String Functions and Operators

**Table 14.12 String Functions and Operators**

| **Name** | **Description** |
| --- | --- |
| [ASCII()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_ascii) | Return numeric value of left-most character |
| [BIN()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_bin) | Return a string containing binary representation of a number |
| [BIT\_LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_bit-length) | Return length of argument in bits |
| [CHAR()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char) | Return the character for each integer passed |
| [CHAR\_LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char-length) | Return number of characters in argument |
| [CHARACTER\_LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_character-length) | Synonym for CHAR\_LENGTH() |
| [CONCAT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_concat) | Return concatenated string |
| [CONCAT\_WS()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_concat-ws) | Return concatenate with separator |
| [ELT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_elt) | Return string at index number |
| [EXPORT\_SET()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_export-set) | 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 |
| [FIELD()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_field) | Index (position) of first argument in subsequent arguments |
| [FIND\_IN\_SET()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_find-in-set) | Index (position) of first argument within second argument |
| [FORMAT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_format) | Return a number formatted to specified number of decimal places |
| [HEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_hex) | Hexadecimal representation of decimal or string value |
| [INSERT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_insert) | Insert substring at specified position up to specified number of characters |
| [INSTR()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_instr) | Return the index of the first occurrence of substring |
| [LCASE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_lcase) | Synonym for LOWER() |
| [LEFT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_left) | Return the leftmost number of characters as specified |
| [LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_length) | Return the length of a string in bytes |
| [LIKE](https://dev.mysql.com/doc/refman/8.4/en/string-comparison-functions.html#operator_like) | Simple pattern matching |
| [LOAD\_FILE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_load-file) | Load the named file |
| [LOCATE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_locate) | Return the position of the first occurrence of substring |
| [LOWER()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_lower) | Return the argument in lowercase |
| [LPAD()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_lpad) | Return the string argument, left-padded with the specified string |
| [LTRIM()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_ltrim) | Remove leading spaces |
| [MAKE\_SET()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_make-set) | Return a set of comma-separated strings that have the corresponding bit in bits set |
| [MATCH()](https://dev.mysql.com/doc/refman/8.4/en/fulltext-search.html#function_match) | Perform full-text search |
| [MID()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_mid) | Return a substring starting from the specified position |
| [NOT LIKE](https://dev.mysql.com/doc/refman/8.4/en/string-comparison-functions.html#operator_not-like) | Negation of simple pattern matching |
| [NOT REGEXP](https://dev.mysql.com/doc/refman/8.4/en/regexp.html#operator_not-regexp) | Negation of REGEXP |
| [OCT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_oct) | Return a string containing octal representation of a number |
| [OCTET\_LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_octet-length) | Synonym for LENGTH() |
| [ORD()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_ord) | Return character code for leftmost character of the argument |
| [POSITION()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_position) | Synonym for LOCATE() |
| [QUOTE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_quote) | Escape the argument for use in an SQL statement |
| [REGEXP](https://dev.mysql.com/doc/refman/8.4/en/regexp.html#operator_regexp) | Whether string matches regular expression |
| [REGEXP\_INSTR()](https://dev.mysql.com/doc/refman/8.4/en/regexp.html#function_regexp-instr) | Starting index of substring matching regular expression |
| [REGEXP\_LIKE()](https://dev.mysql.com/doc/refman/8.4/en/regexp.html#function_regexp-like) | Whether string matches regular expression |
| [REGEXP\_REPLACE()](https://dev.mysql.com/doc/refman/8.4/en/regexp.html#function_regexp-replace) | Replace substrings matching regular expression |
| [REGEXP\_SUBSTR()](https://dev.mysql.com/doc/refman/8.4/en/regexp.html#function_regexp-substr) | Return substring matching regular expression |
| [REPEAT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_repeat) | Repeat a string the specified number of times |
| [REPLACE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_replace) | Replace occurrences of a specified string |
| [REVERSE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_reverse) | Reverse the characters in a string |
| [RIGHT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_right) | Return the specified rightmost number of characters |
| [RLIKE](https://dev.mysql.com/doc/refman/8.4/en/regexp.html#operator_regexp) | Whether string matches regular expression |
| [RPAD()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_rpad) | Append string the specified number of times |
| [RTRIM()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_rtrim) | Remove trailing spaces |
| [SOUNDEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_soundex) | Return a soundex string |
| [SOUNDS LIKE](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#operator_sounds-like) | Compare sounds |
| [SPACE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_space) | Return a string of the specified number of spaces |
| [STRCMP()](https://dev.mysql.com/doc/refman/8.4/en/string-comparison-functions.html#function_strcmp) | Compare two strings |
| [SUBSTR()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substr) | Return the substring as specified |
| [SUBSTRING()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substring) | Return the substring as specified |
| [SUBSTRING\_INDEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substring-index) | Return a substring from a string before the specified number of occurrences of the delimiter |
| [TRIM()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_trim) | Remove leading and trailing spaces |
| [UCASE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_ucase) | Synonym for UPPER() |
| [UNHEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_unhex) | Return a string containing hex representation of a number |
| [UPPER()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_upper) | Convert to uppercase |
| [WEIGHT\_STRING()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_weight-string) | Return the weight string for a string |

String-valued functions return NULL if the length of the result would be greater than the value of the [max\_allowed\_packet](https://dev.mysql.com/doc/refman/8.4/en/server-system-variables.html#sysvar_max_allowed_packet) system variable. See [Section 7.1.1, “Configuring the Server”](https://dev.mysql.com/doc/refman/8.4/en/server-configuration.html).

For functions that operate on string positions, the first position is numbered 1.

For functions that take length arguments, noninteger arguments are rounded to the nearest integer.

* [ASCII(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ascii)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ascii)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ascii)

Returns the numeric value of the leftmost character of the string ***str***. Returns 0 if ***str*** is the empty string. Returns NULL if ***str*** is NULL. [ASCII()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_ascii) works for 8-bit characters.

mysql> SELECT ASCII('2');

-> 50

mysql> SELECT ASCII(2);

-> 50

mysql> SELECT ASCII('dx');

-> 100

See also the [ORD()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_ord) function.

