SQL - Date Functions

The following table has a list of all the important Date and Time related functions available through SQL. There are various other functions supported by your RDBMS. The given list is based on MySQL RDBMS.

Sr.No.	Function & Description
1	ADDDATE() Adds dates
2	ADDTIME() Adds time
3	CONVERT_TZ() Converts from one timezone to another
4	CURDATE() Returns the current date
5	CURRENT_DATE(), CURRENT_DATE Synonyms for CURDATE()
6	CURRENT_TIME(), CURRENT_TIME Synonyms for CURTIME()
7	CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP Synonyms for NOW()
8	CURTIME() Returns the current time
9	DATE_ADD() Adds two dates
10	DATE_FORMAT() Formats date as specified
11	DATE_SUB() Subtracts two dates
12	DATE() Extracts the date part of a date or datetime expression

13	DATEDIFF() Subtracts two dates
14	DAY() Synonym for DAYOFMONTH()
15	DAYNAME() Returns the name of the weekday
16	DAYOFMONTH() Returns the day of the month (1-31)
17	DAYOFWEEK() Returns the weekday index of the argument
18	DAYOFYEAR() Returns the day of the year (1-366)
19	EXTRACT Extracts part of a date
20	FROM_DAYS() Converts a day number to a date
21	FROM_UNIXTIME() Formats date as a UNIX timestamp
22	HOUR() Extracts the hour
23	LAST_DAY Returns the last day of the month for the argument
24	LOCALTIME(), LOCALTIME Synonym for NOW()

25	LOCALTIMESTAMP, LOCALTIMESTAMP() Synonym for NOW()
26	MAKEDATE() Creates a date from the year and day of year
27	MAKETIME MAKETIME()
28	MICROSECOND() Returns the microseconds from argument
29	MINUTE() Returns the minute from the argument
30	MONTH() Return the month from the date passed
31	MONTHNAME() Returns the name of the month
32	NOW() Returns the current date and time
33	PERIOD_ADD() Adds a period to a year-month
34	PERIOD_DIFF() Returns the number of months between periods
35	QUARTER() Returns the quarter from a date argument
36	SEC_TO_TIME() Converts seconds to 'HH:MM:SS' format
37	SECOND()

	Returns the second (0-59)
38	STR_TO_DATE() Converts a string to a date
39	SUBDATE() When invoked with three arguments a synonym for DATE_SUB()
40	SUBTIME() Subtracts times
41	SYSDATE() Returns the time at which the function executes
42	TIME_FORMAT() Formats as time
43	TIME_TO_SEC() Returns the argument converted to seconds
44	TIME() Extracts the time portion of the expression passed
45	TIMEDIFF() Subtracts time
46	TIMESTAMP() With a single argument this function returns the date or datetime expression. With two arguments, the sum of the arguments
47	TIMESTAMPADD() Adds an interval to a datetime expression
48	TIMESTAMPDIFF() Subtracts an interval from a datetime expression
49	TO_DAYS()

	Returns the date argument converted to days
50	UNIX_TIMESTAMP() Returns a UNIX timestamp
51	UTC_DATE() Returns the current UTC date
52	UTC_TIME() Returns the current UTC time
53	UTC_TIMESTAMP() Returns the current UTC date and time
54	WEEK() Returns the week number
55	WEEKDAY() Returns the weekday index
56	WEEKOFYEAR() Returns the calendar week of the date (1-53)
57	YEAR() Returns the year
58	YEARWEEK() Returns the year and week

ADDDATE(date,INTERVAL expr unit), ADDDATE(expr,days)

When invoked with the INTERVAL form of the second argument, ADDDATE() is a synonym for DATE_ADD(). The related function SUBDATE() is a synonym for DATE_SUB(). For information on the INTERVAL unit argument, see the discussion for DATE_ADD().

```
mysql> SELECT DATE_ADD('1998-01-02', INTERVAL 31 DAY);
+------------------------------------------------------+
| DATE_ADD('1998-01-02', INTERVAL 31 DAY)              |
+------------------------------------------------------+
```

```
| 1998-02-02                                                |
+-----------------------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT ADDDATE('1998-01-02', INTERVAL 31 DAY);
+-----------------------------------------------------------+
| ADDDATE('1998-01-02', INTERVAL 31 DAY)                    |
+-----------------------------------------------------------+
| 1998-02-02                                                |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

When invoked with the days form of the second argument, MySQL treats it as an integer number of days to be added to expr.

