Numeric functions in SQL are a set of built-in functions designed to perform mathematical and numerical operations on numeric data types, such as integers, floats, or decimals. These functions are useful for calculations, transformations, and statistical analysis within a database query. Below are common categories and examples of numeric functions:

**1. Basic Arithmetic Functions**

* **ABS(number)**: Returns the absolute value of a number.
  + Example: SELECT ABS(-10); → Result: 10
* **MOD(dividend, divisor)**: Returns the remainder of the division.
  + Example: SELECT MOD(10, 3); → Result: 1
* **CEIL(number) / CEILING(number)**: Rounds a number up to the nearest integer.
  + Example: SELECT CEIL(4.2); → Result: 5
* **FLOOR(number)**: Rounds a number down to the nearest integer.
  + Example: SELECT FLOOR(4.8); → Result: 4
* **ROUND(number, decimals)**: Rounds a number to the specified number of decimal places.
  + Example: SELECT ROUND(3.14159, 2); → Result: 3.14
* **SIGN(number)**: Returns the sign of a number (-1 for negative, 0 for zero, 1 for positive).
  + Example: SELECT SIGN(-15); → Result: -1

**2. Trigonometric Functions**

* **SIN(number)**, **COS(number)**, **TAN(number)**: Compute the sine, cosine, and tangent of a number (angle in radians).
  + Example: SELECT SIN(PI() / 2); → Result: 1
* **ASIN(number)**, **ACOS(number)**, **ATAN(number)**: Compute the arc sine, arc cosine, and arc tangent.
  + Example: SELECT ACOS(1); → Result: 0
* **ATAN2(y, x)**: Returns the arc tangent of the two arguments, often used for calculating angles in coordinate geometry.
  + Example: SELECT ATAN2(1, 1); → Result: 0.785398

**3. Exponential and Logarithmic Functions**

* **POWER(base, exponent)**: Raises the base to the power of the exponent.
  + Example: SELECT POWER(2, 3); → Result: 8
* **EXP(number)**: Returns *e* raised to the power of the given number.
  + Example: SELECT EXP(1); → Result: 2.718281
* **LOG(number)** or **LN(number)**: Returns the natural logarithm (base *e*) of a number.
  + Example: SELECT LOG(2.718281); → Result: 1
* **LOG10(number)**: Returns the logarithm of a number in base 10.
  + Example: SELECT LOG10(1000); → Result: 3
* **SQRT(number)**: Returns the square root of a number.
  + Example: SELECT SQRT(16); → Result: 4

**4. Random and Special Functions**

* **RAND(seed)**: Returns a random number between 0 and 1. The optional seed ensures reproducibility.
  + Example: SELECT RAND(); → Result: 0.738920
* **PI()**: Returns the value of π (Pi).
  + Example: SELECT PI(); → Result: 3.141593

**5. Statistical and Aggregate Functions**

Some functions, though technically aggregates, deal with numeric data:

* **AVG(column)**: Calculates the average of a column's values.
  + Example: SELECT AVG(salary) FROM employees;
* **SUM(column)**: Calculates the total sum of a column's values.
  + Example: SELECT SUM(sales) FROM transactions;
* **COUNT(column)**: Counts the number of non-NULL rows in a column.
  + Example: SELECT COUNT(\*) FROM orders;

**Usage in SQL Queries**

Numeric functions are often used in SELECT, WHERE, GROUP BY, and ORDER BY clauses to perform calculations or filter data based on numeric conditions.

**Example Query:**

SELECT product\_id,

ROUND(price \* 1.1, 2) AS increased\_price,

CEIL(price) AS rounded\_up\_price

FROM products

WHERE ABS(discount) > 5

ORDER BY increased\_price DESC;

This query calculates a price increase, rounds it, and filters products with significant discounts.