* [BIN(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_bin)***[N](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_bin)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_bin)

Returns a string representation of the binary value of ***N***, where ***N*** is a longlong ([BIGINT](https://dev.mysql.com/doc/refman/8.4/en/integer-types.html)) number. This is equivalent to [CONV(***N***,10,2)](https://dev.mysql.com/doc/refman/8.4/en/mathematical-functions.html#function_conv). Returns NULL if ***N*** is NULL.

mysql> SELECT BIN(12);

-> '1100'

* [BIT\_LENGTH(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_bit-length)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_bit-length)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_bit-length)

Returns the length of the string ***str*** in bits. Returns NULL if ***str*** is NULL.

mysql> SELECT BIT\_LENGTH('text');

-> 32

* [CHAR(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_char)***[N](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_char)***[,... [USING](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_char)***[charset\_name](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_char)***[])](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_char)

[CHAR()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_char) interprets each argument ***N*** as an integer and returns a string consisting of the characters given by the code values of those integers. NULL values are skipped.

mysql> SELECT CHAR(77,121,83,81,'76');

+--------------------------------------------------+

| CHAR(77,121,83,81,'76') |

+--------------------------------------------------+

| 0x4D7953514C |

+--------------------------------------------------+

1 row in set (0.00 sec)

mysql> SELECT CHAR(77,77.3,'77.3');

+--------------------------------------------+

| CHAR(77,77.3,'77.3') |

+--------------------------------------------+

| 0x4D4D4D |

+--------------------------------------------+

1 row in set (0.00 sec)

By default, [CHAR()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char) returns a binary string. To produce a string in a given character set, use the optional USING clause:

mysql> SELECT CHAR(77,121,83,81,'76' USING utf8mb4);

+---------------------------------------+

| CHAR(77,121,83,81,'76' USING utf8mb4) |

+---------------------------------------+

| MySQL |

+---------------------------------------+

1 row in set (0.00 sec)

mysql> SELECT CHAR(77,77.3,'77.3' USING utf8mb4);

+------------------------------------+

| CHAR(77,77.3,'77.3' USING utf8mb4) |

+------------------------------------+

| MMM |

+------------------------------------+

1 row in set, 1 warning (0.00 sec)

mysql> SHOW WARNINGS;

+---------+------+-------------------------------------------+

| Level | Code | Message |

+---------+------+-------------------------------------------+

| Warning | 1292 | Truncated incorrect INTEGER value: '77.3' |

+---------+------+-------------------------------------------+

1 row in set (0.00 sec)

If USING is given and the result string is illegal for the given character set, a warning is issued. Also, if strict SQL mode is enabled, the result from [CHAR()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char) becomes NULL.

If [CHAR()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char) is invoked from within the [**mysql**](https://dev.mysql.com/doc/refman/8.4/en/mysql.html) client, binary strings display using hexadecimal notation, depending on the value of the [--binary-as-hex](https://dev.mysql.com/doc/refman/8.4/en/mysql-command-options.html#option_mysql_binary-as-hex). For more information about that option, see [Section 6.5.1, “mysql — The MySQL Command-Line Client”](https://dev.mysql.com/doc/refman/8.4/en/mysql.html).

[CHAR()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char) arguments larger than 255 are converted into multiple result bytes. For example, [CHAR(256)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char) is equivalent to [CHAR(1,0)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char), and [CHAR(256\*256)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char) is equivalent to [CHAR(1,0,0)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char):

mysql> SELECT HEX(CHAR(1,0)), HEX(CHAR(256));

+----------------+----------------+

| HEX(CHAR(1,0)) | HEX(CHAR(256)) |

+----------------+----------------+

| 0100 | 0100 |

+----------------+----------------+

1 row in set (0.00 sec)

mysql> SELECT HEX(CHAR(1,0,0)), HEX(CHAR(256\*256));

+------------------+--------------------+

| HEX(CHAR(1,0,0)) | HEX(CHAR(256\*256)) |

+------------------+--------------------+

| 010000 | 010000 |

+------------------+--------------------+

1 row in set (0.00 sec)

* [CHAR\_LENGTH(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_char-length)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_char-length)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_char-length)

Returns the length of the string ***str***, measured in code points. A multibyte character counts as a single code point. This means that, for a string containing two 3-byte characters, [LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_length) returns 6, whereas [CHAR\_LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char-length) returns 2, as shown here:

mysql> SET ***@dolphin***:='海豚';

Query OK, 0 rows affected (0.01 sec)

mysql> SELECT LENGTH(***@dolphin***), CHAR\_LENGTH(***@dolphin***);

+------------------+-----------------------+

| LENGTH(@dolphin) | CHAR\_LENGTH(@dolphin) |

+------------------+-----------------------+

| 6 | 2 |

+------------------+-----------------------+

1 row in set (0.00 sec)

CHAR\_LENGTH() returns NULL if ***str*** is NULL.

* [CHARACTER\_LENGTH(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_character-length)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_character-length)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_character-length)

[CHARACTER\_LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_character-length) is a synonym for [CHAR\_LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char-length).

* [CONCAT(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat)***[str1](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat)***[str2](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat)***[,...)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat)

Returns the string that results from concatenating the arguments. May have one or more arguments. If all arguments are nonbinary strings, the result is a nonbinary string. If the arguments include any binary strings, the result is a binary string. A numeric argument is converted to its equivalent nonbinary string form.

[CONCAT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_concat) returns NULL if any argument is NULL.

mysql> SELECT CONCAT('My', 'S', 'QL');

-> 'MySQL'

mysql> SELECT CONCAT('My', NULL, 'QL');

-> NULL

mysql> SELECT CONCAT(14.3);

-> '14.3'

For quoted strings, concatenation can be performed by placing the strings next to each other:

mysql> SELECT 'My' 'S' 'QL';

-> 'MySQL'

If [CONCAT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_concat) is invoked from within the [**mysql**](https://dev.mysql.com/doc/refman/8.4/en/mysql.html) client, binary string results display using hexadecimal notation, depending on the value of the [--binary-as-hex](https://dev.mysql.com/doc/refman/8.4/en/mysql-command-options.html#option_mysql_binary-as-hex). For more information about that option, see [Section 6.5.1, “mysql — The MySQL Command-Line Client”](https://dev.mysql.com/doc/refman/8.4/en/mysql.html).

