DATE & TIME FUNCTION

Date & Time

MySQL comes with the following data types for storing a date or a date/time value in the database:

- `DATE` - format YYYY-MM-DD
- `DATETIME` - format: YYYY-MM-DD HH:MI:SS
- `TIMESTAMP` - format: YYYY-MM-DD HH:MI:SS
- `YEAR` - format YYYY or YY

SQL Server comes with the following data types for storing a date or a date/time value in the database:

- `DATE` - format YYYY-MM-DD
- `DATETIME` - format: YYYY-MM-DD HH:MI:SS
- `SMALLDATETIME` - format: YYYY-MM-DD HH:MI:SS
- `TIMESTAMP` - format: a unique number

Note: The date types are chosen for a column when you create a new table in your database!

This function only work with `date` data type, otherwise it will not work.

To convert string/numerical value into date value use `to_date()` function.

1. Current Date

```
SELECT CURRENT_DATE;

-- OUTPUT
-- 2023-06-16
```

2. Current Time

```
SELECT CURRENT_TIMESTAMP;

-- OUTPUT
-- 2023-06-16 11:32:17.890 -0700
-- This time zone is of the server where it is located to get indian time add +5:30
-- YYYY-MM-DD HH:MM:SS.miliseconds

SELECT CURRENT_TIME;

-- OUTPUT
-- Same its not indian time zone.
-- 11:34:19
```

3. Get `Year` from DATE — CURRENT DATE IS COLUMN NAME, SO PROVIDE ONLY COLUMN NAME WHICH HAVE `DATE-TYPE` FUNCTION ONLY.

```
SELECT DATE_TRUNC('YEAR', CURRENT_DATE) AS YR_FROM_DATE;

-- OUTPUT
-- 2023-01-01

-- Format of Output in Date_trunc is YYYY-MM-DD
-- Whenever you get year from the data your month and date
-- Start from YYYY-01-01 because it will take all the current year-date
-- from the data because i have provide the `current_date` with it.
```

4. Get `Month` from DATE

```
SELECT DATE_TRUNC('MONTH', CURRENT_DATE) AS MTH_FROM_DATE;

-- OUTPUT
-- 2023-06-01

-- In this you will get same year and month but day will start
-- from 01.
```

5. Get `day` From DATE

```
SELECT DATE_TRUNC('DAY', CURRENT_DATE) AS DAY_FROM_DATE;

-- OUTPUT
```

```
-- 2023-06-17

-- THis will give you current Day from current Date.
```

6. Get `Year` from particular DATE - USE OF `TO_DATE` FUNCTION.

```
SELECT DATE_TRUNC('YEAR', TO_DATE('2022-11-25')) AS YR_FROM_DATE;

-- OUTPUT
-- 2022-01-01
```

7. Get particular `week-date` from DATE

```
SELECT DATE_TRUNC('YEAR', TO_DATE('2023-06-17)) AS WEEK_DATE_FROM_DATE;

-- OUTPUT
-- 2023-06-12

-- Here my output is 12 where i have given it 17, because week
-- start from monday and monday date was 12
```

Example:

```
SELECT day(current_timestamp),
    hour(current_timestamp),
    minute(current_timestamp),
    second(current_timestamp),
    month(current_timestamp);
```

OUTPUT:

	DAY(CURRENT_TIMESTAMP)	HOUR(CURRENT_TIMESTAMP)	···	MINUTE(CURRENT_TIMESTAMP)	SECOND(CURRENT_TIMESTAMP)	MONTH(CURRENT_TIMESTAMP)
1	17	10		42	44	6

8. Get `total no. of weeks` from DATE

```
SELECT WEEK(CURRENT_DATE) AS WEEK_FROM_START_OF_THE_YEAR;

-- OUTPUT
-- 24
-- FOR 2023 TOTAL NO. WEEK PASSED NOW IS 24.
-- TODAY DATE IS 2023-06-17
```

9. Get `total no. of Months` from DATE

```
SELECT MONTH(CURRENT_DATE) AS WEEK_FROM_START_OF_THE_YEAR;

-- OUTPUT
-- 6
```

10. Get `today date` from DATE

```
SELECT DAY(CURRENT_DATE) AS WEEK_FROM_START_OF_THE_YEAR;

-- OUTPUT
-- 17
-- TODAY DATE IS 2023-06-17
```

