

# MySQL Date and Time Functions

Here are the commonly used functions for date and time operations:

Function	Description
<b>NOW()</b>	Returns the current date and time.
<b>CURDATE()</b>	Returns the current date.
<b>CURTIME()</b>	Returns the current time.
<b>DATE(expr)</b>	Extracts the date part from a datetime expression.
<b>TIME(expr)</b>	Extracts the time part from a datetime expression.
<b>YEAR(date)</b>	Returns the year from a date.
<b>MONTH(date)</b>	Returns the month from a date.
<b>DAY(date)</b>	Returns the day of the month from a date.
<b>DATEDIFF(date1, date2)</b>	Returns the difference in days between two dates.
<b>DATE_ADD(date, INTERVAL expr)</b>	Adds a time interval to a date.
<b>DATE_SUB(date, INTERVAL expr)</b>	Subtracts a time interval from a date.
<b>DATE_FORMAT(date, format)</b>	Formats a date based on a specified format string.
<b>UNIX_TIMESTAMP(date)</b>	Converts a DATE or DATETIME to a UNIX timestamp (seconds since '1970-01-01 00:00:00').
<b>FROM_UNIXTIME(timestamp)</b>	Converts a UNIX timestamp to a DATETIME.
<b>LAST_DAY(date)</b>	Returns the last day of the month for a given date.
<b>WEEK(date)</b>	Returns the week number for a given date.
<b>DAYOFWEEK(date)</b>	Returns the weekday index (1 = Sunday, 7 = Saturday).
<b>STR_TO_DATE(str, format)</b>	Parses a string into a date using the specified format.
<b>ADDTIME(time, expr)</b>	Adds a time interval to a time or datetime value.
<b>SUBTIME(time, expr)</b>	Subtracts a time interval from a time or datetime value.

# Scenario-Based Operations on Date and Time in MySQL

## 1. Get Current Date and Time

SELECT NOW(); -- Returns current date and time

SELECT CURDATE(); -- Returns current date

SELECT CURTIME(); -- Returns current time

---

## 2. Calculate the Difference Between Two Dates

- Example: Calculate the number of days between two dates.

SELECT DATEDIFF('2025-01-17', '2024-12-25') AS Days\_Difference;

---

## 3. Filter Data Based on Date

- Example: Fetch records created in the last 7 days.

SELECT \*

FROM orders

WHERE order\_date >= DATE\_SUB(CURDATE(), INTERVAL 7 DAY);

---

## 4. Add or Subtract Date Intervals

- Add 10 days to a specific date.

SELECT DATE\_ADD('2025-01-17', INTERVAL 10 DAY) AS New\_Date;

- Subtract 3 months from the current date.

SELECT DATE\_SUB(NOW(), INTERVAL 3 MONTH) AS Modified\_Date;

---

## 5. Extract Components from a Date

- Get year, month, and day from a date.

SELECT YEAR(order\_date) AS Year,

MONTH(order\_date) AS Month,

DAY(order\_date) AS Day

FROM orders;

---

## 6. Format a Date

- Convert a date to a specific string format.

```
SELECT DATE_FORMAT(order_date, '%Y-%m-%d') AS Formatted_Date  
FROM orders;
```

---

## 7. Find the Last Day of a Month

- Example: Get the last day of January 2025.

```
SELECT LAST_DAY('2025-01-01') AS Last_Day;
```

---

## 8. Convert Between Date and Timestamp

- Convert a date to a UNIX timestamp.

```
SELECT UNIX_TIMESTAMP('2025-01-17 14:30:00') AS Timestamp;
```

- Convert a UNIX timestamp back to a DATETIME.

```
SELECT FROM_UNIXTIME(1737102600) AS DateTime;
```

---

## 9. Work with Time Zones

- Display the current UTC date and time.

```
SELECT UTC_DATE(), UTC_TIME(), UTC_TIMESTAMP();
```

---

## 10. Find Weekdays

- Example: Get the weekday name of a specific date.

```
SELECT DAYNAME('2025-01-17') AS Weekday;
```

- Get the weekday index.

```
SELECT DAYOFWEEK('2025-01-17') AS Weekday_Index; -- 1 = Sunday
```

---

### 11. Find Records Matching Specific Time Intervals

- Example: Fetch records created between 9 AM and 5 PM.

```
SELECT *  
FROM attendance  
WHERE TIME(entry_time) BETWEEN '09:00:00' AND '17:00:00';
```

---

### 12. Check if a Date is a Weekend

```
SELECT CASE WHEN DAYOFWEEK(order_date) IN (1, 7) THEN 'Weekend' ELSE  
'Weekday' END AS Day_Type  
FROM orders;
```

---

### 13. Calculate Age from a Birthdate

```
SELECT YEAR(CURDATE()) - YEAR(birth_date) -  
      (DATE_FORMAT(CURDATE(), '%m%d') < DATE_FORMAT(birth_date, '%m%d'))  
AS Age  
FROM employees;
```

---

### 14. Group Data by Date/Month/Year

- Example: Group orders by year and calculate total sales.

```
SELECT YEAR(order_date) AS Order_Year,  
      SUM(total_amount) AS Total_Sales  
FROM orders  
GROUP BY YEAR(order_date);
```

---

### 15. Generate a Sequence of Dates

- Example: Create a list of dates between two given dates.

```
WITH RECURSIVE date_sequence AS (  
    SELECT '2025-01-01' AS generated_date  
    UNION ALL  
    SELECT DATE_ADD(generated_date, INTERVAL 1 DAY)  
    FROM date_sequence  
    WHERE generated_date < '2025-01-10'  
)  
SELECT * FROM date_sequence;
```