* [CONCAT\_WS(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat-ws)***[separator](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat-ws)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat-ws)***[str1](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat-ws)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat-ws)***[str2](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat-ws)***[,...)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat-ws)

[CONCAT\_WS()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_concat-ws) stands for Concatenate With Separator and is a special form of [CONCAT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_concat). The first argument is the separator for the rest of the arguments. The separator is added between the strings to be concatenated. The separator can be a string, as can the rest of the arguments. If the separator is NULL, the result is NULL.

mysql> SELECT CONCAT\_WS(',', 'First name', 'Second name', 'Last Name');

-> 'First name,Second name,Last Name'

mysql> SELECT CONCAT\_WS(',', 'First name', NULL, 'Last Name');

-> 'First name,Last Name'

[CONCAT\_WS()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_concat-ws) does not skip empty strings. However, it does skip any NULL values after the separator argument.

* [ELT(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_elt)***[N](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_elt)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_elt)***[str1](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_elt)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_elt)***[str2](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_elt)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_elt)***[str3](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_elt)***[,...)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_elt)

[ELT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_elt) returns the ***N***th element of the list of strings: ***str1*** if ***N*** = 1, ***str2*** if ***N*** = 2, and so on. Returns NULL if ***N*** is less than 1, greater than the number of arguments, or NULL. [ELT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_elt) is the complement of [FIELD()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_field).

mysql> SELECT ELT(1, 'Aa', 'Bb', 'Cc', 'Dd');

-> 'Aa'

mysql> SELECT ELT(4, 'Aa', 'Bb', 'Cc', 'Dd');

-> 'Dd'

* [EXPORT\_SET(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)***[bits](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)***[on](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)***[off](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)***[[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)***[separator](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)***[[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)***[number\_of\_bits](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)***[]])](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_export-set)

Returns a string such that for every bit set in the value ***bits***, you get an ***on*** string and for every bit not set in the value, you get an ***off*** string. Bits in ***bits*** are examined from right to left (from low-order to high-order bits). Strings are added to the result from left to right, separated by the ***separator*** string (the default being the comma character ,). The number of bits examined is given by ***number\_of\_bits***, which has a default of 64 if not specified. ***number\_of\_bits*** is silently clipped to 64 if larger than 64. It is treated as an unsigned integer, so a value of −1 is effectively the same as 64.

mysql> SELECT EXPORT\_SET(5,'Y','N',',',4);

-> 'Y,N,Y,N'

mysql> SELECT EXPORT\_SET(6,'1','0',',',10);

-> '0,1,1,0,0,0,0,0,0,0'

* [FIELD(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_field)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_field)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_field)***[str1](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_field)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_field)***[str2](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_field)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_field)***[str3](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_field)***[,...)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_field)

Returns the index (position) of ***str*** in the ***str1***, ***str2***, ***str3***, ... list. Returns 0 if ***str*** is not found.

If all arguments to [FIELD()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_field) are strings, all arguments are compared as strings. If all arguments are numbers, they are compared as numbers. Otherwise, the arguments are compared as double.

If ***str*** is NULL, the return value is 0 because NULL fails equality comparison with any value. [FIELD()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_field) is the complement of [ELT()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_elt).

mysql> SELECT FIELD('Bb', 'Aa', 'Bb', 'Cc', 'Dd', 'Ff');

-> 2

mysql> SELECT FIELD('Gg', 'Aa', 'Bb', 'Cc', 'Dd', 'Ff');

-> 0

* [FIND\_IN\_SET(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_find-in-set)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_find-in-set)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_find-in-set)***[strlist](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_find-in-set)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_find-in-set)

Returns a value in the range of 1 to ***N*** if the string ***str*** is in the string list ***strlist*** consisting of ***N*** substrings. A string list is a string composed of substrings separated by , characters. If the first argument is a constant string and the second is a column of type [SET](https://dev.mysql.com/doc/refman/8.4/en/set.html), the [FIND\_IN\_SET()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_find-in-set) function is optimized to use bit arithmetic. Returns 0 if ***str*** is not in ***strlist*** or if ***strlist*** is the empty string. Returns NULL if either argument is NULL. This function does not work properly if the first argument contains a comma (,) character.

mysql> SELECT FIND\_IN\_SET('b','a,b,c,d');

-> 2

* [FORMAT(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_format)***[X](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_format)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_format)***[D](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_format)***[[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_format)***[locale](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_format)***[])](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_format)

Formats the number ***X*** to a format like '#,###,###.##', rounded to ***D*** decimal places, and returns the result as a string. If ***D*** is 0, the result has no decimal point or fractional part. If ***X*** or ***D*** is NULL, the function returns NULL.

The optional third parameter enables a locale to be specified to be used for the result number's decimal point, thousands separator, and grouping between separators. Permissible locale values are the same as the legal values for the [lc\_time\_names](https://dev.mysql.com/doc/refman/8.4/en/server-system-variables.html#sysvar_lc_time_names) system variable (see [Section 12.16, “MySQL Server Locale Support”](https://dev.mysql.com/doc/refman/8.4/en/locale-support.html)). If the locale is NULL or not specified, the default locale is 'en\_US'.

mysql> SELECT FORMAT(12332.123456, 4);

-> '12,332.1235'

mysql> SELECT FORMAT(12332.1,4);

-> '12,332.1000'

mysql> SELECT FORMAT(12332.2,0);

-> '12,332'

mysql> SELECT FORMAT(12332.2,2,'de\_DE');

-> '12.332,20'

* [FROM\_BASE64(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_from-base64)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_from-base64)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_from-base64)

Takes a string encoded with the base-64 encoded rules used by [TO\_BASE64()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_to-base64) and returns the decoded result as a binary string. The result is NULL if the argument is NULL or not a valid base-64 string. See the description of [TO\_BASE64()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_to-base64) for details about the encoding and decoding rules.

mysql> SELECT TO\_BASE64('abc'), FROM\_BASE64(TO\_BASE64('abc'));

-> 'JWJj', 'abc'

If [FROM\_BASE64()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_from-base64) is invoked from within the [**mysql**](https://dev.mysql.com/doc/refman/8.4/en/mysql.html) client, binary strings display using hexadecimal notation. You can disable this behavior by setting the value of the [--binary-as-hex](https://dev.mysql.com/doc/refman/8.4/en/mysql-command-options.html#option_mysql_binary-as-hex) to 0 when starting the [**mysql**](https://dev.mysql.com/doc/refman/8.4/en/mysql.html) client. For more information about that option, see [Section 6.5.1, “mysql — The MySQL Command-Line Client”](https://dev.mysql.com/doc/refman/8.4/en/mysql.html).

