



MYSQL DATE FUNCTIONS

Shadab Khan



Contents

CONVERT_TZ(dt,from_tz,to_tz).....	2
CURDATE()	2
CURRENT_DATE and CURRENT_DATE()	2
CURTIME().....	2
CURRENT_TIME and CURRENT_TIME().....	3
DATE(expr).....	3
DATEDIFF(expr1,expr2).....	3
DATE_FORMAT(date,format)	4
DAY(date)	7
DAYNAME(date).....	7
DAYOFMONTH(date)	7
DAYOFWEEK(date).....	7
DAYOFYEAR(date)	8
FROM_DAYS(N).....	8
HOUR(time).....	8
LAST_DAY(date).....	9
LOCALTIME and LOCALTIME()	9
MAKEDATE(year,dayofyear).....	9
MINUTE(time).....	10
MONTH(date).....	10
MONTHNAME(date).....	10
NOW().....	11
SECOND(time).....	11
SEC_TO_TIME(seconds).....	11
STR_TO_DATE(str,format).....	12
SYSDATE().....	12
TIME(expr).....	12
TIME_TO_SEC(time)	13
TO_DAYS(date)	13
WEEK(date[,mode]).....	13
WEEKDAY(date)	14

WEEKOFYEAR(date)	15
YEAR(date)	15

CONVERT_TZ(dt,from_tz,to_tz)

This converts a datetime value dt from the time zone given by from_tz to the time zone given by to_tz and returns the resulting value. This function returns NULL if the arguments are invalid.

```
mysql> SELECT CONVERT_TZ('2004-01-01 12:00:00','+00:00','+10:00');
+-----+
| CONVERT_TZ('2004-01-01 12:00:00','+00:00','+10:00') |
+-----+
| 2004-01-01 22:00:00 |
+-----+
1 row in set (0.00 sec)
```

CURDATE()

Returns the current date as a value in 'YYYY-MM-DD' or YYYYMMDD format, depending on whether the function is used in a string or numeric context.

```
mysql> SELECT CURDATE();

mysql> SELECT CURDATE() + 0;
```

CURRENT_DATE and CURRENT_DATE()

CURRENT_DATE and CURRENT_DATE() are synonyms for CURDATE()

CURTIME()

Returns the current time as a value in 'HH:MM:SS' or HHMMSS format, depending on whether the function is used in a string or numeric context. The value is expressed in the current time zone.

```
mysql> SELECT CURTIME();
+-----+
```

```

| CURTIME() |
+-----+
| 23:50:26 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT CURTIME() + 0;
+-----+
| CURTIME() + 0 |
+-----+
| 235026 |
+-----+
1 row in set (0.00 sec)

```

CURRENT_TIME and CURRENT_TIME()

CURRENT_TIME and CURRENT_TIME() are synonyms for CURTIME().

DATE(expr)

Extracts the date part of the date or datetime expression expr.

```

mysql> SELECT DATE('2003-12-31 01:02:03');
+-----+
| DATE('2003-12-31 01:02:03') |
+-----+
| 2003-12-31 |
+-----+
1 row in set (0.00 sec)

```

DATEDIFF(expr1,expr2)

DATEDIFF() returns expr1 . expr2 expressed as a value in days from one date to the other. expr1 and expr2 are date or date-and-time expressions. Only the date parts of the values are used in the calculation.

```

mysql> SELECT DATEDIFF('1997-12-31 23:59:59','1997-12-30');

```

```
+-----+
| DATEDIFF('1997-12-31 23:59:59','1997-12-30') |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)
```

DATE_FORMAT(date,format)

Formats the date value according to the format string.

The following specifiers may be used in the format string. The .%. character is required before format specifier characters.

Specifier	Description
%a	Abbreviated weekday name (Sun..Sat)
%b	Abbreviated month name (Jan..Dec)
%c	Month, numeric (0..12)
%D	Day of the month with English suffix (0th, 1st, 2nd, 3rd, .)
%d	Day of the month, numeric (00..31)
%e	Day of the month, numeric (0..31)
%f	Microseconds (000000..999999)
%H	Hour (00..23)
%h	Hour (01..12)

%I	Hour (01..12)
%i	Minutes, numeric (00..59)
%j	Day of year (001..366)
%k	Hour (0..23)
%l	Hour (1..12)
%M	Month name (January..December)
%m	Month, numeric (00..12)
%p	AM or PM
%r	Time, 12-hour (hh:mm:ss followed by AM or PM)
%S	Seconds (00..59)
%s	Seconds (00..59)
%T	Time, 24-hour (hh:mm:ss)
%U	Week (00..53), where Sunday is the first day of the week
%u	Week (00..53), where Monday is the first day of the week
%V	Week (01..53), where Sunday is the first day of the week; used with %X

