

# SQLITE - DATE & TIME

[http://www.tutorialspoint.com/sqlite/sqlite\\_date\\_time.htm](http://www.tutorialspoint.com/sqlite/sqlite_date_time.htm)

Copyright © tutorialspoint.com

SQLite supports five date and time functions as follows:

S.N.	Function	Example
1	<code>datetimestring, modifiers. . .</code>	This returns the date in this format: YYYY-MM-DD
2	<code>timetimestring, modifiers. . .</code>	This returns the time as HH:MM:SS
3	<code>datetimetimestring, modifiers. . .</code>	This returns YYYY-MM-DD HH:MM:SS
4	<code>juliandaytimestring, modifiers. . .</code>	This returns the number of days since noon in Greenwich on November 24, 4714 B.C.
5	<code>strftimetimestring, modifiers. . .</code>	This returns the date formatted according to the format string specified as the first argument formatted as per formatters explained below.

All the above five date and time functions take a time string as an argument. The time string is followed by zero or more modifiers. The `strftime` function also takes a format string as its first argument. Following section will give you detail on different types of time strings and modifiers.

## Time Strings:

A time string can be in any of the following formats:

S.N.	Time String	Example
1	YYYY-MM-DD	2010-12-30
2	YYYY-MM-DD HH:MM	2010-12-30 12:10
3	YYYY-MM-DD HH:MM:SS.SSS	2010-12-30 12:10:04.100
4	MM-DD-YYYY HH:MM	30-12-2010 12:10
5	HH:MM	12:10
6	YYYY-MM-DDTHH:MM	2010-12-30 12:10
7	HH:MM:SS	12:10:01
8	YYYYMMDD HHMMSS	20101230 121001
9	now	2013-05-07

You can use the "T" as a literal character separating the date and the time.

## Modifiers

The time string can be followed by zero or more modifiers that will alter date and/or time returned by any of the above five functions. Modifiers are applied from left to right and following modifiers are available in SQLite:

- NNN days
- NNN hours

- NNN minutes
- NNN.NNNN seconds
- NNN months
- NNN years
- start of month
- start of year
- start of day
- weekday N
- unixepoch
- localtime
- utc

## Formatters:

SQLite provides very handy function **strftime** to format any date and time. You can use following substitutions to format your date and time:

Substitution	Description
%d	Day of month, 01-31
%f	Fractional seconds, SS.SSS
%H	Hour, 00-23
%j	Day of year, 001-366
%J	Julian day number, DDDD.DDDD
%m	Month, 00-12
%M	Minute, 00-59
%s	Seconds since 1970-01-01
%S	Seconds, 00-59
%w	Day of week, 0-6 <i>0 is Sunday</i>
%W	Week of year, 01-53
%Y	Year, YYYY
%%	% symbol

## Examples

Let's try various examples now using SQLite prompt. Following computes the current date:

```
sqlite> SELECT date('now');
2013-05-07
```

Following computes the last day of the current month:

```
sqlite> SELECT date('now', 'start of month', '+1 month', '-1 day');  
2013-05-31
```

Following computes the date and time for a given UNIX timestamp 1092941466:

```
sqlite> SELECT datetime(1092941466, 'unixepoch');  
2004-08-19 18:51:06
```

Following computes the date and time for a given UNIX timestamp 1092941466 and compensate for your local timezone:

```
sqlite> SELECT datetime(1092941466, 'unixepoch', 'localtime');  
2004-08-19 13:51:06
```

Following computes the current UNIX timestamp:

```
sqlite> SELECT strftime('%s', 'now');  
1393348134
```

Following computes the number of days since the signing of the US Declaration of Independence:

```
sqlite> SELECT julianday('now') - julianday('1776-07-04');  
86798.7094695023
```

Following computes the number of seconds since a particular moment in 2004:

```
sqlite> SELECT strftime('%s', 'now') - strftime('%s', '2004-01-01 02:34:56');  
295001572
```

Following computes the date of the first Tuesday in October for the current year:

```
sqlite> SELECT date('now', 'start of year', '+9 months', 'weekday 2');  
2013-10-01
```

Following computes the time since the UNIX epoch in seconds *like* `strftime()` except includes fractional part):

```
sqlite> SELECT (julianday('now') - 2440587.5)*86400.0;  
1367926077.12598
```

To convert between UTC and local time values when formatting a date, use the `utc` or `localtime` modifiers as follows:

```
sqlite> SELECT time('12:00', 'localtime');  
05:00:00
```

```
sqlite> SELECT time('12:00', 'utc');  
19:00:00
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js