* [HEX(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_hex)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_hex)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_hex), [HEX(***N***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_hex)

For a string argument ***str***, [HEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_hex) returns a hexadecimal string representation of ***str*** where each byte of each character in ***str*** is converted to two hexadecimal digits. (Multibyte characters therefore become more than two digits.) The inverse of this operation is performed by the [UNHEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_unhex) function.

For a numeric argument ***N***, [HEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_hex) returns a hexadecimal string representation of the value of ***N*** treated as a longlong ([BIGINT](https://dev.mysql.com/doc/refman/8.4/en/integer-types.html)) number. This is equivalent to [CONV(***N***,10,16)](https://dev.mysql.com/doc/refman/8.4/en/mathematical-functions.html#function_conv). The inverse of this operation is performed by [CONV(HEX(***N***),16,10)](https://dev.mysql.com/doc/refman/8.4/en/mathematical-functions.html#function_conv).

For a NULL argument, this function returns NULL.

mysql> SELECT X'616263', HEX('abc'), UNHEX(HEX('abc'));

-> 'abc', 616263, 'abc'

mysql> SELECT HEX(255), CONV(HEX(255),16,10);

-> 'FF', 255

* [INSERT(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_insert)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_insert)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_insert)***[pos](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_insert)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_insert)***[len](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_insert)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_insert)***[newstr](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_insert)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_insert)

Returns the string ***str***, with the substring beginning at position ***pos*** and ***len*** characters long replaced by the string ***newstr***. Returns the original string if ***pos*** is not within the length of the string. Replaces the rest of the string from position ***pos*** if ***len*** is not within the length of the rest of the string. Returns NULL if any argument is NULL.

mysql> SELECT INSERT('Quadratic', 3, 4, 'What');

-> 'QuWhattic'

mysql> SELECT INSERT('Quadratic', -1, 4, 'What');

-> 'Quadratic'

mysql> SELECT INSERT('Quadratic', 3, 100, 'What');

-> 'QuWhat'

This function is multibyte safe.

* [INSTR(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_instr)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_instr)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_instr)***[substr](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_instr)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_instr)

Returns the position of the first occurrence of substring ***substr*** in string ***str***. This is the same as the two-argument form of [LOCATE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_locate), except that the order of the arguments is reversed.

mysql> SELECT INSTR('foobarbar', 'bar');

-> 4

mysql> SELECT INSTR('xbar', 'foobar');

-> 0

This function is multibyte safe, and is case-sensitive only if at least one argument is a binary string. If either argument is NULL, this functions returns NULL.

* [LCASE(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lcase)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lcase)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lcase)

[LCASE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lcase) is a synonym for [LOWER()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_lower).

LCASE() used in a view is rewritten as LOWER() when storing the view's definition. (Bug #12844279)

* [LEFT(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_left)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_left)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_left)***[len](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_left)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_left)

Returns the leftmost ***len*** characters from the string ***str***, or NULL if any argument is NULL.

mysql> SELECT LEFT('foobarbar', 5);

-> 'fooba'

This function is multibyte safe.

* [LENGTH(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_length)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_length)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_length)

Returns the length of the string ***str***, measured in bytes. A multibyte character counts as multiple bytes. This means that for a string containing five 2-byte characters, [LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_length) returns 10, whereas [CHAR\_LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_char-length) returns 5. Returns NULL if ***str*** is NULL.

mysql> SELECT LENGTH('text');

-> 4

**Note**

The Length() OpenGIS spatial function is named [ST\_Length()](https://dev.mysql.com/doc/refman/8.4/en/gis-linestring-property-functions.html#function_st-length) in MySQL.

* [LOAD\_FILE(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_load-file)***[file\_name](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_load-file)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_load-file)

Reads the file and returns the file contents as a string. To use this function, the file must be located on the server host, you must specify the full path name to the file, and you must have the [FILE](https://dev.mysql.com/doc/refman/8.4/en/privileges-provided.html#priv_file) privilege. The file must be readable by the server and its size less than [max\_allowed\_packet](https://dev.mysql.com/doc/refman/8.4/en/server-system-variables.html#sysvar_max_allowed_packet) bytes. If the [secure\_file\_priv](https://dev.mysql.com/doc/refman/8.4/en/server-system-variables.html#sysvar_secure_file_priv) system variable is set to a nonempty directory name, the file to be loaded must be located in that directory.

If the file does not exist or cannot be read because one of the preceding conditions is not satisfied, the function returns NULL.

The [character\_set\_filesystem](https://dev.mysql.com/doc/refman/8.4/en/server-system-variables.html#sysvar_character_set_filesystem) system variable controls interpretation of file names that are given as literal strings.

mysql> UPDATE t

SET blob\_col=LOAD\_FILE('/tmp/picture')

WHERE id=1;

* [LOCATE(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_locate)***[substr](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_locate)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_locate)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_locate)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_locate), [LOCATE(***substr***,***str***,***pos***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_locate)

The first syntax returns the position of the first occurrence of substring ***substr*** in string ***str***. The second syntax returns the position of the first occurrence of substring ***substr*** in string ***str***, starting at position ***pos***. Returns 0 if ***substr*** is not in ***str***. Returns NULL if any argument is NULL.

mysql> SELECT LOCATE('bar', 'foobarbar');

-> 4

mysql> SELECT LOCATE('xbar', 'foobar');

-> 0

mysql> SELECT LOCATE('bar', 'foobarbar', 5);

-> 7

This function is multibyte safe, and is case-sensitive only if at least one argument is a binary string.

