



# Module 27: Mathematical Functions



## Why Use Math Functions?

Mathematical functions in SQL help perform calculations like **rounding**, **generating random numbers**, and **power/exponents** — useful in reporting, simulations, discount logic, and more.

1

**CEILING()** and **FLOOR()**

### ✓ Definitions:

- **CEILING()** returns the **next highest integer** (rounds up).
- **FLOOR()** returns the **next lowest integer** (rounds down).

### ✓ Syntax:

`CEILING(numeric_expression)`

`FLOOR(numeric_expression)`

### 📌 Example:

```
SELECT CEILING(45.23) AS CeilValue, FLOOR(45.23) AS FloorValue;
```

-- Output: CeilValue = 46, FloorValue = 45

The screenshot shows a SQL query window with the following details:

- Query text: `SELECT CEILING(45.23) AS CeilValue, FLOOR(45.23) AS FloorValue;`
- Execution results:

	CeilValue	FloorValue
1	46	45
- UI elements: A vertical toolbar on the left, a status bar showing "100 %", and tabs for "Results" and "Messages".

2

**RAND()** – Random Number Generator

### ✓ Definition:

Returns a **random float value** between **0 (inclusive)** and **1 (exclusive)**.

### ✓ Syntax:

`RAND()` -- no arguments = random decimal between 0 and 1

## 📌 Example:

```
SELECT RAND();
```

The screenshot shows a SQL query window with the following content:

```
SELECT RAND();
```

Execution results:

	(No column name)
1	0.666053418994843

## ✓ Custom Range Example:

-- Random value between 4 and 10 (float)

```
SELECT RAND() * (10 - 4) + 4 AS rand_;
```

-- Random integer between 4 and 10

```
SELECT FLOOR(RAND() * (10 - 4 + 1)) + 4 AS floor_rand;
```

## Explanation:

- Multiply by range size → `RAND() * (high - low)`
- Add the lower bound → `+ low`
- Use `FLOOR()` to convert to an integer (optional)

The screenshot shows a SQL query window with the following content:

```
-- Random value between 4 and 10 (float)
SELECT RAND() * (10 - 4) + 4 AS rand_;

-- Random integer between 4 and 10
SELECT FLOOR(RAND() * (10 - 4 + 1)) + 4 AS floor_rand;
```

Execution results:

	rand_
1	9.40162843753211

  

	floor_rand
1	8

**3**

## ROUND()

### ✓ Definition:

Rounds a number to a specific number of decimal places.

### ✓ Syntax:

ROUND(numeric\_expression, decimal\_places)

### 📌 Example:

```
SELECT ROUND(123.45678, 2) AS RoundedVal;
```

The screenshot shows a SQL query window with the following content:

```
SELECT ROUND(123.45678, 2) AS RoundedVal;
```

The results pane displays a single row with the following data:

	RoundedVal
1	123.46000

**4**

## POWER()

### ✓ Definition:

Returns the **result of a number raised to a power** (exponentiation).

### ✓ Syntax:

POWER(base, exponent)

### 📌 Example:

```
SELECT POWER(5, 3) AS Result;
```

The screenshot shows a SQL query window with the following content:

```
SELECT POWER(5, 3) AS Result;
```

The results pane displays a single row with the following data:

	Result
1	125



## Key Points to Remember

Function	Use Case
<code>CEILING()</code>	Always rounds up
<code>FLOOR()</code>	Always rounds down
<code>RAND()</code>	Returns a <b>float</b> between 0 and 1
<code>ROUND()</code>	Useful for formatting decimal output
<code>POWER()</code>	Calculate exponents (e.g., square, cube)
<code>FLOOR(RAND()*N)+M</code>	Formula for random <b>integers in range</b>

- ✓ `CEILING`, `FLOOR`, and `ROUND` are good for **billing, formatting, and data cleaning**
- ✓ `RAND()` helps in generating **test data** or random IDs
- ✓ `POWER()` can be used for **growth calculations, interest, etc.**