```
mysql> SELECT ADDDATE('1998-01-02', 31);
+-----------------------------------------------------------+
| DATE_ADD('1998-01-02', INTERVAL 31 DAY)                   |
+-----------------------------------------------------------+
| 1998-02-02                                                |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

ADDTIME(expr1,expr2)

ADDTIME() adds expr2 to expr1 and returns the result. The expr1 is a time or datetime expression, while the expr2 is a time expression.

```
mysql> SELECT ADDTIME('1997-12-31 23:59:59.999999','1 1:1:1.000002');
+-----------------------------------------------------------+
| DATE_ADD('1997-12-31 23:59:59.999999','1 1:1:1.000002') |
+-----------------------------------------------------------+
| 1998-01-02 01:01:01.000001                                |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

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','GMT','MET');
+-----------------------------------------------------------+
| CONVERT_TZ('2004-01-01 12:00:00','GMT','MET')             |
+-----------------------------------------------------------+
```

```
| 2004-01-01 13:00:00                                    |
+--------------------------------------------------------+
1 row in set (0.00 sec)

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 in a numeric context.

```
mysql> SELECT CURDATE();
+------------------------------------------------------+
| CURDATE()                                            |
+------------------------------------------------------+
| 1997-12-15                                           |
+------------------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT CURDATE() + 0;
+------------------------------------------------------+
| CURDATE() + 0                                        |
+------------------------------------------------------+
| 19971215                                             |
+------------------------------------------------------+
1 row in set (0.00 sec)
```

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 in a 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().

CURRENT_TIMESTAMP and CURRENT_TIMESTAMP()

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

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. Both 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_ADD(date,INTERVAL expr unit), DATE_SUB(date,INTERVAL expr unit)

These functions perform date arithmetic. The **date** is a DATETIME or DATE value specifying the starting date. The **expr** is an expression specifying the interval value to be added or subtracted from the starting date. The expr is a string; it may start with a '-' for negative intervals.

A **unit** is a keyword indicating the units in which the expression should be interpreted.

The **INTERVAL** keyword and the unit specifier are not case sensitive.

The following table shows the expected form of the expr argument for each unit value.

unit Value	Expected exprFormat
MICROSECOND	MICROSECONDS
SECOND	SECONDS
MINUTE	MINUTES
HOUR	HOURS
DAY	DAYS
WEEK	WEEKS
MONTH	MONTHS
QUARTER	QUARTERS
YEAR	YEARS
SECOND_MICROSECOND	'SECONDS.MICROSECONDS'
MINUTE_MICROSECOND	'MINUTES.MICROSECONDS'
MINUTE_SECOND	'MINUTES:SECONDS'
HOUR_MICROSECOND	'HOURS.MICROSECONDS'
HOUR_SECOND	'HOURS:MINUTES:SECONDS'
HOUR_MINUTE	'HOURS:MINUTES'
DAY_MICROSECOND	'DAYS.MICROSECONDS'
DAY_SECOND	'DAYS HOURS:MINUTES:SECONDS'
DAY_MINUTE	'DAYS HOURS:MINUTES'
DAY_HOUR	'DAYS HOURS'
YEAR_MONTH	'YEARS-MONTHS'

The values **QUARTER** and **WEEK** are available from the MySQL 5.0.0. version.

```
mysql> SELECT DATE_ADD('1997-12-31 23:59:59',
    -> INTERVAL '1:1' MINUTE_SECOND);
+-----------------------------------------------------+
| DATE_ADD('1997-12-31 23:59:59', INTERVAL...         |
```