* [LOWER(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lower)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lower)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lower)

Returns the string ***str*** with all characters changed to lowercase according to the current character set mapping, or NULL if ***str*** is NULL. The default character set is utf8mb4.

mysql> SELECT LOWER('QUADRATICALLY');

-> 'quadratically'

[LOWER()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_lower) (and [UPPER()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_upper)) are ineffective when applied to binary strings ([BINARY](https://dev.mysql.com/doc/refman/8.4/en/binary-varbinary.html), [VARBINARY](https://dev.mysql.com/doc/refman/8.4/en/binary-varbinary.html), [BLOB](https://dev.mysql.com/doc/refman/8.4/en/blob.html)). To perform lettercase conversion of a binary string, first convert it to a nonbinary string using a character set appropriate for the data stored in the string:

mysql> SET ***@str*** = BINARY 'New York';

mysql> SELECT LOWER(***@str***), LOWER(CONVERT(***@str*** USING utf8mb4));

+-------------+------------------------------------+

| LOWER(@str) | LOWER(CONVERT(@str USING utf8mb4)) |

+-------------+------------------------------------+

| New York | new york |

+-------------+------------------------------------+

For collations of Unicode character sets, [LOWER()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_lower) and [UPPER()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_upper) work according to the Unicode Collation Algorithm (UCA) version in the collation name, if there is one, and UCA 4.0.0 if no version is specified. For example, utf8mb4\_0900\_ai\_ci and utf8mb3\_unicode\_520\_ci work according to UCA 9.0.0 and 5.2.0, respectively, whereas utf8mb3\_unicode\_ci works according to UCA 4.0.0. See [Section 12.10.1, “Unicode Character Sets”](https://dev.mysql.com/doc/refman/8.4/en/charset-unicode-sets.html).

This function is multibyte safe.

LCASE() used within views is rewritten as LOWER().

* [LPAD(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lpad)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lpad)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lpad)***[len](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lpad)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lpad)***[padstr](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lpad)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_lpad)

Returns the string ***str***, left-padded with the string ***padstr*** to a length of ***len*** characters. If ***str*** is longer than ***len***, the return value is shortened to ***len*** characters.

mysql> SELECT LPAD('hi',4,'??');

-> '??hi'

mysql> SELECT LPAD('hi',1,'??');

-> 'h'

Returns NULL if any of its arguments are NULL.

* [LTRIM(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ltrim)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ltrim)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ltrim)

Returns the string ***str*** with leading space characters removed. Returns NULL if ***str*** is NULL.

mysql> SELECT LTRIM(' barbar');

-> 'barbar'

This function is multibyte safe.

* [MAKE\_SET(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_make-set)***[bits](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_make-set)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_make-set)***[str1](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_make-set)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_make-set)***[str2](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_make-set)***[,...)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_make-set)

Returns a set value (a string containing substrings separated by , characters) consisting of the strings that have the corresponding bit in ***bits*** set. ***str1*** corresponds to bit 0, ***str2*** to bit 1, and so on. NULL values in ***str1***, ***str2***, ... are not appended to the result.

mysql> SELECT MAKE\_SET(1,'a','b','c');

-> 'a'

mysql> SELECT MAKE\_SET(1 | 4,'hello','nice','world');

-> 'hello,world'

mysql> SELECT MAKE\_SET(1 | 4,'hello','nice',NULL,'world');

-> 'hello'

mysql> SELECT MAKE\_SET(0,'a','b','c');

-> ''

* [MID(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid)***[pos](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid), [MID(***str*** FROM ***pos***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_mid), [MID(***str***,***pos***,***len***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_mid), [MID(***str*** FROM ***pos*** FOR ***len***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_mid)

[MID(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid)***[pos](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid)***[len](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_mid) is a synonym for [SUBSTRING(***str***,***pos***,***len***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substring).

* [OCT(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_oct)***[N](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_oct)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_oct)

Returns a string representation of the octal value of ***N***, where ***N*** is a longlong ([BIGINT](https://dev.mysql.com/doc/refman/8.4/en/integer-types.html)) number. This is equivalent to [CONV(***N***,10,8)](https://dev.mysql.com/doc/refman/8.4/en/mathematical-functions.html#function_conv). Returns NULL if ***N*** is NULL.

mysql> SELECT OCT(12);

-> '14'

* [OCTET\_LENGTH(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_octet-length)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_octet-length)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_octet-length)

[OCTET\_LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_octet-length) is a synonym for [LENGTH()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_length).

* [ORD(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ord)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ord)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ord)

If the leftmost character of the string ***str*** is a multibyte character, returns the code for that character, calculated from the numeric values of its constituent bytes using this formula:

(1st byte code)

+ (2nd byte code \* 256)

+ (3rd byte code \* 256^2) ...

If the leftmost character is not a multibyte character, [ORD()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_ord) returns the same value as the [ASCII()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_ascii) function. The function returns NULL if ***str*** is NULL.

mysql> SELECT ORD('2');

-> 50

* [POSITION(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_position)***[substr](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_position)***[IN](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_position)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_position)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_position)

[POSITION(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_position)***[substr](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_position)***[IN](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_position)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_position)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_position) is a synonym for [LOCATE(***substr***,***str***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_locate).

* [QUOTE(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_quote)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_quote)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_quote)

Quotes a string to produce a result that can be used as a properly escaped data value in an SQL statement. The string is returned enclosed by single quotation marks and with each instance of backslash (\), single quote ('), ASCII NUL, and Control+Z preceded by a backslash. If the argument is NULL, the return value is the word “NULL” without enclosing single quotation marks.

mysql> SELECT QUOTE('Don\'t!');

-> 'Don\'t!'

mysql> SELECT QUOTE(NULL);

-> NULL

For comparison, see the quoting rules for literal strings and within the C API in [Section 11.1.1, “String Literals”](https://dev.mysql.com/doc/refman/8.4/en/string-literals.html), and [mysql\_real\_escape\_string\_quote()](https://dev.mysql.com/doc/c-api/8.4/en/mysql-real-escape-string-quote.html).

