manipulation Language) o Lenguaje de Manipulación de Datos es la parte de SQL dedicada a la manipulación de los datos.

Las sentencias DML son las siguientes:

- **SELECT**: se utiliza para realizar consultas y extraer información de la base de datos.
- **INSERT**: se utiliza para insertar registros en las tablas de la base de datos.
- **UPDATE**: se utiliza para actualizar los registros de una tabla.
- **DELETE**: se utiliza para eliminar registros de una tabla.

En esta Unidad seguimos centrándonos en el uso de la sentencia **SELECT**.

<https://dev.mysql.com/doc/refman/8.0/en/select.html>



Funciones Consultas Selección

- **MySQL** como el resto de SGBDs relacionales va más allá de simplemente almacenar y recuperar datos. También podemos realizar manipulaciones en los datos antes de recuperarlos o guardarlos.

Funciones

- Las **funciones** nos permiten realizar la manipulación de los datos. Operan sobre ellos y luego devuelven un resultado. Algunas soportan parámetros y otras no.
- Con un programa de aplicación, un programador podría lograr lo mismo sin hacer uso de funciones de BBDD pero.... **No sería óptimo** en términos de eficiencia (sobrecarga de trabajo del servidor o sobrecarga de tráfico de red en las aplicaciones cte-servidor ...)
- En las consultas **SELECT**, las funciones tienen cabida únicamente en las cláusulas SELECT, WHERE y ORDER BY.
- Las categorías de funciones que veremos son:
 1. Funciones matemáticas (aritméticas)
 2. Funciones de cadenas de caracteres
 3. Funciones de manipulación de fecha y hora
 4. Funciones de conversión

```
SELECT
[ALL | DISTINCT | DISTINCTROW ]
[HIGH_PRIORITY]
[STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
[SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
select_expr [, select_expr] ...
[into_option]
[FROM table_references
[PARTITION partition_list]]
[WHERE where_condition]
[GROUP BY {col_name | expr | position}, ... [WITH ROLLUP]]
[HAVING where_condition]
[WINDOW window_name AS (window_spec)
[, window_name AS (window_spec)] ...]
[ORDER BY {col_name | expr | position}
[ASC | DESC], ... [WITH ROLLUP]]
[LIMIT {[offset,] row_count | row_count OFFSET offset}]
[into_option]
[FOR {UPDATE | SHARE}
[OF tbl_name [, tbl_name] ...]
[NOWAIT | SKIP LOCKED]
| LOCK IN SHARE MODE]
[into_option]

into_option: {
    INTO OUTFILE 'file_name'
        [CHARACTER SET charset_name]
        export_options
    | INTO DUMPFILE 'file_name'
    | INTO var_name [, var_name] ...
}
```

<https://dev.mysql.com/doc/refman/8.0/en/select.html>

Todas las funciones matemáticas retornan NULL en caso de error.



Funciones matemáticas

De entre las funciones matemáticas disponibles en MySQL, deberíamos familiarizarnos con al menos:

- **CEILING(x), CEIL(x)**: Retorna el entero más pequeño no menor a X.
- **FLOOR(x)**: Retorna el valor entero más grande pero no mayor a X.
- **MOD(n,m)** , **n % m**, **n MOD m**: Retorna el resto de N dividido por M.
- **POW(x,y)** , **POWER(x,y)**: Retorna el valor de X elevado a Y.
- **RAND()**, **RAND(N)** : Retorna un valor aleatorio en coma flotante del rango de 0 a 1. Usando en ORDER BY y combinado con LIMIT nos permite seleccionar una muestra aleatoria de registros.
- **ROUND(x)** : Retorna el argumento x, redondeado al entero más cercano.
- **SIGN(x)** : Retorna el signo del argumento como -1, 0 ó 1, en función de si x es negativo, cero o positivo.
- **TRUNCATE(x,d)** : Retorna el número x, truncado a d decimales. Si d es 0, el resultado no tendrá decimales. d puede ser negativo para truncar (hacer cero) d dígitos de la parte entera del valor x.
- **ABS(x)** : Retorna el valor absoluto de x.

