MySQL provides a wide range of built-in functions to handle various tasks, including string manipulation, mathematical calculations, date and time operations, and data formatting. Below is a comprehensive overview of some commonly used MySQL functions, categorized by their functionality:

**1. String Functions**

* **CONCAT(str1, str2, ...)**
  + **Description:** Concatenates two or more strings.
  + **Example:**

SELECT CONCAT('Hello', ' ', 'World');

-- Output: Hello World

* **SUBSTRING(str, start, length)**
  + **Description:** Extracts a substring from a string.
  + **Example:**

SELECT SUBSTRING('Hello World', 7, 5);

-- Output: World

* **LENGTH(str)**
  + **Description:** Returns the length of a string.
  + **Example:**

SELECT LENGTH('Hello World');

-- Output: 11

* **UPPER(str)**
  + **Description:** Converts a string to uppercase.
  + **Example:**

SELECT UPPER('Hello World');

-- Output: HELLO WORLD

* **LOWER(str)**
  + **Description:** Converts a string to lowercase.
  + **Example:**

SELECT LOWER('Hello World');

-- Output: hello world

* **REPLACE(str, search, replace)**
  + **Description:** Replaces occurrences of a substring with another substring.
  + **Example:**

SELECT REPLACE('Hello World', 'World', 'MySQL');

-- Output: Hello MySQL

* **TRIM(str)**
  + **Description:** Removes leading and trailing spaces from a string.
  + **Example:**

SELECT TRIM(' Hello World ');

-- Output: Hello World

**2. Mathematical Functions**

* **ABS(x)**
  + **Description:** Returns the absolute value of a number.
  + **Example:**

SELECT ABS(-10);

-- Output: 10

* **ROUND(x, d)**
  + **Description:** Rounds a number to d decimal places.
  + **Example:**

SELECT ROUND(123.4567, 2);

-- Output: 123.46

* **FLOOR(x)**
  + **Description:** Rounds a number down to the nearest integer.
  + **Example:**

SELECT FLOOR(12.99);

-- Output: 12

* **CEIL(x)**
  + **Description:** Rounds a number up to the nearest integer.
  + **Example:**

SELECT CEIL(12.01);

-- Output: 13

* **RAND()**
  + **Description:** Returns a random floating-point number between 0 and 1.
  + **Example:**

SELECT RAND();

-- Output: 0.123456 (value will vary)

**3. Date and Time Functions**

* **NOW()**
  + **Description:** Returns the current date and time.
  + **Example:**

SELECT NOW();

-- Output: 2024-08-27 14:30:00

* **CURDATE()**
  + **Description:** Returns the current date.
  + **Example:**

SELECT CURDATE();

-- Output: 2024-08-27

* **DATE\_ADD(date, INTERVAL expr unit)**
  + **Description:** Adds a time interval to a date.
  + **Example:**

SELECT DATE\_ADD('2024-08-27', INTERVAL 10 DAY);

-- Output: 2024-09-06

* **DATE\_SUB(date, INTERVAL expr unit)**
  + **Description:** Subtracts a time interval from a date.
  + **Example:**

SELECT DATE\_SUB('2024-08-27', INTERVAL 10 DAY);

-- Output: 2024-08-17

* **DATEDIFF(date1, date2)**
  + **Description:** Returns the number of days between two dates.
  + **Example:**

SELECT DATEDIFF('2024-08-27', '2024-08-17');

-- Output: 10

* **YEAR(date)**
  + **Description:** Returns the year from a date.
  + **Example:**

SELECT YEAR('2024-08-27');

-- Output: 2024

* **MONTH(date)**
  + **Description:** Returns the month from a date.
  + **Example:**

SELECT MONTH('2024-08-27');

-- Output: 8

* **DAY(date)**
  + **Description:** Returns the day of the month from a date.
  + **Example:**

SELECT DAY('2024-08-27');

-- Output: 27

**4. Aggregation Functions**

* **COUNT(expression)**
  + **Description:** Returns the number of rows that match a specified condition.
  + **Example:**

SELECT COUNT(\*) FROM employees;

-- Output: (number of employees)

* **SUM(expression)**
  + **Description:** Returns the sum of values in a column.
  + **Example:**

SELECT SUM(salary) FROM employees;

-- Output: (total salary)

* **AVG(expression)**
  + **Description:** Returns the average value of a column.
  + **Example:**

SELECT AVG(salary) FROM employees;

-- Output: (average salary)

* **MAX(expression)**
  + **Description:** Returns the maximum value in a column.
  + **Example:**

SELECT MAX(salary) FROM employees;

-- Output: (highest salary)

* **MIN(expression)**
  + **Description:** Returns the minimum value in a column.
  + **Example:**

SELECT MIN(salary) FROM employees;

-- Output: (lowest salary)

**5. Control Flow Functions**

* **IF(condition, true\_value, false\_value)**
  + **Description:** Returns true\_value if condition is true, otherwise returns false\_value.
  + **Example:**

SELECT IF(salary > 50000, 'High', 'Low') AS salary\_level FROM employees;

* **CASE**
  + **Description:** Provides conditional logic to return different values based on conditions.
  + **Example:**

SELECT

CASE

WHEN salary > 50000 THEN 'High'

WHEN salary BETWEEN 30000 AND 50000 THEN 'Medium'

ELSE 'Low'

END AS salary\_level

FROM employees;

These functions can help you perform a wide range of operations and data manipulations in MySQL. Understanding and utilizing these functions effectively will enhance your ability to work with MySQL databases and write more efficient queries.