

3.1

Date Functions

Current Date/Time Functions

- There are a number of functions that give the current date and time. The DATE() function is a date formatting function, but I include it in the list because it is often confused with the NOW() function
- CURRENT_DATE, CURRENT_TIME, UTC_DATE, UTC_TIME can be used with the parentheses “()” or not. They accept no parameters

Table 1. Current Date Functions

Function	Type	Example	Result
NOW() * Returns current local date and time.	date/time	NOW()	ex. '2020-02-24 09:31:31'
DATE(date) * extracts the date from input. If time is included, the time is dropped.	date/time	DATE('2020-01-01 11:31:31')	'2020-02-24'
CURRENT_DATE() * Returns current local date	date	CURRENT_DATE	'2020-02-24'
CURRENT_TIME() * Returns current local time.	time	CURRENT_TIME	'11:52:10'
UTC_DATE() * Returns current UTC date.	date	UTC_DATE	'2020-02-24'
UTC_TIME() * Returns current UTC date.	time	UTC_TIME	'18:52:10'

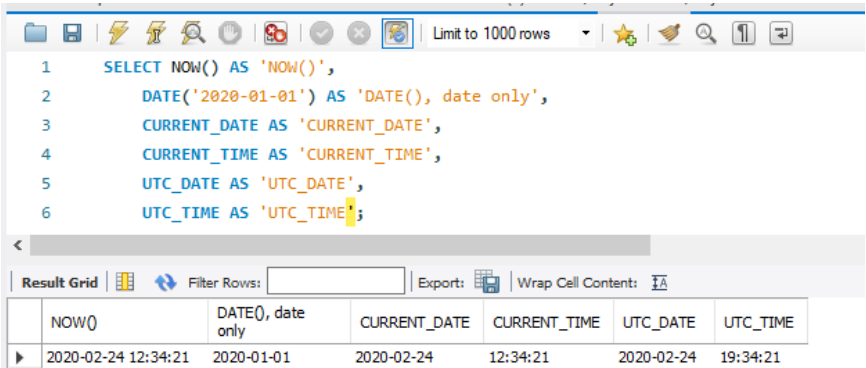
SELECT NOW() AS 'NOW() ',

```

DATE('2020-01-01') AS 'DATE()', date only',
CURRENT_DATE AS 'CURRENT_DATE',
CURRENT_TIME AS 'CURRENT_TIME',
UTC_DATE AS 'UTC_DATE',
UTC_TIME AS 'UTC_TIME';

```

Results:



DATE_ADD

- Returns a date with a DATE or DATETIME value equal to the original value plus the specified interval.

Table 2. DATE_ADD Function

Function	Type	Example	Result
DATE_ADD(date, interval expression unit)	DATE, DATETIME	DATE_ADD('2020-01-01', INTERVAL 1 DAY)	'202-01-02'

Code Snippet:

```

USE bike;
SELECT order_date,

```

```

DATE_ADD(order_date, INTERVAL 1 DAY) AS 'ORDER
DATE PLUS 1 day',
DATE_ADD(order_date, INTERVAL 6 MONTH) AS 'ORDER
DATE PLUS 6 months',
DATE_ADD(order_date, INTERVAL '2 12' DAY_HOUR)
AS 'ORDER DATE PLUS 2 days 1 hour'
FROM cust_order;

```

Results:

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, search, and execution, along with a 'Limit to 1000 rows' dropdown. The query editor contains the following SQL code:

```

1 • USE bike;
2 • SELECT order_date,
3     DATE_ADD(order_date, INTERVAL 1 DAY)
4     AS 'ORDER DATE PLUS 1 day',
5     DATE_ADD(order_date, INTERVAL 6 MONTH)
6     AS 'ORDER DATE PLUS 6 months',
7     DATE_ADD(order_date, INTERVAL '2 12' DAY_HOUR)
8     AS 'ORDER DATE PLUS 2 days 1 hour'
9 FROM cust_order;

```

Below the query editor is a 'Result Grid' showing the output of the query. The grid has four columns: 'order_date', 'ORDER DATE PLUS 1 day', 'ORDER DATE PLUS 6 months', and 'ORDER DATE PLUS 2 days 1 hour'. The first row is selected, showing the date 2016-01-01 and its corresponding values for the three date calculations.

order_date	ORDER DATE PLUS 1 day	ORDER DATE PLUS 6 months	ORDER DATE PLUS 2 days 1 hour
2016-01-01	2016-01-02	2016-07-01	2016-01-03 12:00:00
2016-01-01	2016-01-02	2016-07-01	2016-01-03 12:00:00
2016-01-02	2016-01-03	2016-07-02	2016-01-04 12:00:00
2016-01-03	2016-01-04	2016-07-03	2016-01-05 12:00:00
2016-01-03	2016-01-04	2016-07-03	2016-01-05 12:00:00