11. Get `5 days back date` from DATE

```
SELECT DAY(CURRENT_DATE - INTERVAL'5 DAY') AS FIVE_DAYS_BACK_DATE;

-- OUTPUT
-- 12
-- TODAY IS 17

SELECT DAY(CURRENT_DATE - INTERVAL'3 WEEK') AS FIVE_DAYS_BACK_DATE;

-- OUTPUT
-- 27
-- 3 WEEKS BACK FROM DAY 17.
```

12. Get `Last day` of current month

```
SELECT LAST_DAY(DATE('2023-06-17')) AS LAST_DAY_OF_CURRENT_MONTH;

-- OUTPUT
-- 2023-06-30
```

13. Get `last day` of previous month.

```
SELECT LAST_DAY(CURRENT_DATE - INTERVAL '1 MONTH') AS LAST_DAY_OF_PREV_MONTH;

-- OUTPUT
-- 2023-05-31

SELECT LAST_DAY(CURRENT_DATE - INTERVAL '3 MONTH') AS LAST_DAY_OF_PAST_3_MONTH;

-- OUTPUT
-- 2023-03-31
```

14. Get `quater` from DATE - like (JAN TO MARCH - 1 QUATER, 12 MONTH DIVIDE BY 3 YOU WILL GET YOUR 4 QUATER.)

```
SELECT QUATER(CURRENT_DATE) AS QTR;

-- OUTPUT
-- 2
-- JUNE COME IN 2 QUATER.
```

`TO_DATE ( String, [Format], [NLS Setting] )`

Format	Description
AD A.D.	AD indicator to use in conjunction with the year
AM A.M. PM P.M.	Meridian indicator
BC B.C.	BC indicator to use in conjunction with the year
D	Day of week (1-7)
DAY	Name of day
DD	Day of month (1-31)
DDD	Day of year (1-366)
DY	Abbreviated name of day

HH	Hour of day (1-12)
HH24	Hour of day (0-23)
MI	Minutes (0-59)
MM	Month (01-12)
MON	Abbreviated name of month
MONTH	Name of month
RM	Month in Roman Numerals (I - XII)
RR	Accepts a 2-digit input, and returns a 4-digit year. A value between '00' and '49' returns the year in the same century. A value between '50' and '99' returns a year in the previous century.
RRRR	Accepts a 2-digit input or a 4-digit input, and returns a 4-digit year. For 4-digit input, the same value is returned. For 2 digit input, a value between '00' and '49' returns the year in the same century, and a value between '50' and '99' returns a year in the previous century.
SS	Second (0-59)
SSSSS	Seconds past midnight (0-86399)
Y	Accepts a 1-digit input, and returns a 4-digit year in that decade.
YY	Accepts a 2-digit input, and returns a 4-digit year in that century.
YYY	Accepts a 3-digit input, and returns a 4-digit year in that millennium.
YYYY SYYYY	Accepts a 4-digit input, and returns a 4-digits year.

15. Changing to Date

```
SELECT to_date('23-08-2022', 'YYYY-MM-DD');

-- OUTPUT
-- 2022-08-23
-- Output will always come in this format C.
```

16. Changing the format of DATE

```
SELECT to_char(to_date('2022-08-23'), 'dd-mm-yy');

-- OUTPUT
-- 23-08-22
-- When want in this type of format, always use yyyy-mm-dd format
-- query then you can get in your type of format.

SELECT to_char(to_date('2022-08-23'), 'mm-yy');

-- OUTPUT
-- 08-22
-- If client only want month and year only in the data this will
-- be very helpful.
```

17. IF client want `month name` and year - `MON` will give month name.

```
SELECT to_char(to_date('2022-08-23'), 'mon-yy');

-- OUTPUT
-- Aug-22
```

18. IF client want `DAY name` - `DY` will give day name.

```
SELECT to_char(to_date('2022-08-23'), 'dy');

-- OUTPUT
-- Tue

SELECT DAYNAME(CURRENT_DATE);
```

19. Date difference

```
SELECT DATEDIFF('MONTH', '2023-03-17','2023-06-17');

-- OUTPUT
-- 3
-- 3 MONTH DIFFERENCE IS THERE
```

- DATEDIFF FROM 2 COLUMNS

```
SELECT DATEDIFF('DAY', COLUMN1, COLUMN2) AS column_name FROM TABLE_NAME;
```