

# Mathematical Functions and Operators

# Aggregate arithmetic

## Recall:

COUNT, SUM, AVG, MIN, MAX

Each requires multiple rows as input

```
SELECT ABS(-17.4); -- 17.4
SELECT CEIL(42.2); -- 43
SELECT FACTORIAL(5); -- 120
```

Function Name	Example	SQL	Python
Absolute Value	<code>abs(-17.4) → 17.4</code>	<code>ABS(x)</code>	<code>abs(x)</code>
Ceiling	<code>ceil(42.2) → 43</code>	<code>CEIL(x)</code>	<code>math.ceil(x)</code>
Factorial	<code>factorial(5) → 120</code>	<code>FACTORIAL(x)</code>	<code>math.factorial(x)</code>
Floor	<code>floor(42.8) → 42</code>	<code>FLOOR(x)</code>	<code>math.floor(x)</code>
Greatest Common Divisor	<code>gcd(1071, 462) → 21</code>	<code>GCD(x, y)</code>	<code>math.gcd(x, y)</code>
Least Common Multiple	<code>lcm(1071, 462) → 23562</code>	<code>LCM(x, y)</code>	<code>math.lcm(x, y)</code>
Natural Log	<code>ln(2.0) → 0.693147</code>	<code>LN(x)</code>	<code>math.log(x)</code>
Log of <b>x</b> to Base <b>b</b>	<code>log(2.0, 64.0) → 6</code>	<code>LOG(b, x)</code>	<code>math.log(x, b)</code>
Modulo	<code>mod(9,4) → 1</code>	<code>MOD(x,y)</code>	<code>math.remainder(x,y)</code>
Power	<code>power(9, 3) → 729</code>	<code>POWER(x, y)</code>	<code>pow(x,y)</code>
Round <b>x</b> to <b>y</b> Decimal Places	<code>round(42.4382, 2) → 42.44</code>	<code>ROUND(x, y)</code>	<code>round(x, y)</code>
Square Root	<code>sqrt(2) → 1.41421</code>	<code>SQRT(x)</code>	<code>math.sqrt(x)</code>
Truncate	<code>trunc(42.8) → 42</code>	<code>TRUNC(x)</code>	<code>math.trunc(x)</code>

## Pure Functions:

Same input always returns same output  
No side effects (no external impact)

Example of **impure** function: NOW()

May return different values, given same input

Function Name	Example	SQL	Python
Absolute Value	<code>abs(-17.4) → 17.4</code>	<code>ABS(x)</code>	<code>abs(x)</code>
Ceiling	<code>ceil(42.2) → 43</code>	<code>CEIL(x)</code>	<code>math.ceil(x)</code>
Factorial	<code>factorial(5) → 120</code>	<code>FACTORIAL(x)</code>	<code>math.factorial(x)</code>
Floor	<code>floor(42.8) → 42</code>	<code>FLOOR(x)</code>	<code>math.floor(x)</code>
Greatest Common Divisor	<code>gcd(1071, 462) → 21</code>	<code>GCD(x, y)</code>	<code>math.gcd(x, y)</code>
Least Common Multiple	<code>lcm(1071, 462) → 23562</code>	<code>LCM(x, y)</code>	<code>math.lcm(x, y)</code>
Natural Log	<code>ln(2.0) → 0.693147</code>	<code>LN(x)</code>	<code>math.log(x)</code>
Log of <b>x</b> to Base <b>b</b>	<code>log(2.0, 64.0) → 6</code>	<code>LOG(b, x)</code>	<code>math.log(x, b)</code>
Modulo	<code>mod(9,4) → 1</code>	<code>MOD(x,y)</code>	<code>math.remainder(x,y)</code>
Power	<code>power(9, 3) → 729</code>	<code>POWER(x, y)</code>	<code>pow(x,y)</code>
Round <b>x</b> to <b>y</b> Decimal Places	<code>round(42.4382, 2) → 42.44</code>	<code>ROUND(x, y)</code>	<code>round(x, y)</code>
Square Root	<code>sqrt(2) → 1.41421</code>	<code>SQRT(x)</code>	<code>math.sqrt(x)</code>
Truncate	<code>trunc(42.8) → 42</code>	<code>TRUNC(x)</code>	<code>math.trunc(x)</code>

```
SELECT ABS(-17.4); -- 17.4
SELECT CEIL(42.2); -- 43
SELECT FACTORIAL(5); -- 120
```

# Mathematical operators

Operation	Example	SQL	Python
Addition	$2 + 3 \rightarrow 5$	$x + y$	$x + y$
Subtraction	$2 - 3 \rightarrow -1$	$x - y$	$x - y$
Negation	$-(-4) \rightarrow 4$	$-x$	$-x$
Multiplication	$2 * 3 \rightarrow 6$	$x * y$	$x * y$
Division	$5.0 / 2 \rightarrow 2.50$	$x / y$	$x / y$
Modulo	$5 \% 4 \rightarrow 1$	$x \% y$	$x \% y$
Power	$2 ^ 3 \rightarrow 8$	$x^y$	$x ** y$

```
SELECT 2 + 3; -- 5
```

```
SELECT 2 - 3; -- -1
```

```
SELECT -(-4); -- 4
```