```
+-----------------------------------------------------+
| 1998-01-01 00:01:00                                 |
+-----------------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT DATE_ADD('1999-01-01', INTERVAL 1 HOUR);
+-----------------------------------------------------+
| DATE_ADD('1999-01-01', INTERVAL 1 HOUR)             |
+-----------------------------------------------------+
| 1999-01-01 01:00:00                                 |
+-----------------------------------------------------+
1 row in set (0.00 sec)
```

DATE_FORMAT(date,format)

This command formats the date value as per the format string.

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

Sr.No.	Specifier & Description
1	**%a** Abbreviated weekday name (Sun..Sat)
2	**%b** Abbreviated month name (Jan..Dec)
3	**%c** Month, numeric (0..12)
4	**%D** Day of the month with English suffix (0th, 1st, 2nd, 3rd, .)
5	**%d** Day of the month, numeric (00..31)
6	**%e** Day of the month, numeric (0..31)
7	**%f** Microseconds (000000..999999)
8	**%H** Hour (00..23)
9	**%h** Hour (01..12)
10	**%I** Hour (01..12)
11	**%i**

	Minutes, numeric (00..59)
12	**%j** Day of year (001..366)
13	**%k** Hour (0..23)
14	**%l** Hour (1..12)
15	**%M** Month name (January..December)
16	**%m** Month, numeric (00..12)
17	**%p** AM or PM
18	**%r** Time, 12-hour (hh:mm:ss followed by AM or PM)
19	**%S** Seconds (00..59)
20	**%s** Seconds (00..59)
21	**%T** Time, 24-hour (hh:mm:ss)
22	**%U**

	Week (00..53), where Sunday is the first day of the week
23	**%u** Week (00..53), where Monday is the first day of the week
24	**%V** Week (01..53), where Sunday is the first day of the week; used with %X
25	**%v** Week (01..53), where Monday is the first day of the week; used with %x
26	**%W** Weekday name (Sunday..Saturday)
27	**%w** Day of the week (0=Sunday..6=Saturday)
28	**%X** Year for the week where Sunday is the first day of the week, numeric, four digits; used with %V
29	**%x** Year for the week, where Monday is the first day of the week, numeric, four digits; used with %v
30	**%Y** Year, numeric, four digits
31	**%y** Year, numeric (two digits)
32	**%%** A literal .%. character

33	**%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)
```

DATE_SUB(date,INTERVAL expr unit)

This is similar to the DATE_ADD() function.

DAY(date)

The DAY() is a synonym for the DAYOFMONTH() function.

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)
```

EXTRACT(unit FROM date)

The EXTRACT() function uses the same kinds of unit specifiers as DATE_ADD() or DATE_SUB(), but extracts parts from the date rather than performing date arithmetic.

```
mysql> SELECT EXTRACT(YEAR FROM '1999-07-02');
+-----------------------------------------------------------+
| EXTRACT(YEAR FROM '1999-07-02')                           |
+-----------------------------------------------------------+
| 1999                                                      |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

```
mysql> SELECT EXTRACT(YEAR_MONTH FROM '1999-07-02 01:02:03');
+-----------------------------------------------------+
| EXTRACT(YEAR_MONTH FROM '1999-07-02 01:02:03')      |
+-----------------------------------------------------+
| 199907                                              |
+-----------------------------------------------------+
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)
```

Note − 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).

FROM_UNIXTIME(unix_timestamp)

FROM_UNIXTIME(unix_timestamp,format)

Returns a representation of the **unix_timestamp** argument as a value in 'YYYY-MM-DD HH:MM:SS or YYYYMMDDHHMMSS format, depending on whether the function is used in a string or in a numeric context. The value is expressed in the current time zone. The unix_timestamp argument is an internal timestamp values, which are produced by the **UNIX_TIMESTAMP()** function.

If the format is given, the result is formatted according to the format string, which is used in the same way as is listed in the entry for the **DATE_FORMAT()** function.

```
mysql> SELECT FROM_UNIXTIME(875996580);
+-------------------------------------------------------+
| FROM_UNIXTIME(875996580)                              |
+-------------------------------------------------------+
| 1997-10-04 22:23:00                                   |
+-------------------------------------------------------+
1 row in set (0.00 sec)
```

HOUR(time)

Returns the hour for 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().

LOCALTIMESTAMP and LOCALTIMESTAMP()

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

MAKEDATE(year,dayofyear)

Returns a date, given year and day-of-year values. The dayofyear value must be greater than 0 or the result will be 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)
```

MAKETIME(hour,minute,second)

Returns a time value calculated from the hour, minute and second arguments.

```
mysql> SELECT MAKETIME(12,15,30);
+-----------------------------------------------------------+
| MAKETIME(12,15,30)                                        |
+-----------------------------------------------------------+
| '12:15:30'                                                |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

MICROSECOND(expr)

Returns the microseconds from the time or datetime expression (expr) as a number in the range from 0 to 999999.

```
mysql> SELECT MICROSECOND('12:00:00.123456');
+-----------------------------------------------------------+
| MICROSECOND('12:00:00.123456')                            |
+-----------------------------------------------------------+
| 123456                                                    |
+-----------------------------------------------------------+
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 a 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. This 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)
```

PERIOD_ADD(P,N)

Adds N months to a period P (in the format YYMM or YYYYMM). Returns a value in the format YYYYMM. Note that the period argument P is not a date value.

```
mysql> SELECT PERIOD_ADD(9801,2);
+-------------------------------------------------------+
| PERIOD_ADD(9801,2)                                    |
+-------------------------------------------------------+
| 199803                                                |
+-------------------------------------------------------+
1 row in set (0.00 sec)
```