* [REPEAT(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_repeat)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_repeat)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_repeat)***[count](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_repeat)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_repeat)

Returns a string consisting of the string ***str*** repeated ***count*** times. If ***count*** is less than 1, returns an empty string. Returns NULL if ***str*** or ***count*** is NULL.

mysql> SELECT REPEAT('MySQL', 3);

-> 'MySQLMySQLMySQL'

* [REPLACE(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_replace)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_replace)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_replace)***[from\_str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_replace)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_replace)***[to\_str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_replace)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_replace)

Returns the string ***str*** with all occurrences of the string ***from\_str*** replaced by the string ***to\_str***. [REPLACE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_replace) performs a case-sensitive match when searching for ***from\_str***.

mysql> SELECT REPLACE('www.mysql.com', 'w', 'Ww');

-> 'WwWwWw.mysql.com'

This function is multibyte safe. It returns NULL if any of its arguments are NULL.

* [REVERSE(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_reverse)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_reverse)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_reverse)

Returns the string ***str*** with the order of the characters reversed, or NULL if ***str*** is NULL.

mysql> SELECT REVERSE('abc');

-> 'cba'

This function is multibyte safe.

* [RIGHT(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_right)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_right)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_right)***[len](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_right)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_right)

Returns the rightmost ***len*** characters from the string ***str***, or NULL if any argument is NULL.

mysql> SELECT RIGHT('foobarbar', 4);

-> 'rbar'

This function is multibyte safe.

* [RPAD(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_rpad)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_rpad)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_rpad)***[len](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_rpad)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_rpad)***[padstr](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_rpad)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_rpad)

Returns the string ***str***, right-padded with the string ***padstr*** to a length of ***len*** characters. If ***str*** is longer than ***len***, the return value is shortened to ***len*** characters. If ***str***, ***padstr***, or ***len*** is NULL, the function returns NULL.

mysql> SELECT RPAD('hi',5,'?');

-> 'hi???'

mysql> SELECT RPAD('hi',1,'?');

-> 'h'

This function is multibyte safe.

* [RTRIM(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_rtrim)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_rtrim)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_rtrim)

Returns the string ***str*** with trailing space characters removed.

mysql> SELECT RTRIM('barbar ');

-> 'barbar'

This function is multibyte safe, and returns NULL if ***str*** is NULL.

* [SOUNDEX(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_soundex)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_soundex)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_soundex)

Returns a soundex string from ***str***, or NULL if ***str*** is NULL. Two strings that sound almost the same should have identical soundex strings. A standard soundex string is four characters long, but the [SOUNDEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_soundex) function returns an arbitrarily long string. You can use [SUBSTRING()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substring) on the result to get a standard soundex string. All nonalphabetic characters in ***str*** are ignored. All international alphabetic characters outside the A-Z range are treated as vowels.

**Important**

When using [SOUNDEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_soundex), you should be aware of the following limitations:

* + This function, as currently implemented, is intended to work well with strings that are in the English language only. Strings in other languages may not produce reliable results.
  + This function is not guaranteed to provide consistent results with strings that use multibyte character sets, including utf-8. See Bug #22638 for more information.

mysql> SELECT SOUNDEX('Hello');

-> 'H400'

mysql> SELECT SOUNDEX('Quadratically');

-> 'Q36324'

**Note**

This function implements the original Soundex algorithm, not the more popular enhanced version (also described by D. Knuth). The difference is that original version discards vowels first and duplicates second, whereas the enhanced version discards duplicates first and vowels second.

* ***[expr1](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "operator_sounds-like)***[SOUNDS LIKE](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "operator_sounds-like)***[expr2](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "operator_sounds-like)***

This is the same as [SOUNDEX(***expr1***) = SOUNDEX(***expr2***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_soundex).

* [SPACE(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_space)***[N](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_space)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_space)

Returns a string consisting of ***N*** space characters, or NULL if ***N*** is NULL.

mysql> SELECT SPACE(6);

-> ' '

* [SUBSTR(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substr)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substr)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substr)***[pos](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substr)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substr), [SUBSTR(***str*** FROM ***pos***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substr), [SUBSTR(***str***,***pos***,***len***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substr), [SUBSTR(***str*** FROM ***pos*** FOR ***len***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substr)

[SUBSTR()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substr) is a synonym for [SUBSTRING()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substring).

* [SUBSTRING(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring)***[pos](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring), [SUBSTRING(***str*** FROM ***pos***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substring), [SUBSTRING(***str***,***pos***,***len***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substring), [SUBSTRING(***str*** FROM ***pos*** FOR ***len***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substring)

The forms without a ***len*** argument return a substring from string ***str*** starting at position ***pos***. The forms with a ***len*** argument return a substring ***len*** characters long from string ***str***, starting at position ***pos***. The forms that use FROM are standard SQL syntax. It is also possible to use a negative value for ***pos***. In this case, the beginning of the substring is ***pos*** characters from the end of the string, rather than the beginning. A negative value may be used for ***pos*** in any of the forms of this function. A value of 0 for ***pos*** returns an empty string.

For all forms of [SUBSTRING()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substring), the position of the first character in the string from which the substring is to be extracted is reckoned as 1.

mysql> SELECT SUBSTRING('Quadratically',5);

-> 'ratically'

mysql> SELECT SUBSTRING('foobarbar' FROM 4);

-> 'barbar'

mysql> SELECT SUBSTRING('Quadratically',5,6);

-> 'ratica'

mysql> SELECT SUBSTRING('Sakila', -3);

-> 'ila'

mysql> SELECT SUBSTRING('Sakila', -5, 3);

-> 'aki'

mysql> SELECT SUBSTRING('Sakila' FROM -4 FOR 2);

-> 'ki'

This function is multibyte safe. It returns NULL if any of its arguments are NULL.

If ***len*** is less than 1, the result is the empty string.

* [SUBSTRING\_INDEX(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring-index)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring-index)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring-index)***[delim](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring-index)***[,](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring-index)***[count](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring-index)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_substring-index)

Returns the substring from string ***str*** before ***count*** occurrences of the delimiter ***delim***. If ***count*** is positive, everything to the left of the final delimiter (counting from the left) is returned. If ***count*** is negative, everything to the right of the final delimiter (counting from the right) is returned. [SUBSTRING\_INDEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_substring-index) performs a case-sensitive match when searching for ***delim***.

mysql> SELECT SUBSTRING\_INDEX('www.mysql.com', '.', 2);

-> 'www.mysql'

mysql> SELECT SUBSTRING\_INDEX('www.mysql.com', '.', -2);

-> 'mysql.com'

This function is multibyte safe.

