Date And Time Functions

SQLite supports five date and time functions as follows:

1. `date(`*`timestring, modifier, modifier, ...`*`)`

   The date() function returns the date in this format: YYYY-MM-DD.
2. `time(`*`timestring, modifier, modifier, ...`*`)`

   The time() function returns the time as HH:MM:SS.
3. `datetime(`*`timestring, modifier, modifier, ...`*`)`

   The datetime() function returns "YYYY-MM-DD HH:MM:SS".
4. `julianday(`*`timestring, modifier, modifier, ...`*`)`

   The julianday() function returns the `Julian day` - the number of days since noon in Greenwich on November 24, 4714 B.C. (`Proleptic Gregorian calendar`).
5. `strftime(`*`format, timestring, modifier, modifier, ...`*`)`

   The strftime() routine returns the date formatted according to the format string specified as the first argument. The format string supports the most common substitutions found in the `strftime() function` from the standard C library plus two new substitutions, %f and %J.

Examples:

1. Compute the current date:
```
sqlite> SELECT date('now');
output: 2017-09-25
```
2. Compute the last day of the current month:
```
sqlite> SELECT date('now','start of month','+1 month','-1
day');
output: 2017-09-30
```
3. Compute the current date and time with desired format:
```
SELECT strftime(format, 'now'):
SELECT strftime('%Y-%m-%d','now');
OUTPUT: 2017-09-25
SELECT strftime('%Y-%m-%d %H-%M','now');
OUTPUT: 2017-09-25 20-36
SELECT strftime('%Y-%m-%d %H-%M-%S','now');
```

```
OUTPUT: 2017-09-25 20-36-57
```
4. Extract the current year or month from system date:
```
SELECT strftime('%Y','now');
OUTPUT: 2017
SELECT strftime('%m','now');
OUTPUT: 09
```
5. Compute the date of the first Tuesday in October for the current year.
```
sqlite> SELECT date('now','start of year','+9 months','weekday
2');
OUTPUT: 2017-10-03
```
6. Compute the number of seconds since a particular moment in 2004.
```
sqlite> SELECT strftime('%s','now') - strftime('%s','2004-01-01
02:34:56');
OUTPUT: 433448465
```
7. Convert between UTC and local time values when formatting a date, use the utc or localtime modifiers:
```
sqlite> SELECT time('12:00', 'localtime');
OUTPUT: 05:00:00
sqlite> SELECT time('12:00', 'utc');
OUTPUT: 19:00:00
```
8. Some extra examples with datetime function:

a. Working with years:
```
    sqlite> SELECT datetime('2012-10-23 09:23:10','-2
    years');
    OUTPUT: 2010-10-23 09:23:10
```
b. Working with days:
```
    sqlite> SELECT datetime('2014-10-23','+7 days');
    OUTPUT: 2014-10-30 00:00:00
```
c. Working with hours:
```
    SELECT datetime('2014-10-23 11:23:02','+2 hours');
    OUTPUT: 2014-10-23 13:23:02
```
d. Working with minutes:
```
    sqlite> SELECT datetime('2014-10-23 11:15:02','+15
    minutes');
    OUTPUT: 2014-10-23 11:30:02
```

9. Some extra examples with date function:

a. Working with days:
```
    sqlite> SELECT date('2014-10-16', 'start of month');
    OUTPUT: 2014-10-01

    sqlite> SELECT date('2014-10-16', '-10 days');
```

```
OUTPUT: 2014-10-06
```

b. Working with years:

```
sqlite> SELECT date('2014-10-16','+2 years');
OUTPUT: 2016-10-16
```

10. Some extra examples with time function:
    a. Working with hours:

    ```
    sqlite> SELECT time('11:23:02','-2 hours');
    OUTOUT: 09:23:02
    ```

    b. Working with minutes:

    ```
    sqlite> SELECT time('11:15:02','-15 minutes');
    OUTPUT: 11:00:02
    ```

    c. Working with hours, minutes and seconds:

    ```
    sqlite> SELECT time('11:15:02','-2 hours','-15
    minutes','+13 seconds');
    OUTPUT: 09:00:15
    ```