DATE_FORMAT

- Dates must be enclosed in quotes
- You can pass a DATE or DATETIME datatype to DATE_FORMAT

Table 3. DATE_FORMAT Function

Function	Type	Example	Result
DATE_FORMAT	DATE	DATE_FORMAT('2020-09-03', '%m/%d/%y')	09/03/14

Code Snippet:

```
USE world;
SELECT name, continent, DATE_FORMAT('2020-01-28',
'%m/%d/%y')
FROM country;
```

Results:

The screenshot shows a MySQL database client interface. The SQL editor contains the following query:

```
1 USE world;
2 SELECT name, continent, DATE_FORMAT('2020-01-28', '%m/%d/%y')
3 FROM country;
4
```

Below the editor, the results are displayed in a grid format. The grid has four columns: name, continent, and DATE_FORMAT('2020-01-28', '%m/%d/%y'). The results are as follows:

	name	continent	DATE_FORMAT('2020-01-28', '%m/%d/%y')
▶	Aruba	North America	01/28/20
	Afghanistan	Asia	01/28/20
	Angola	Africa	01/28/20
	Anguilla	North America	01/28/20
	Albania	Europe	01/28/20
	Andorra	Europe	01/28/20
	Netherlands Antilles	North America	01/28/20
	United Arab Emirates	Asia	01/28/20
	Argentina	South America	01/28/20
	Armenia	Asia	01/28/20

Table 4. Format List

Specifier	Description
%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; WEEK() mode 0
%u	Week (00..53), where Monday is the first day of the week; WEEK() mode 1
%V	Week (01..53), where Sunday is the first day of the week; WEEK() mode 2; used with %X
%v	Week (01..53), where Monday is the first day of the week; WEEK() mode 3; 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
%x	x, for any "x" not listed above

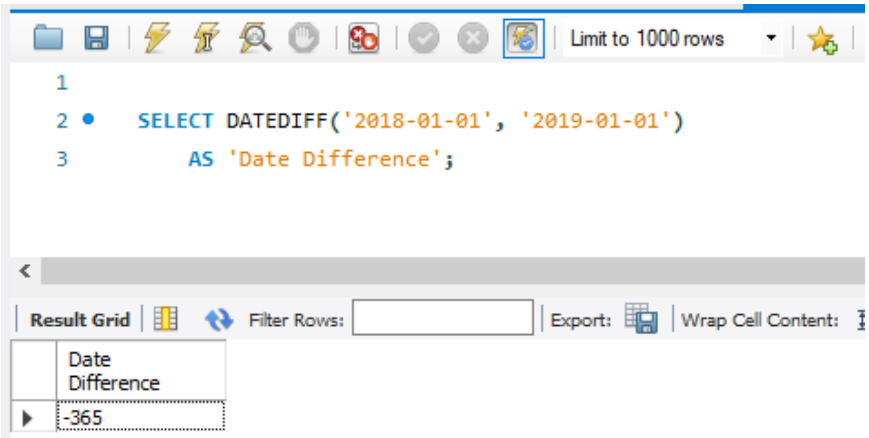
DATEDIFF

- The DATEDIFF function has two parameters. Both are dates.
- The value returned by the function is an integer and is the number of days between the two dates.
- If you provide the latest date, first the results will be positive. Otherwise, it will be negative.

Example:

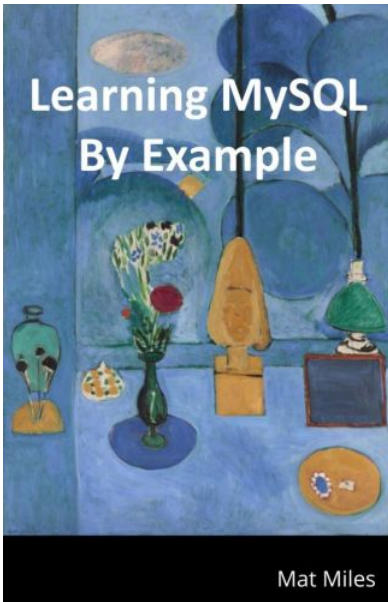
```
SELECT DATEDIFF('2018-01-01', '2019-01-01')  
AS 'Date Difference';
```

Results:



The screenshot shows a MySQL query editor interface. The query is: `SELECT DATEDIFF('2018-01-01', '2019-01-01') AS 'Date Difference';`. The result grid below shows a single row with the value -365.

Date Difference
-365



Miles, M. (2021). *Learning MySQL By Example*. EdTech Books. https://edtechbooks.org/learning_mysql