SUBSTRING\_INDEX() returns NULL if any of its arguments are NULL.

* [TO\_BASE64(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_to-base64)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_to-base64)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_to-base64)

Converts the string argument to base-64 encoded form and returns the result as a character string with the connection character set and collation. If the argument is not a string, it is converted to a string before conversion takes place. The result is NULL if the argument is NULL. Base-64 encoded strings can be decoded using the [FROM\_BASE64()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_from-base64) function.

mysql> SELECT TO\_BASE64('abc'), FROM\_BASE64(TO\_BASE64('abc'));

-> 'JWJj', 'abc'

Different base-64 encoding schemes exist. These are the encoding and decoding rules used by [TO\_BASE64()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_to-base64) and [FROM\_BASE64()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_from-base64):

* + The encoding for alphabet value 62 is '+'.
  + The encoding for alphabet value 63 is '/'.
  + Encoded output consists of groups of 4 printable characters. Each 3 bytes of the input data are encoded using 4 characters. If the last group is incomplete, it is padded with '=' characters to a length of 4.
  + A newline is added after each 76 characters of encoded output to divide long output into multiple lines.
  + Decoding recognizes and ignores newline, carriage return, tab, and space.

* [TRIM([{BOTH | LEADING | TRAILING} [](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_trim)***[remstr](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_trim)***[] FROM]](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_trim)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_trim)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_trim), [TRIM([***remstr*** FROM] ***str***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_trim)

Returns the string ***str*** with all ***remstr*** prefixes or suffixes removed. If none of the specifiers BOTH, LEADING, or TRAILING is given, BOTH is assumed. ***remstr*** is optional and, if not specified, spaces are removed.

mysql> SELECT TRIM(' bar ');

-> 'bar'

mysql> SELECT TRIM(LEADING 'x' FROM 'xxxbarxxx');

-> 'barxxx'

mysql> SELECT TRIM(BOTH 'x' FROM 'xxxbarxxx');

-> 'bar'

mysql> SELECT TRIM(TRAILING 'xyz' FROM 'barxxyz');

-> 'barx'

This function is multibyte safe. It returns NULL if any of its arguments are NULL.

* [UCASE(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ucase)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ucase)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ucase)

[UCASE()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_ucase) is a synonym for [UPPER()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_upper).

UCASE() used within views is rewritten as UPPER().

* [UNHEX(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_unhex)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_unhex)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_unhex)

For a string argument ***str***, [UNHEX(***str***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_unhex) interprets each pair of characters in the argument as a hexadecimal number and converts it to the byte represented by the number. The return value is a binary string.

mysql> SELECT UNHEX('4D7953514C');

-> 'MySQL'

mysql> SELECT X'4D7953514C';

-> 'MySQL'

mysql> SELECT UNHEX(HEX('string'));

-> 'string'

mysql> SELECT HEX(UNHEX('1267'));

-> '1267'

The characters in the argument string must be legal hexadecimal digits: '0' .. '9', 'A' .. 'F', 'a' .. 'f'. If the argument contains any nonhexadecimal digits, or is itself NULL, the result is NULL:

mysql> SELECT UNHEX('GG');

+-------------+

| UNHEX('GG') |

+-------------+

| NULL |

+-------------+

mysql> SELECT UNHEX(NULL);

+-------------+

| UNHEX(NULL) |

+-------------+

| NULL |

+-------------+

A NULL result can also occur if the argument to [UNHEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_unhex) is a [BINARY](https://dev.mysql.com/doc/refman/8.4/en/binary-varbinary.html) column, because values are padded with 0x00 bytes when stored but those bytes are not stripped on retrieval. For example, '41' is stored into a CHAR(3) column as '41 ' and retrieved as '41' (with the trailing pad space stripped), so [UNHEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_unhex) for the column value returns X'41'. By contrast, '41' is stored into a BINARY(3) column as '41\0' and retrieved as '41\0' (with the trailing pad 0x00 byte not stripped). '\0' is not a legal hexadecimal digit, so [UNHEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_unhex) for the column value returns NULL.

For a numeric argument ***N***, the inverse of [HEX(***N***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_hex) is not performed by [UNHEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_unhex). Use [CONV(HEX(***N***),16,10)](https://dev.mysql.com/doc/refman/8.4/en/mathematical-functions.html#function_conv) instead. See the description of [HEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_hex).

If [UNHEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_unhex) is invoked from within the [**mysql**](https://dev.mysql.com/doc/refman/8.4/en/mysql.html) client, binary strings display using hexadecimal notation, depending on the value of the [--binary-as-hex](https://dev.mysql.com/doc/refman/8.4/en/mysql-command-options.html#option_mysql_binary-as-hex). For more information about that option, see [Section 6.5.1, “mysql — The MySQL Command-Line Client”](https://dev.mysql.com/doc/refman/8.4/en/mysql.html).

* [UPPER(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_upper)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_upper)***[)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_upper)

Returns the string ***str*** with all characters changed to uppercase according to the current character set mapping, or NULL if ***str*** is NULL. The default character set is utf8mb4.

mysql> SELECT UPPER('Hej');

-> 'HEJ'

See the description of [LOWER()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_lower) for information that also applies to [UPPER()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_upper). This included information about how to perform lettercase conversion of binary strings ([BINARY](https://dev.mysql.com/doc/refman/8.4/en/binary-varbinary.html), [VARBINARY](https://dev.mysql.com/doc/refman/8.4/en/binary-varbinary.html), [BLOB](https://dev.mysql.com/doc/refman/8.4/en/blob.html)) for which these functions are ineffective, and information about case folding for Unicode character sets.

This function is multibyte safe.

UCASE() used within views is rewritten as UPPER().