PERIOD_DIFF(P1,P2)

Returns the number of months between periods P1 and P2. These periods P1 and P2 should be in the format YYMM or YYYYMM. Note that the period arguments P1 and P2 are not date values.

```
mysql> SELECT PERIOD_DIFF(9802,199703);
+--------------------------------------------------------+
| PERIOD_DIFF(9802,199703)                               |
+--------------------------------------------------------+
| 11                                                     |
+--------------------------------------------------------+
1 row in set (0.00 sec)
```

QUARTER(date)

Returns the quarter of the year for date, in the range 1 to 4.

```
mysql> SELECT QUARTER('98-04-01');
+--------------------------------------------------------+
| QUARTER('98-04-01')                                    |
+--------------------------------------------------------+
| 2                                                      |
+--------------------------------------------------------+
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. The STR_TO_DATE() function returns a DATETIME value if the format string contains both date and time parts. Else, it returns 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)
```

SUBDATE(date,INTERVAL expr unit) and SUBDATE(expr,days)

When invoked with the INTERVAL form of the second argument, SUBDATE() is a synonym for DATE_SUB(). For information on the INTERVAL unit argument, see the discussion for DATE_ADD().

```
mysql> SELECT DATE_SUB('1998-01-02', INTERVAL 31 DAY);
+------------------------------------------------------------+
| DATE_SUB('1998-01-02', INTERVAL 31 DAY)                    |
+------------------------------------------------------------+
| 1997-12-02                                                 |
+------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT SUBDATE('1998-01-02', INTERVAL 31 DAY);
+------------------------------------------------------------+
| SUBDATE('1998-01-02', INTERVAL 31 DAY)                     |
+------------------------------------------------------------+
| 1997-12-02                                                 |
+------------------------------------------------------------+
1 row in set (0.00 sec)
```

SUBTIME(expr1,expr2)

The SUBTIME() function returns expr1 . expr2 expressed as a value in the same format as expr1. The expr1 value is a time or a datetime expression, while the expr2 value is a time expression.

```
mysql> SELECT SUBTIME('1997-12-31 23:59:59.999999',
    -> '1 1:1:1.000002');
+-----------------------------------------------------------+
| SUBTIME('1997-12-31 23:59:59.999999'...                   |
+-----------------------------------------------------------+
| 1997-12-30 22:58:58.999997                                |
+-----------------------------------------------------------+
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 in a numeric context.

```
mysql> SELECT SYSDATE();
+-----------------------------------------------------------+
| SYSDATE()                                                 |
+-----------------------------------------------------------+
| 2006-04-12 13:47:44                                       |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

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)
```

TIMEDIFF(expr1,expr2)

The TIMEDIFF() function returns expr1 . expr2 expressed as a time value. These expr1 and expr2 values are time or date-and-time expressions, but both must be of the same type.

```
mysql> SELECT TIMEDIFF('1997-12-31 23:59:59.000001',
    -> '1997-12-30 01:01:01.000002');
```

```
+----------------------------------------------------------+
| TIMEDIFF('1997-12-31 23:59:59.000001'.....               |
+----------------------------------------------------------+
|  46:58:57.999999                                         |
+----------------------------------------------------------+
1 row in set (0.00 sec)
```

TIMESTAMP(expr), TIMESTAMP(expr1,expr2)

With a single argument, this function returns the date or datetime expression expr as a datetime value. With two arguments, it adds the time expression expr2 to the date or datetime expression **expr1** and returns the result as a datetime value.

```
mysql> SELECT TIMESTAMP('2003-12-31');
+----------------------------------------------------------+
| TIMESTAMP('2003-12-31')                                  |
+----------------------------------------------------------+
| 2003-12-31 00:00:00                                      |
+----------------------------------------------------------+
1 row in set (0.00 sec)
```

TIMESTAMPADD(unit,interval,datetime_expr)

This function adds the integer expression interval to the date or datetime expression **datetime_expr**. The unit for interval is given by the unit argument, which should be one of the following values −

- FRAC_SECOND
- SECOND, MINUTE
- HOUR, DAY
- WEEK
- MONTH
- QUARTER or
- YEAR

The unit value may be specified using one of the keywords as shown or with a prefix of SQL_TSI_.

For example, DAY and SQL_TSI_DAY both are legal.

