SQL DATE Type
Type Conversion

Date/Time Types

MySQL supports four[1] different date/time related types.

- DATE
- DATETIME
- TIMESTAMP
- TIME

DATE

The DATE type is used for values with a date part but no time part eg. April 14th 2013. MySQL retrieves and displays DATE values in 'YYYY-MM-DD' format. The DATE type has a range of '1000-01-01' to '9999-12-31'.

DATETIME

The DATETIME type is used for values that contain both date and time parts. MySQL retrieves and displays DATETIME values in 'YYYY-MM-DD HH:MM:SS' format. The DATETIME has a range of '1000-01-01 00:00:00' to '9999-12-31 23:59:59'.

TIMESTAMP

The TIMESTAMP data type is used for values that contain both date and time parts. TIMESTAMP has a range of '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC.

Difference Between TIMESTAMP and DATETIME

MySQL converts TIMESTAMP values from the current time zone to UTC for storage, and back from UTC to the current time zone for retrieval. (This does not occur for other types such as DATETIME.) By default, the current time zone for each connection is the server's time. The time zone can be set on a per-connection basis. As long as the time zone setting remains constant, you get back the same value you store. If you store a TIMESTAMP value, and then change the time zone and retrieve the value, the retrieved value is different from the value you stored. This occurs because the same time zone was not used for conversion in both directions. The current time zone is available as the value of the time_zone system variable.

[1] Actually five. There is a YEAR type that we will not discuss.

TIME

MySQL retrieves and displays TIME values in 'HH:MM:SS' format (or 'HHH:MM:SS' format for large hours values). TIME values may range from '-838:59:59' to '838:59:59'. The hours part may be so large because the TIME type can be used not only to represent a time of day (which must be less than 24 hours), but also elapsed time or a time interval between two events (which may be much greater than 24 hours, or even negative).

Date and Time Functions

MySQL has a large selection of function to manipulate temporal values. We will only be discussing a small subset, but the entire reference can be found here: `http://dev.mysql.com/doc/refman/5.1/en/date-and-time-functions.html`

DATE(str)
: Parses a string and returns a DATE. Note that any time portion of the string is ignored.

```
MariaDB [(none)]> SELECT DATE('2003-12-31 01:02:03');
+----------------------------+
| DATE('2003-12-31 01:02:03') |
+----------------------------+
| 2003-12-31                 |
+----------------------------+
```

TIME(str)
: Parses a string an returns a TIME. Note that any date portion of the string is ignored.

```
MariaDB [(none)]> SELECT TIME('2003-12-31 01:02:03');
+----------------------------+
| TIME('2003-12-31 01:02:03') |
+----------------------------+
| 01:02:03                   |
+----------------------------+
```

TIMESTAMP(str)
: Parses a string an returns a TIMESTAMP. Note that there is no "DATETIME" function, only this. If you want to parse a DATETIME, you will need to use the complete STR_TO_DATE function.

```
MariaDB [(none)]> SELECT TIMESTAMP('2003-12-31 01:02:03');
+---------------------------------+
| TIMESTAMP('2003-12-31 01:02:03') |
+---------------------------------+
| 2003-12-31 01:02:03             |
+---------------------------------+
```

STR_TO_DATE(str, format)
: The all purpose String to DATE/DATETIME/TIME converting function. This function takes a string representing the date and a format string defining the format of the date string. The format specifiers are defined later in this handout. This function will convert to either a DATE (if no time portion is specified), TIME (if no date portion is specified), or DATETIME (if both date and time are specified).

```
MariaDB [(none)]> SELECT STR_TO_DATE('01,5,2013','%d,%m,%Y');
+-----------------------------------+
| STR_TO_DATE('01,5,2013','%d,%m,%Y') |
+-----------------------------------+
| 2013-05-01                        |
+-----------------------------------+


MariaDB [(none)]> SELECT STR_TO_DATE('May 1, 2013','%M %d,%Y');
+--------------------------------------+
| STR_TO_DATE('May 1, 2013','%M %d,%Y') |
+--------------------------------------+
| 2013-05-01                           |
+--------------------------------------+
```

DATE_FORMAT(date, format)

