

# 5

## Datum & Zeitstempel in SQLite

Datentypen DATE und DATETIME können aus Strings erstellt und direkt modifiziert werden

```
DATE('2022-07-19')  
DATE('now', '-1 day')  
DATE('now', '+1 year')  
DATE('now', '+1 month', '-1 year')
```

Achtung: In anderen SQL-Dialekten (z.B. PostgreSQL) gibt es andere Funktionen

```
DATETIME('2022-07-19 12:15')  
DATETIME('2022-07-19 12:15:12')  
DATETIME('2022-07-19 12:15:12.123', '-12 hours')  
DATETIME('now', '+3 minutes')  
DATETIME('now', 'localtime')
```

Modifizierer	Beschreibung
'XXX days'	addiert/subtrahiert XXX Tage
'XXX hours'	addiert/subtrahiert XXX Stunden
'XXX minutes'	addiert/subtrahiert XXX Minuten
'XXX.XXXX seconds'	addiert/subtrahiert XXX Sekunden
'XXX months'	addiert/subtrahiert XXX Monate
'XXX years'	addiert/subtrahiert XXX Jahre
'start of month'	zurück zum Monatsanfang
'start of year'	zurück zum Jahresanfang
'start of day'	zurück zum Tagesanfang
'weekday N'	nächstes Datum, das der Wochentag ist (0=Sonntag, 1=Montag, ...)
'unixepoch'	Unix Zeit
'localtime'	lokale Zeit
'utc'	UTC Zeit

Differenz zwischen zwei Daten (exkl. beider Tage):

```
julianday(arrival) - julianday(departure) AS date_difference
```

Differenz zwischen zwei Zeitstempeln in Minuten:

```
CAST((julianday(arrival) - julianday(departure))*24*60 AS real) AS delta_min
```

Umwandeln in einen String mittels  
`strftime(format, datum)`

```
SELECT strftime('%s', 'now');  
SELECT strftime('%d.%m.%Y', 'now');  
SELECT strftime('%H:%M', 'now');  
SELECT strftime('%w', 'now');
```

Kürzel	Bedeutung
%Y	vierstelliges Jahr
%m	zweistelliger Monat (01-12)
%d	zweistelliger Tag (01-31)
%H	zweistellige Stunde (00-24)
%M	zweistellige Minute (00-59)
%S	zweistellige Sekunde (00-59)
%s	Sekunden seit 01.01.1970
%j	Tag des Jahres (001-366)
%w	Wochentag (0 = Sonntag, 6 = Samstag)
%W	Kalenderwoche (00-53)