

MySQL Functions — Complete Guide

1. Aggregate Functions

Used for summarizing data (often with GROUP BY).

Function	Description	Example
COUNT()	Counts rows	SELECT COUNT(*) FROM employees;
SUM()	Adds up values	SELECT SUM(salary) FROM employees;
AVG()	Finds average	SELECT AVG(salary) FROM employees;
MIN()	Finds smallest value	SELECT MIN(age) FROM students;
MAX()	Finds largest value	SELECT MAX(age) FROM students;
GROUP_CONCAT()	Joins strings from grouped rows	SELECT dept, GROUP_CONCAT(name) FROM employees GROUP BY dept;

2. String Functions

Used to manipulate and format text.

Function	Description	Example
CONCAT()	Joins strings	SELECT CONCAT(first_name, ' ', last_name);
LENGTH()	Returns string length	SELECT LENGTH('MySQL');
LOWER() / UPPER()	Changes case	SELECT LOWER(name);
LEFT(str, n) / RIGHT(str, n)	Extracts part of string	SELECT LEFT('Developer', 3); -- Dev
SUBSTRING(str, start, len)	Gets substring	SELECT SUBSTRING('DataScience', 5, 7);
REPLACE(str, from, to)	Replaces substring	SELECT REPLACE('AI rocks', 'rocks', 'rules');
TRIM() / LTRIM() / RTRIM()	Removes spaces	SELECT TRIM(' space ');
LOCATE(substr, str)	Finds substring position	SELECT LOCATE('a', 'Data');
REVERSE(str)	Reverses string	SELECT REVERSE('MySQL');

3. Numeric Functions

Function	Description	Example
ROUND(num, decimals)	Rounds value	SELECT ROUND(12.3456, 2);
CEIL() / FLOOR()	Round up/down	SELECT CEIL(3.2), FLOOR(3.8);
ABS()	Absolute value	SELECT ABS(-50);
MOD(a, b)	Remainder	SELECT MOD(10, 3);
POWER(a, b)	a raised to b	SELECT POWER(2, 3);
RAND()	Random number (0–1)	SELECT RAND();
TRUNCATE(num, decimals)	Trims decimals without rounding	SELECT TRUNCATE(12.987, 2);

4. Date & Time Functions

Function	Description	Example
NOW() / CURRENT_TIMESTAMP()	Current date & time	SELECT NOW();
CURDATE() / CURTIME()	Current date/time separately	SELECT CURDATE(), CURTIME();
DATE()	Extracts date from datetime	SELECT DATE(NOW());
YEAR() / MONTH() / DAY()	Extract parts	SELECT YEAR(NOW()), MONTH(NOW());
DATE_ADD(date, INTERVAL n unit)	Adds interval	SELECT DATE_ADD(NOW(), INTERVAL 7 DAY);
DATE_SUB(date, INTERVAL n unit)	Subtracts interval	SELECT DATE_SUB(NOW(), INTERVAL 1 MONTH);
DATEDIFF(date1, date2)	Difference in days	SELECT DATEDIFF('2025-10-10','2025-10- 01');
TIMESTAMPDIFF(unit, date1, date2)	Flexible difference	SELECT TIMESTAMPDIFF(HOUR, start_time, end_time);
DAYNAME() / MONTHNAME()	Name of day/month	SELECT DAYNAME('2025-10-10');
STR_TO_DATE(str, format)	Converts string to date	SELECT STR_TO_DATE('10-10-2025', '%d- %m-%Y');
DATE_FORMAT(date, format)	Formats date output	SELECT DATE_FORMAT(NOW(), '%d-%b- %Y');

5. Conditional & Logical Functions

Function	Description	Example
IF(condition, true_val, false_val)	Simple condition	SELECT IF(salary > 50000, 'High', 'Low');
IFNULL(expr, alt)	Replace NULL	SELECT IFNULL(phone, 'Not Provided');
NULLIF(a, b)	Returns NULL if equal	SELECT NULLIF(10, 10);
CASE WHEN	Conditional logic	SELECT CASE WHEN marks >= 90 THEN 'A' WHEN marks >= 75 THEN 'B' ELSE 'C' END;

6. Mathematical Functions

Function	Description	Example
SQRT()	Square root	SELECT SQRT(25);
EXP()	Exponential	SELECT EXP(2);
LOG() / LOG10()	Logarithm	SELECT LOG(100);
PI()	Value of π	SELECT PI();
SIN(), COS(), TAN()	Trig functions	SELECT SIN(PI()/2);

7. JSON Functions (Modern MySQL)

Function	Description	Example
JSON_OBJECT()	Create JSON	SELECT JSON_OBJECT('name','John',age,30);
JSON_EXTRACT()	Get value from JSON	SELECT JSON_EXTRACT('{ "a":10}', '\$.a');
JSON_ARRAY()	Create array	SELECT JSON_ARRAY('AI','ML','DL');
JSON_CONTAINS()	Checks if value exists	SELECT JSON_CONTAINS('["AI","ML"]', '"AI"');

8. Information & System Functions

Function	Description	Example
VERSION()	MySQL version	SELECT VERSION();
DATABASE()	Current DB name	SELECT DATABASE();
USER()	Current user	SELECT USER();
LAST_INSERT_ID()	Last auto ID	SELECT LAST_INSERT_ID();

9. Window (Analytic) Functions (MySQL 8+)

Used for advanced analytics — no need to GROUP BY.

Function	Description	Example
ROW_NUMBER()	Row sequence	SELECT name, ROW_NUMBER() OVER(ORDER BY salary DESC);
RANK() / DENSE_RANK()	Ranking	SELECT RANK() OVER(ORDER BY score DESC);
LEAD() / LAG()	Previous/next row value	SELECT LAG(salary) OVER(ORDER BY id);
NTILE(n)	Divide rows into buckets	SELECT NTILE(4) OVER(ORDER BY marks DESC);
SUM() OVER()	Running total	SELECT name, SUM(sales) OVER(ORDER BY date);