%v	Week (01..53), where Monday is the first day of the week; used with %x
%W	Weekday name (Sunday..Saturday)
%w	Day of the week (0=Sunday..6=Saturday)
%X	Year for the week where Sunday is the first day of the week, numeric, four digits; used with %V
%x	Year for the week, where Monday is the first day of the week, numeric, four digits; used with %v
%Y	Year, numeric, four digits
%y	Year, numeric (two digits)
%%	A literal .%. character
%x	x, for any .x. not listed above

```
mysql> SELECT DATE_FORMAT('1997-10-04 22:23:00', '%W %M %Y');
```

```
+-----+
| DATE_FORMAT('1997-10-04 22:23:00', '%W %M %Y') |
+-----+
| Saturday October 1997 |
+-----+
```

```
1 row in set (0.00 sec)
```

```
mysql> SELECT DATE_FORMAT('1997-10-04 22:23:00'
-> '%H %k %I %r %T %S %w');
```

```
+-----+
| DATE_FORMAT('1997-10-04 22:23:00.....' |
```

```

+-----+
| 22 22 10 10:23:00 PM 22:23:00 00 6 |
+-----+
1 row in set (0.00 sec)

```

DAY(date)

DAY() is a synonym for DAYOFMONTH().

DAYNAME(date)

Returns the name of the weekday for date.

```

mysql> SELECT DAYNAME('1998-02-05');
+-----+
| DAYNAME('1998-02-05') |
+-----+
| Thursday               |
+-----+
1 row in set (0.00 sec)

```

DAYOFMONTH(date)

Returns the day of the month for date, in the range 0 to 31.

```

mysql> SELECT DAYOFMONTH('1998-02-03');
+-----+
| DAYOFMONTH('1998-02-03') |
+-----+
| 3                         |
+-----+
1 row in set (0.00 sec)

```

DAYOFWEEK(date)

Returns the weekday index for date (1 = Sunday, 2 = Monday, ..., 7 = Saturday). These index values correspond to the ODBC standard.

```

mysql> SELECT DAYOFWEEK('1998-02-03');

```



```

+-----+
| DAYOFWEEK('1998-02-03') |
+-----+
| 3 |
+-----+
1 row in set (0.00 sec)

```

DAYOFYEAR(date)

Returns the day of the year for date, in the range 1 to 366.

```

mysql> SELECT DAYOFYEAR('1998-02-03');
+-----+
| DAYOFYEAR('1998-02-03') |
+-----+
| 34 |
+-----+
1 row in set (0.00 sec)

```

FROM_DAYS(N)

Given a day number N, returns a DATE value.

```

mysql> SELECT FROM_DAYS(729669);
+-----+
| FROM_DAYS(729669) |
+-----+
| 1997-10-07 |
+-----+
1 row in set (0.00 sec)

```

Use FROM_DAYS() with caution on old dates. It is not intended for use with values that precede the advent of the Gregorian calendar (1582).

HOUR(time)

Returns the hour for the time. The range of the return value is 0 to 23 for time-of-day values. However, the range of TIME values actually is much larger, so HOUR can return values greater than 23.

```
mysql> SELECT HOUR('10:05:03');
+-----+
| HOUR('10:05:03') |
+-----+
| 10                |
+-----+
1 row in set (0.00 sec)
```

LAST_DAY(date)

Takes a date or datetime value and returns the corresponding value for the last day of the month. Returns NULL if the argument is invalid.

```
mysql> SELECT LAST_DAY('2003-02-05');
+-----+
| LAST_DAY('2003-02-05') |
+-----+
| 2003-02-28             |
+-----+
1 row in set (0.00 sec)
```

LOCALTIME and LOCALTIME()

LOCALTIME and LOCALTIME() are synonyms for NOW().

MAKEDATE(year,dayofyear)

Returns a date, given year and day-of-year values. dayofyear must be greater than 0 or the result is NULL.

```
mysql> SELECT MAKEDATE(2001,31), MAKEDATE(2001,32);
+-----+
| MAKEDATE(2001,31), MAKEDATE(2001,32) |
+-----+
```

```
| '2001-01-31', '2001-02-01' |
+-----+
1 row in set (0.00 sec)
```

MINUTE(time)

Returns the minute for time, in the range 0 to 59.

```
mysql> SELECT MINUTE('98-02-03 10:05:03');
+-----+
| MINUTE('98-02-03 10:05:03') |
+-----+
| 5 |
+-----+
1 row in set (0.00 sec)
```

MONTH(date)

Returns the month for date, in the range 0 to 12.

```
mysql> SELECT MONTH('1998-02-03')
+-----+
| MONTH('1998-02-03') |
+-----+
| 2 |
+-----+
1 row in set (0.00 sec)
```

MONTHNAME(date)

Returns the full name of the month for date.

```
mysql> SELECT MONTHNAME('1998-02-05');
+-----+
| MONTHNAME('1998-02-05') |
+-----+
| February |
+-----+
```

```
1 row in set (0.00 sec)
```