* [WEIGHT\_STRING(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_weight-string)***[str](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_weight-string)***[[AS {CHAR|BINARY}(](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_weight-string)***[N](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_weight-string)***[)] [](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_weight-string)***[flags](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_weight-string)***[])](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html" \l "function_weight-string)

This function returns the weight string for the input string. The return value is a binary string that represents the comparison and sorting value of the string, or NULL if the argument is NULL. It has these properties:

* + If [WEIGHT\_STRING(***str1***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_weight-string) = [WEIGHT\_STRING(***str2***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_weight-string), then ***str1*** = ***str2*** (***str1*** and ***str2*** are considered equal)
  + If [WEIGHT\_STRING(***str1***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_weight-string) < [WEIGHT\_STRING(***str2***)](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_weight-string), then ***str1*** < ***str2*** (***str1*** sorts before ***str2***)

[WEIGHT\_STRING()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_weight-string) is a debugging function intended for internal use. Its behavior can change without notice between MySQL versions. It can be used for testing and debugging of collations, especially if you are adding a new collation. See [Section 12.14, “Adding a Collation to a Character Set”](https://dev.mysql.com/doc/refman/8.4/en/adding-collation.html).

This list briefly summarizes the arguments. More details are given in the discussion following the list.

* + ***str***: The input string expression.
  + AS clause: Optional; cast the input string to a given type and length.
  + ***flags***: Optional; unused.

The input string, ***str***, is a string expression. If the input is a nonbinary (character) string such as a [CHAR](https://dev.mysql.com/doc/refman/8.4/en/char.html), [VARCHAR](https://dev.mysql.com/doc/refman/8.4/en/char.html), or [TEXT](https://dev.mysql.com/doc/refman/8.4/en/blob.html) value, the return value contains the collation weights for the string. If the input is a binary (byte) string such as a [BINARY](https://dev.mysql.com/doc/refman/8.4/en/binary-varbinary.html), [VARBINARY](https://dev.mysql.com/doc/refman/8.4/en/binary-varbinary.html), or [BLOB](https://dev.mysql.com/doc/refman/8.4/en/blob.html) value, the return value is the same as the input (the weight for each byte in a binary string is the byte value). If the input is NULL, [WEIGHT\_STRING()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_weight-string) returns NULL.

Examples:

mysql> SET ***@s*** = \_utf8mb4 'AB' COLLATE utf8mb4\_0900\_ai\_ci;

mysql> SELECT ***@s***, HEX(***@s***), HEX(WEIGHT\_STRING(***@s***));

+------+---------+------------------------+

| @s | HEX(@s) | HEX(WEIGHT\_STRING(@s)) |

+------+---------+------------------------+

| AB | 4142 | 1C471C60 |

+------+---------+------------------------+

mysql> SET ***@s*** = \_utf8mb4 'ab' COLLATE utf8mb4\_0900\_ai\_ci;

mysql> SELECT ***@s***, HEX(***@s***), HEX(WEIGHT\_STRING(***@s***));

+------+---------+------------------------+

| @s | HEX(@s) | HEX(WEIGHT\_STRING(@s)) |

+------+---------+------------------------+

| ab | 6162 | 1C471C60 |

+------+---------+------------------------+

mysql> SET ***@s*** = CAST('AB' AS BINARY);

mysql> SELECT ***@s***, HEX(***@s***), HEX(WEIGHT\_STRING(***@s***));

+------+---------+------------------------+

| @s | HEX(@s) | HEX(WEIGHT\_STRING(@s)) |

+------+---------+------------------------+

| AB | 4142 | 4142 |

+------+---------+------------------------+

mysql> SET ***@s*** = CAST('ab' AS BINARY);

mysql> SELECT ***@s***, HEX(***@s***), HEX(WEIGHT\_STRING(***@s***));

+------+---------+------------------------+

| @s | HEX(@s) | HEX(WEIGHT\_STRING(@s)) |

+------+---------+------------------------+

| ab | 6162 | 6162 |

+------+---------+------------------------+

The preceding examples use [HEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_hex) to display the [WEIGHT\_STRING()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_weight-string) result. Because the result is a binary value, [HEX()](https://dev.mysql.com/doc/refman/8.4/en/string-functions.html#function_hex) can be especially useful when the result contains nonprinting values, to display it in printable form:

mysql> SET ***@s*** = CONVERT(X'C39F' USING utf8mb4) COLLATE utf8mb4\_czech\_ci;

mysql> SELECT HEX(WEIGHT\_STRING(***@s***));

+------------------------+

| HEX(WEIGHT\_STRING(@s)) |

+------------------------+

| 0FEA0FEA |

+------------------------+

For non-NULL return values, the data type of the value is [VARBINARY](https://dev.mysql.com/doc/refman/8.4/en/binary-varbinary.html) if its length is within the maximum length for [VARBINARY](https://dev.mysql.com/doc/refman/8.4/en/binary-varbinary.html), otherwise the data type is [BLOB](https://dev.mysql.com/doc/refman/8.4/en/blob.html).

The AS clause may be given to cast the input string to a nonbinary or binary string and to force it to a given length:

* + AS CHAR(***N***) casts the string to a nonbinary string and pads it on the right with spaces to a length of ***N*** characters. ***N*** must be at least 1. If ***N*** is less than the length of the input string, the string is truncated to ***N*** characters. No warning occurs for truncation.
  + AS BINARY(***N***) is similar but casts the string to a binary string, ***N*** is measured in bytes (not characters), and padding uses 0x00 bytes (not spaces).

mysql> SET NAMES 'latin1';

mysql> SELECT HEX(WEIGHT\_STRING('ab' AS CHAR(4)));

+-------------------------------------+

| HEX(WEIGHT\_STRING('ab' AS CHAR(4))) |

+-------------------------------------+

| 41422020 |

+-------------------------------------+

mysql> SET NAMES 'utf8mb4';

mysql> SELECT HEX(WEIGHT\_STRING('ab' AS CHAR(4)));

+-------------------------------------+

| HEX(WEIGHT\_STRING('ab' AS CHAR(4))) |

+-------------------------------------+

| 1C471C60 |

+-------------------------------------+

mysql> SELECT HEX(WEIGHT\_STRING('ab' AS BINARY(4)));

+---------------------------------------+

| HEX(WEIGHT\_STRING('ab' AS BINARY(4))) |

+---------------------------------------+

| 61620000 |

+------------------------------------