Name	Description
<u>ABS ()</u>	Return the absolute value
<u>ACOS ()</u>	Return the arc cosine
<u>ASIN ()</u>	Return the arc sine
<u>ATAN ()</u>	Return the arc tangent
<u>ATAN2 (), ATAN ()</u>	Return the arc tangent of the two arguments
<u>CEIL ()</u>	Return the smallest integer value not less than the argument
<u>CEILING ()</u>	Return the smallest integer value not less than the argument
<u>CONV ()</u>	Convert numbers between different number bases
<u>COS ()</u>	Return the cosine
<u>COT ()</u>	Return the cotangent
<u>CRC32 ()</u>	Compute a cyclic redundancy check value
<u>DEGREES ()</u>	Convert radians to degrees
<u>EXP ()</u>	Raise to the power of
<u>FLOOR ()</u>	Return the largest integer value not greater than the argument
<u>LN ()</u>	Return the natural logarithm of the argument
<u>LOG ()</u>	Return the natural logarithm of the first argument
<u>LOG10 ()</u>	Return the base-10 logarithm of the argument
<u>LOG2 ()</u>	Return the base-2 logarithm of the argument
<u>MOD ()</u>	Return the remainder
<u>PI ()</u>	Return the value of pi
<u>POW ()</u>	Return the argument raised to the specified power
<u>POWER ()</u>	Return the argument raised to the specified power
<u>RADIANS ()</u>	Return argument converted to radians
<u>RAND ()</u>	Return a random floating-point value
<u>ROUND ()</u>	Round the argument
<u>SIGN ()</u>	Return the sign of the argument
<u>SIN ()</u>	Return the sine of the argument
<u>SQRT ()</u>	Return the square root of the argument
<u>TAN ()</u>	Return the tangent of the argument
<u>TRUNCATE ()</u>	Truncate to specified number of decimal places

<https://dev.mysql.com/doc/refman/8.0/en/mathematical-functions.html>



Funciones de cadenas de caracteres

De entre las funciones de cadenas de caracteres disponibles en MySQL, deberíamos familiarizarnos con al menos:

- **LOWER(cad), UPPER(cad):** Retorna la cadena cad en minúsculas o mayúsculas.
- **CHAR_LENGTH(cad):** Retorna la longitud de la cadena de caracteres cad, medida en caracteres.
- **LENGTH(cad):** Retorna la longitud de la cadena de caracteres cad, medida en bytes.
- **CONCAT(cad1,cad2,...):** Retorna la cadena resultado de concatenar los argumentos o NULL si alguna fuese NULL.
- **LEFT(cad,n), RIGHT(cad,n):** Retornan los n caracteres empezando por la izquierda o por la derecha de la cadena cad.
- **SUBSTRING(cad, pos):** Retornan una subcadena de la cadena cad comenzando en la posición pos.
- **REPLACE(cad,x,y):** Retorna la cadena cad sustituyendo las ocurrencias de x por y.
- **RTRIM(cad), LTRIM(cad), TRIM(cad):** Retorna la cadena cad con los espacios del final o iniciales eliminados o ambos.

Name	Description
ASCII()	Return numeric value of leftmost character
BIN()	Return a string containing binary representation of a number
BIT_LENGTH()	Return length of argument in bits
CHAR()	Return the character for each integer passed
CHAR_LENGTH()	Return number of characters in argument
CHARACTER_LENGTH()	Synonym for CHAR_LENGTH()
CONCAT()	Return concatenated string
CONCAT_WS()	Return concatenated with separator
FIND_IN_SET()	Return string as index number
GET_BINARY()	Return a string such that for every bit set in the value bits, you get an on string and for every unset bit, you get an off string
INDEX()	Index (position) of first argument in subsequent arguments
INSTR()	Index (position) of first argument within second argument
OCTET_LENGTH()	Return a number formatted to specified number of decimal places
ORD()	Decode base64 encoded string and return result
HEX()	Hexadecimal representation of decimal or string value
LEFT()	Insert substring at specified position up to specified number of characters
LOCATE()	Return the index of the first occurrence of substring
LOWER()	Synonym for LOWER()
LEFTS()	Return the leftmost number of characters as specified
LENGTH()	Return the length of a string in bytes
LIKE()	Simple pattern matching
LOAD_FILE()	Load the named file
LOCATE()	Return the position of the first occurrence of substring
LOWER()	Return the argument in lowercase
LPAD()	Return the string argument, left-padded with the specified string
LTRIM()	Remove leading spaces
MATCH()	Return a set of comma-separated strings that have the corresponding bit in bits set
MATCH()	Perform full-text search
MID()	Return a substring starting from the specified position
MOD()	Negation of simple pattern matching
MOD()	Negation of REGEXP
OCTET_LENGTH()	Return a string containing octal representation of a number
ORD()	Synonym for LENGTH()
ORD()	Return character code for leftmost character of the argument
ORD()	Synonym for LOCATE()
QUOTE()	Escape the argument for use in an SQL statement
REGEXP()	Whether string matches regular expression
REGEXP()	Starting index of substring matching regular expression
REGEXP()	Whether string matches regular expression
REGEXP_REPLACE()	Replace substrings matching regular expression
REGEXP_REPLACE()	Return substring matching regular expression
REPEAT()	Repeat a string the specified number of times
REPLACE()	Replace occurrences of a specified string
REVERSE()	Reverse the characters in a string
RIGHT()	Return the specified rightmost number of characters
RIGHTS()	Whether string matches regular expression
RIGHTS()	Append string the specified number of times
RTRIM()	Remove trailing spaces
SOUNDEX()	Return a soundex string
SOUNDEX()	Compare sounds
SPACE()	Return a string of the specified number of spaces
SPACE()	Compare two strings
SUBSTR()	Return the substring as specified
SUBSTR()	Return the substring as specified
SUBSTR_INDEX()	Return a substring from a string before the specified number of occurrences of the delimiter
TO_BASE64()	Return the argument converted to a base-64 string
TRIM()	Remove leading and trailing spaces
UPPER()	Synonym for UPPER()
UPPER()	Return a string containing hex representation of a number
UPPER()	Convert to uppercase
WEIGHT_STRING()	Return the weight string for a string