NOW()

Returns the current date and time as a value in 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMMSS format, depending on whether the function is used in a string or numeric context. The value is expressed in the current time zone.

```
mysql> SELECT NOW();
+-----+
| NOW() |
+-----+
| 1997-12-15 23:50:26 |
+-----+
1 row in set (0.00 sec)
```

SECOND(time)

Returns the second for time, in the range 0 to 59.

```
mysql> SELECT SECOND('10:05:03');
+-----+
| SECOND('10:05:03') |
+-----+
| 3 |
+-----+
1 row in set (0.00 sec)
```

SEC_TO_TIME(seconds)

Returns the seconds argument, converted to hours, minutes, and seconds, as a value in 'HH:MM:SS' or HHMMSS format, depending on whether the function is used in a string or numeric context.

```
mysql> SELECT SEC_TO_TIME(2378);
+-----+
| SEC_TO_TIME(2378) |
+-----+
```

```
| 00:39:38 |
+-----+
1 row in set (0.00 sec)
```

STR_TO_DATE(str,format)

This is the inverse of the DATE_FORMAT() function. It takes a string str and a format string format. STR_TO_DATE() returns a DATETIME value if the format string contains both date and time parts, or a DATE or TIME value if the string contains only date or time parts.

```
mysql> SELECT STR_TO_DATE('04/31/2004', '%m/%d/%Y');
+-----+
| STR_TO_DATE('04/31/2004', '%m/%d/%Y') |
+-----+
| 2004-04-31 |
+-----+
1 row in set (0.00 sec)
```

SYSDATE()

Returns the current date and time as a value in 'YYYY-MM-DD HH:MM:SS' or YYYYMMDDHHMMSS format, depending on whether the function is used in a string or numeric context.

```
mysql> SELECT SYSDATE();
```

TIME(expr)

Extracts the time part of the time or datetime expression expr and returns it as a string.

```
mysql> SELECT TIME('2003-12-31 01:02:03');
+-----+
| TIME('2003-12-31 01:02:03') |
+-----+
| 01:02:03 |
+-----+
```

```
+-----+
1 row in set (0.00 sec)
```

TIME_TO_SEC(time)

Returns the time argument, converted to seconds.

```
mysql> SELECT TIME_TO_SEC('22:23:00');

+-----+
| TIME_TO_SEC('22:23:00') |
+-----+
| 80580                    |
+-----+

1 row in set (0.00 sec)
```

TO_DAYS(date)

Given a date, returns a day number (the number of days since year 0).

```
mysql> SELECT TO_DAYS(950501);

+-----+
| TO_DAYS(950501) |
+-----+
| 728779          |
+-----+

1 row in set (0.00 sec)
```

WEEK(date[,mode])

This function returns the week number for date. The two-argument form of WEEK() allows you to specify whether the week starts on Sunday or Monday and whether the return value should be in the range from 0 to 53 or from 1 to 53. If the mode argument is omitted, the value of the default_week_format system variable is used

Mode	First Day of week	Range	Week 1 is the first week .

0	Sunday	0-53	with a Sunday in this year
1	Monday	0-53	with more than 3 days this year
2	Sunday	1-53	with a Sunday in this year
3	Monday	1-53	with more than 3 days this year
4	Sunday	0-53	with more than 3 days this year
5	Monday	0-53	with a Monday in this year
6	Sunday	1-53	with more than 3 days this year
7	Monday	1-53	with a Monday in this year

```
mysql> SELECT WEEK('1998-02-20');
```

```
+-----+
| WEEK('1998-02-20') |
+-----+
| 7 |
+-----+
```

```
1 row in set (0.00 sec)
```

WEEKDAY(date)

Returns the weekday index for date (0 = Monday, 1 = Tuesday, . 6 = Sunday).

```
mysql> SELECT WEEKDAY('1998-02-03 22:23:00');
```

```
+-----+
| WEEKDAY('1998-02-03 22:23:00') |
+-----+
| 1 |
+-----+
```

```
+-----+
1 row in set (0.00 sec)
```

WEEKOFYEAR(date)

Returns the calendar week of the date as a number in the range from 1 to 53. WEEKOFYEAR() is a compatibility function that is equivalent to WEEK(date,3).

```
mysql> SELECT WEEKOFYEAR('1998-02-20');

+-----+
| WEEKOFYEAR('1998-02-20') |
+-----+
| 8                         |
+-----+

1 row in set (0.00 sec)
```

YEAR(date)

Returns the year for date, in the range 1000 to 9999, or 0 for the .zero. date.

```
mysql> SELECT YEAR('98-02-03');

+-----+
| YEAR('98-02-03') |
+-----+
| 1998              |
+-----+

1 row in set (0.00 sec)
```