SQL: Structured Query Language
MySQL Built-in Function

MySQL (and most other SQL Servers) have many useful built-in functions.

Math Functions

GREATEST(`num1`, `num2`, `num3`, ..., `numN`)
: Find the largest argument.

LEAST(`num1`, `num2`, `num3`, ..., `numN`)
: Find the smallest argument.

CEIL(`num`)
: Return the smallest integer value not less than the argument.

FLOOR(`num`)
: Return the largest integer value not greater than the argument.

ROUND(`num`)
: Do a standard round.

TRUNCATE(`num`, `places`)
: Truncate to `places` number of decimal places. If you let `places` = 0, then you can do an integer truncation.

POW(base, exponent)
: Raises `base` to `exponent`.

SQRT(`num`)
: Finds the square root of `num`.

RAND([`seed`])
: Returns a random floating point value `v` in the range $0 \leq v < 1.0$. Can be optionally seeded. Each subsequent call to RAND() with the same seed will give the next number in the sequence.

  Can be used in conjunction with ORDER BY to get a random ordering of your result set:

```
...
ORDER BY RAND()
```

SIGN(`num`)
: Returns the sign (1 or -1) of `num`.

String Functions

STRCMP(`str1`, `str2`)
: Returns 0 if the strings are equal, -1 if `str1` comes before `str2`, and 1 if `str1` comes after `str2`.

LOWER(`str`)
: Returns an all lowercase version of `str`.

UPPER(`str`)
: Returns an all uppercase version of `str`.

REVERSE(`str`)
: Returns a reversed copy of `str`.

TRIM([{BOTH | LEADING | TRAILING} [`removeStr`] FROM] `str`)
: Return a string with all `removeStr` prefixes and/or suffixes removed. If none of the specifiers BOTH, LEADING, or TRAILING is given, BOTH is assumed. `removeStr` is optional and, if not specified, spaces are removed.

`str1` LIKE `str2`
: Very simple pattern matching. Two special characters are recognized (both can be escaped with a backslash):

  % Matches any number of characters, even zero characters.

  _ Matches exactly one character.

`string` REGEXP `pattern`
: Full regular expression matching. See `http://dev.mysql.com/doc/refman/5.0/en/regexp.html` for all the details.

`str1` SOUNDS LIKE `str2`
: Check for equality between the Soundex (`http://en.wikipedia.org/wiki/Soundex`) of two strings. Only works well with English.

CONCAT(`str1`, `str2`, ..., `strN`)
: Return a single string that concatenation of all the arguments.

CONCAT_WS(`separator`, `str1`, `str2`, ..., `strN`)
: Return a single string that concatenation of all the arguments with each argument delimited with `separator`.

Compression

COMPRESS(`str`)
: Compresses `str` into a binary string using whatever compression library MySQL was compiled with. (If it was compiled with no compression library, then this will return null.)

UNCOMPRESS(`compressedStr`)
: Uncompresses `compressedStr` which should have been compressed using COMPRESS(). (Like COMPRESS(), if MySQL was compiled with no compression library, then this will return null.)

Security

MD5(`str`)
: Calculates an MD5 128-bit checksum for `str`. The checksum is returned as a 32 length string in hex.

SHA(`str`)
: Calculates an SHA-1 160-bit checksum for `str`. The checksum is returned as a 40 length string in hex.

AES_ENCRYPT(`str`, `key`)
: Encrypt `str` using the official AES (Advanced Encryption Standard) algorithm with a 128-bit key. `key` is used as the encryption key, and will be padded to the full key size.

AES_DECRYPT(`encryptStr`, `key`)
: Decrypt `encryptStr` using the official AES (Advanced Encryption Standard) algorithm with a 128-bit key. `key` is used as the decryption key, and will be padded to the full key size.

DES_ENCRYPT(`str`, `key`)
: Encrypt `str` using the official Triple-DES. `key` is used as the encryption key. There is a variant of this function that works with a file known keys. (This function only works if MySQL was compiled with SSL support.)

DES_DECRYPT(`encryptStr`, `key`)
: Decrypt `str` using the official Triple-DES. `key` is used as the decryption key. (This function only works if MySQL was compiled with SSL support.)