The opposite of STR_TO_DATE(). Takes in a date, and returns a string that is formatted according to the format string/

```
MariaDB [(none)]> SELECT DATE_FORMAT(TIMESTAMP('2009-10-04 22:23:00'),
                                     '%W %M %Y');
+-----------------------------------------------------------+
| DATE_FORMAT(TIMESTAMP('2009-10-04 22:23:00'), '%W %M %Y') |
+-----------------------------------------------------------+
| Sunday October 2009                                       |
+-----------------------------------------------------------+


MariaDB [(none)]> SELECT DATE_FORMAT(TIMESTAMP('1997-10-04 22:23:00'),
                                     '%H %k %I %r %T %S %w');
+------------------------------------------------------------------------+
| DATE_FORMAT(TIMESTAMP('1997-10-04 22:23:00'), '%H %k %I %r %T %S %w') |
+------------------------------------------------------------------------+
| 22 22 10 10:23:00 PM 22:23:00 00 6                                     |
+------------------------------------------------------------------------+
```

TIME_FORMAT(time, format)

Like DATE_FORMAT(), but the format string may contain format specifiers only for hours, minutes, seconds, and microseconds.

```
MariaDB [(none)]> 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                           |
+-------------------------------------------+
```

SEC_TO_TIME(secs)

Returns the seconds argument, converted to hours, minutes, and seconds, as a TIME value.

```
MariaDB [(none)]> SELECT SEC_TO_TIME(2378);
+------------------+
| SEC_TO_TIME(2378) |
+------------------+
| 00:39:38         |
+------------------+
```

UNIX_TIMESTAMP(), UNIX_TIMESTAMP(date)

If called with no argument, returns the current Unix timestamp (seconds since '1970-01-01 00:00:00' UTC) as an unsigned integer. If called with a DATE/DATETIME/TIMESTAMP as an argument, then UNIX_TIMESTAMP() returns the Unix timestamp representation of that time.

```
MariaDB [(none)]> SELECT UNIX_TIMESTAMP();
+------------------+
| UNIX_TIMESTAMP() |
+------------------+
|       1365789827 |
+------------------+


SELECT UNIX_TIMESTAMP(TIMESTAMP('2007-11-30 10:30:19'));
+-------------------------------------------------+
| UNIX_TIMESTAMP(TIMESTAMP('2007-11-30 10:30:19')) |
+-------------------------------------------------+
|                                      1196447419 |
+-------------------------------------------------+
```

FROM_UNIXTIME(unix_timestamp), FROM_UNIXTIME(unix_timestamp, format)

Returns a string representing the given Unix timestamp. If no format string is specified, then the time is formated as the default DATETIME format ('YYYY-MM-DD HH:MM:SS').

```
MariaDB [(none)]> SELECT FROM_UNIXTIME(1196440219);
+---------------------------+
| FROM_UNIXTIME(1196440219) |
+---------------------------+
| 2007-11-30 08:30:19       |
+---------------------------+


MariaDB [(none)]> SELECT FROM_UNIXTIME(1196440219, '%Y %D %M %h:%i:%s %x');
+-------------------------------------------------+
| FROM_UNIXTIME(1196440219, '%Y %D %M %h:%i:%s %x') |
+-------------------------------------------------+
| 2007 30th November 08:30:19 2007                |
+-------------------------------------------------+
```

NOW()

Return the current date and time. The returned format depends on the context that it is used in.

When used in a String or default context, it will be returned as the default TIMESTAMP format ('YYYY-MM-DD HH:MM:SS'). When used in a numeric context, it will be returned as a number formatted as: YYYYMMDDHHMMSS.uuuuuu.

```
MariaDB [(none)]> SELECT NOW();
+---------------------+
| NOW()               |
+---------------------+
| 2013-04-12 10:46:11 |
+---------------------+


MariaDB [(none)]> SELECT NOW() + 0;
+----------------+
| NOW() + 0      |
+----------------+
| 20130412104615 |
+----------------+
```

Format Specifications

A *date format specification* is a sting which includes special identifiers for various components of the temporal type. The table of formatting components is provided for your convenience below.

specification	explanation
%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

References

http://dev.mysql.com/doc/refman/4.1/en/datetime.html
http://dev.mysql.com/doc/refman/4.1/en/time.html
http://dev.mysql.com/doc/refman/5.1/en/date-and-time-functions.html
http://dev.mysql.com/doc/refman/5.1/en/date-and-time-functions.html#function_date-format