



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 editor with the following query:

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

Below the query editor, there is a tab labeled "Results" which is active. It displays a table with two columns: "CeilValue" and "FloorValue". The first row of data shows the results of the query: 46 for CeilValue and 45 for FloorValue.

	CeilValue	FloorValue
1	46	45

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 editor with the query `SELECT RAND();` entered. Below the editor, the 'Results' tab is active, displaying a single row with the value `0.666053418994843` under the column header '(No column name)'.

	(No column name)
1	0.666053418994843

Custom Range Example:

-- Random value between 4 and 10 (float)

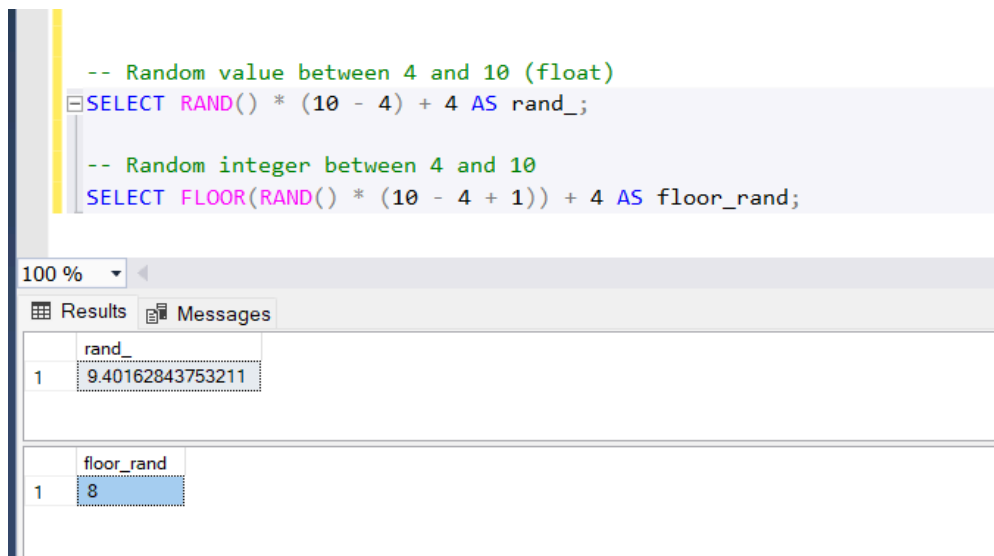
```
SELECT RAND() * (10 - 4) + 4 AS rand_;
```

-- 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 editor with two queries entered. The first query is `SELECT RAND() * (10 - 4) + 4 AS rand_;` and the second is `SELECT FLOOR(RAND() * (10 - 4 + 1)) + 4 AS floor_rand;`. Below the editor, the 'Results' tab is active, displaying two rows of results. The first row has the column header `rand_` and the value `9.40162843753211`. The second row has the column header `floor_rand` and the value `8`.

	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 editor with the query: `SELECT ROUND(123.45678, 2) AS RoundedVal;`. Below the query, there is a toolbar with a dropdown set to '100 %' and two buttons: 'Results' and 'Messages'. The 'Results' button is active, and a table of results is displayed below it.

	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 editor with the query: `SELECT POWER(5, 3) AS Result;`. Below the query, there is a toolbar with a dropdown set to '100 %' and two buttons: 'Results' and 'Messages'. The 'Results' button is active, and a table of results is displayed below it.

	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 float 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 integers in range

- ✓ `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.**