<https://dev.mysql.com/doc/refman/8.0/en/string-functions.html>



Funciones de fechas y horas

De entre las funciones de manipulación de fechas y horas disponibles en MySQL, deberíamos familiarizarnos con al menos:

- **SYSDATE(), NOW():** Retorna la fecha y hora actual.
- **CURDATE():** Retorna la fecha actual.
- **CURTIME():** Retorna la hora actual.
- **DAY(fech):** Retorna el día de la fecha fech.
- **MONTH(fech):** Retorna el mes de la fecha fech.
- **YEAR(fech):** Retorna el año de la fecha fech.
- **TIME(fech):** Retorna la hora de la fecha fech.
- **hour(h):** Retorna las horas de la hora h.
- **minute(h):** Retorna los minutos de la hora h.
- **second(h):** Retorna los segundos de la hora h.
- **DATE_ADD(fech, INTERVAL n day/week/month/year), DATE_SUB(fech, INTERVAL n day/week/month/year) :** Retorna la fecha resultante de añadir o quitar a la fecha fech, n días, semanas, meses o años.
- **DATE_DIFF(fech1, fech2):** Retorna el número de días entre la fecha inicial fech2 y la fecha final fech1, es decir, los días que faltan o los días que han pasado. Si fech1 < fech2 (-) Si fech1 > fech2 (+)
- **DATE_FORMAT(fech, formato):** Formatea la fecha fech según la cadena formato haciendo uso de los especificadores:

Name	Description
ADDDATE()	Add time values (intervals) to a date value
ADDTIME()	Add time
CONVERT_TZ()	Convert from one time zone to another
CURRENT_DATE()	Return the current date
CURRENT_DATE() , CURRENT_DATE()	Synonyms for CURDATE()
CURRENT_TIME() , CURRENT_TIME()	Synonyms for CURTIME()
CURRENT_TIMESTAMP() , CURRENT_TIMESTAMP()	Synonyms for NOW()
CURTIME()	Return the current time
DATE()	Extract the date part of a date or datetime expression
DATE_ADD()	Add time values (intervals) to a date value
DATE_FORMAT()	Format date as specified
DATE_SUB()	Subtract a time value (interval) from a date
DATEDIFF()	Subtract two dates
DAY()	Return the day of the year (1-366)
DAYNAME()	Return the name of the weekday
DAYOFMONTH()	Return the day of the month (1-31)
DAYOFWEEK()	Return the weekday index of the argument
DAYOFYEAR()	Return the day of the year (1-366)
EXTRACT()	Extract part of a date
FROM_UNIXTIME()	Convert a day number to a date
FROM_UNIXTIME()	Format Unix timestamps as a date
HOUR()	Return a date format string
HOUR()	Extract the hour
LAST_DAY()	Return the last day of the month for the argument
LAST_DAY() , LAST_DAY()	Synonyms for NOW()
LAST_DAY() , LAST_DAY()	Synonyms for NOW()
MAKE_DATE()	Create a date from the year and day of year
MAKE_TIME()	Create time from hour, minute, second
MICROSECOND()	Return the microseconds from argument
MINUTE()	Return the minute from the argument
MONTH()	Return the month from the date passed
MONTHNAME()	Return the name of the month
NOW()	Return the current date and time
PERIOD_ADD()	Add a period to a year-month
PERIOD_DIFF()	Return the number of months between periods
QUARTER()	Return the quarter from a date argument
SEC_TO_TIME()	Convert seconds to 'HH:MM:SS' format
SECOND()	Return the second (0-59)
STR_TO_DATE()	Convert a string to a date
SUBDATE()	Synonym for DATE_SUB() when invoked with three arguments
SUBTIME()	Subtract times
TIME()	Return the time at which the function executes
TIME()	Extract the time portion of the expression passed
TIME_FORMAT()	Format as time
TIME_TO_SEC()	Return the argument converted to seconds
TIMEDIFF()	Subtract times
TIMESTAMP()	With a single argument, this function returns the date or datetime expression; with two arguments, the sum of the arguments
TIMESTAMPADD()	Add an interval to a datetime expression
TIMESTAMPDIFF()	Subtract an interval from a datetime expression
TO_DATE()	Return the date argument converted to days
TO_TIMESTAMP()	Return the date or datetime arguments converted to seconds since Year 0
UNIX_TIMESTAMP()	Return a Unix timestamp
UTC_DATE()	Return the current UTC date
UTC_TIME()	Return the current UTC time
UTC_TIMESTAMP()	Return the current UTC date and time
WEEK()	Return the week number
WEEKDAY()	Return the weekday index
WEEKOFYEAR()	Return the calendar week of the date (1-53)
YEAR()	Return the year
YEARWEEK()	Return the year and week

<https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html>

- %d #Día del mes numérico (00...31)
- %H #Hora (00...23)
- %h #Hora (01...12)
- %i #Minutos, numérico (00...59)
- %M #Nombre mes (January...December)
- %m #Mes, numérico (00...12)
- %p #AM o PM
- %W #Nombre día semana (Sunday...Saturday)
- %Y #Año, numérico, cuatro dígitos
- %y #Año, numérico (dos dígitos)
- %s #Segundos (00...59)



Funciones de conversión

De entre las funciones de conversión disponibles en MySQL, deberíamos familiarizarnos con al menos:

- **CAST()**
- **CONVERT()**

Ambas funciones permiten convertir un tipo de datos en otro tipo diferente.

CAST es la función estándar y está disponible en todos los SGBDs debido a su portabilidad y facilidad de uso.

La firma de dichas funciones es:

- **CAST(expresion AS tipo)**
- **CONVERT(expresion, tipo)**

Donde *tipo* puede ser uno de los siguientes valores:

- BINARY
- CHAR (nos devuelve un varchar)
- DATE (nos devuelve "YYYY-MM-DD")
- DATETIME (nos devuelve "YYYY-MM-DD HH:MM:SS")
- DECIMAL
- SIGNED [INTEGER]
- TIME (nos devuelve "HH:MM:SS")
- UNSIGNED [INTEGER]

Name	Description	Deprecated
<u>BINARY</u>	Cast a string to a binary string	8.0.27
<u>CAST ()</u>	Cast a value as a certain type	
<u>CONVERT ()</u>	Cast a value as a certain type	

<https://dev.mysql.com/doc/refman/8.0/en/cast-functions.html>



Funciones de control de flujo

De entre las funciones de control de flujo disponibles en MySQL, deberíamos familiarizarnos con al menos:

- **CASE value WHEN** [compare-value] **THEN result** [ELSE result] **END**
- **IF(expr1,expr2,expr3)**: evalúa expr1, devuelve expr2 si es expr1 es true y expr3 si expr1 es false.
- **IFNULL(expr1,expr2)** : evalúa si es NULL expr1 y si lo es, devuelve expr2

Name	Description
<u>CASE</u>	Case operator
<u>IF ()</u>	If/else construct
<u>IFNULL ()</u>	Null if/else construct
<u>NULLIF ()</u>	Return NULL if expr1 = expr2

<https://dev.mysql.com/doc/refman/8.0/en/flow-control-functions.html>