```
mysql> SELECT TIMESTAMPADD(MINUTE,1,'2003-01-02');
+----------------------------------------------------------+
| TIMESTAMPADD(MINUTE,1,'2003-01-02')                      |
+----------------------------------------------------------+
| 2003-01-02 00:01:00                                      |
```

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

TIMESTAMPDIFF(unit,datetime_expr1,datetime_expr2)

Returns the integer difference between the date or datetime expressions datetime_expr1 and datetime_expr2. The unit for the result is given by the unit argument. The legal values for the unit are the same as those listed in the description of the TIMESTAMPADD() function.

```
mysql> SELECT TIMESTAMPDIFF(MONTH,'2003-02-01','2003-05-01');
+-----------------------------------------------------+
| TIMESTAMPDIFF(MONTH,'2003-02-01','2003-05-01')      |
+-----------------------------------------------------+
| 3                                                   |
+-----------------------------------------------------+
1 row in set (0.00 sec)
```

TIME_FORMAT(time,format)

This function is used like the DATE_FORMAT() function, but the format string may contain format specifiers only for hours, minutes and seconds.

If the time value contains an hour part that is greater than 23, the %**H** and %**k** hour format specifiers produce a value larger than the usual range of 0 to 23. The other hour format specifiers produce the hour value modulo 12.

```
mysql> SELECT TIME_FORMAT('100:00:00', '%H %k %h %I %l');
+-----------------------------------------------------+
| TIME_FORMAT('100:00:00', '%H %k %h %I %l')          |
+-----------------------------------------------------+
| 100 100 04 04 4                                     |
+-----------------------------------------------------+
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)
```

UNIX_TIMESTAMP(), UNIX_TIMESTAMP(date)

If called with no argument, this function returns a Unix timestamp (seconds since '1970-01-01 00:00:00' UTC) as an unsigned integer. If UNIX_TIMESTAMP() is called with a date argument, it returns the value of the argument as seconds since '1970-01-01 00:00:00' UTC. date may be a DATE string, a DATETIME string, a TIMESTAMP, or a number in the format YYMMDD or YYYYMMDD.

```
mysql> SELECT UNIX_TIMESTAMP();
+--------------------------------------------------------+
| UNIX_TIMESTAMP()                                       |
+--------------------------------------------------------+
| 882226357                                              |
+--------------------------------------------------------+
1 row in set (0.00 sec)

mysql> SELECT UNIX_TIMESTAMP('1997-10-04 22:23:00');
+--------------------------------------------------------+
| UNIX_TIMESTAMP('1997-10-04 22:23:00')                  |
+--------------------------------------------------------+
| 875996580                                              |
+--------------------------------------------------------+
1 row in set (0.00 sec)
```

UTC_DATE, UTC_DATE()

Returns the current UTC 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 UTC_DATE(), UTC_DATE() + 0;
+--------------------------------------------------------+
| UTC_DATE(), UTC_DATE() + 0                             |
+--------------------------------------------------------+
| 2003-08-14, 20030814                                   |
```

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

UTC_TIME, UTC_TIME()

Returns the current UTC time 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 UTC_TIME(), UTC_TIME() + 0;
+-----------------------------------------------------+
| UTC_TIME(), UTC_TIME() + 0                          |
+-----------------------------------------------------+
| 18:07:53, 180753                                    |
+-----------------------------------------------------+
1 row in set (0.00 sec)
```

UTC_TIMESTAMP, UTC_TIMESTAMP()

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

```
mysql> SELECT UTC_TIMESTAMP(), UTC_TIMESTAMP() + 0;
+-----------------------------------------------------+
| UTC_TIMESTAMP(), UTC_TIMESTAMP() + 0                |
+-----------------------------------------------------+
| 2003-08-14 18:08:04, 20030814180804                 |
+-----------------------------------------------------+
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 a Sunday or a 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)
```

YEARWEEK(date), YEARWEEK(date,mode)

Returns the year and the week for a date. The mode argument works exactly like the mode argument to the WEEK() function. The year in the result may be different from the year in the date argument for the first and the last week of the year.

```
mysql> SELECT YEARWEEK('1987-01-01');
+-----------------------------------------------------------+
| YEAR('98-02-03')YEARWEEK('1987-01-01')                    |
+-----------------------------------------------------------+
| 198653                                                    |
+-----------------------------------------------------------+
1 row in set (0.00 sec)
```

Note − The week number is different from what the WEEK() function would return (0) for optional arguments 0 or 1, as WEEK() then returns the week in the context of the given year.