

📄

🔍

✍️

ChatGPT

🔗 Explore GPTs

Today

MSSQL String Character Funct

IN vs ANY Explanation

Full Stack SQL Queries

Azure Certification Guide

Yesterday

SQL Queries and Topics

I am a new comer to mssql server and you are experieced teacher and who teaches from scratch gives comprehensive notes. Now explain me functions in mssql - > 1. string function , char functions

Functions in MSSQL - String and Character Functions

SQL Server provides several built-in functions to manipulate string and character data. These functions are commonly used in queries to format, extract, or modify text values.

1. String Functions in MSSQL

String functions in SQL Server are used to perform operations on string values (i.e., VARCHAR , NVARCHAR , TEXT , etc.).

Function	Description	Example	Output
LEN()	Returns the length of a string (excluding trailing spaces).	SELECT LEN('Hello')	5
LEFT()	Extracts a specified number of characters from the left.	SELECT LEFT('SQL Server', 3)	SQL
RIGHT()	Extracts a specified number of characters from the right.	SELECT RIGHT('Database', 4)	base
SUBSTRING()	Extracts a portion of a string.	SELECT SUBSTRING('Microsoft SQL Server', 11, 3)	SQL
CHARINDEX()	Finds the position of a substring in a string (1-based index).	SELECT CHARINDEX('SQL', 'Microsoft SQL Server')	11
PATINDEX()	Similar to CHARINDEX() but supports wildcards (%).	SELECT PATINDEX('%SQL%', 'Microsoft SQL Server')	11
REPLACE()	Replaces occurrences of a substring with another substring.	SELECT REPLACE('Hello World', 'World', 'SQL')	Hello SQL
STUFF()	Deletes and inserts a substring at a given position.	SELECT STUFF('Hello World', 6, 5, 'SQL')	Hello SQL
LTRIM()	Removes leading spaces from a string.	SELECT LTRIM(' SQL Server')	SQL Server
RTRIM()	Removes trailing spaces from a string.	SELECT RTRIM('SQL Server ')	SQL Server
LOWER()	Converts a string to lowercase.	SELECT LOWER('SQL SERVER')	sql server
UPPER()	Converts a string to uppercase.	SELECT UPPER('sql server')	SQL SERVER
CONCAT()	Joins two or more strings together.	SELECT CONCAT('SQL ', 'Server')	SQL Server
CONCAT_WS()	Concatenates strings with a separator.	SELECT CONCAT_WS('-', '2024', '05', '18')	2024-05-18
FORMAT()	Formats a value based on a format string.	SELECT FORMAT(GETDATE(), 'yyyy-MM-dd')	2025-03-18

2. Character Functions in MSSQL

Character functions deal specifically with characters or character codes.

Function	Description	Example	Output
ASCII()	Returns the ASCII code of the first character.	SELECT ASCII('A')	65

CHAR()	Returns the character for a given ASCII code.	SELECT CHAR(65)	A
UNICODE()	Returns the Unicode code point of the first character.	SELECT UNICODE('3')	2309
NCHAR()	Returns the Unicode character for a given code.	SELECT NCHAR(2309)	3

### Example Query

sqlCopyEdit

```
SELECT
  LEN('Microsoft SQL Server') AS Length,
  LEFT('Database', 4) AS LeftPart,
  RIGHT('Database', 3) AS RightPart,
  SUBSTRING('Database', 3, 4) AS MiddlePart,
  CHARINDEX('SQL', 'Microsoft SQL Server') AS PositionSQL,
  REPLACE('Hello World', 'World', 'SQL') AS ReplacedString,
  UPPER('sql server') AS UppercaseString,
  LOWER('SQL SERVER') AS LowercaseString,
  ASCII('A') AS ASCIIValue,
  CHAR(65) AS CharValue;
```

### Summary

- **String functions** help in extracting, modifying, or formatting string values.
- **Character functions** help in working with ASCII or Unicode character codes.

Would you like a practice set to test your understanding? 🚀

good now tell me for 2. Number/mathematical functions 3. List functions 4. date function

Glad you liked the explanation! Now, let's go step by step through **Number/Mathematical Functions**, **List Functions**, and **Date Functions** in SQL Server.

## 2. Number/Mathematical Functions in MSSQL

These functions perform mathematical operations like rounding, truncation, logarithms, and more.

Function	Description	Example	Output
ABS()	Returns the absolute value of a number.	SELECT ABS(-10)	10
CEILING()	Rounds a number up to the next integer.	SELECT CEILING(4.2)	5
FLOOR()	Rounds a number down to the previous integer.	SELECT FLOOR(4.8)	4
ROUND()	Rounds a number to a specified number of decimal places.	SELECT ROUND(12.345, 2)	12.35
POWER()	Returns the result of a number raised to a power.	SELECT POWER(2, 3)	8
SQRT()	Returns the square root of a number.	SELECT SQRT(25)	5
LOG()	Returns the natural logarithm of a number.	SELECT LOG(10)	2.3026
LOG10()	Returns the base-10 logarithm of a number.	SELECT LOG10(1000)	3
EXP()	Returns the exponential value of a number.	SELECT EXP(1)	2.7182
SIGN()	Returns -1, 0, or 1 indicating the sign of a number.	SELECT SIGN(-10)	-1
RAND()	Returns a random floating-point number between 0 and 1.	SELECT RAND()	0.8732

PI()	Returns the value of PI.	SELECT PI()	3.14159
TRUNCATE()	Removes decimal places from a number (not available in SQL Server; use CAST() or CONVERT()).	SELECT CAST(12.345 AS INT)	12

### 3. List (Aggregate) Functions in MSSQL

Aggregate functions operate on multiple rows and return a single value.

Function	Description	Example	Output
SUM()	Returns the total sum of a numeric column.	SELECT SUM(Salary) FROM Employees	50000
AVG()	Returns the average value of a numeric column.	SELECT AVG(Salary) FROM Employees	25000
MIN()	Returns the smallest value in a column.	SELECT MIN(Salary) FROM Employees	10000
MAX()	Returns the largest value in a column.	SELECT MAX(Salary) FROM Employees	40000
COUNT()	Counts the number of rows in a column.	SELECT COUNT(*) FROM Employees	10
COUNT(DISTINCT column_name)	Counts distinct values in a column.	SELECT COUNT(DISTINCT Department) FROM Employees	3
STRING_AGG()	Concatenates values in a column, separated by a delimiter.	SELECT STRING_AGG(Name, ', ' ) FROM Employees	John, Lisa, Mike
GROUP_CONCAT()	Similar to STRING_AGG(), but not available in MSSQL (only in MySQL).	-	-

#### Example Query

```
sql
SELECT
    SUM(Salary) AS TotalSalary,
    AVG(Salary) AS AverageSalary,
    MIN(Salary) AS MinimumSalary,
    MAX(Salary) AS MaximumSalary,
    COUNT(*) AS TotalEmployees
FROM Employees;
```

### 4. Date and Time Functions in MSSQL

These functions work with date and time values.

Function	Description	Example	Output
GETDATE()	Returns the current date and time.	SELECT GETDATE()	2025-03-18 14:30:45
SYSDATETIME()	Returns the current date and time with higher precision.	SELECT SYSDATETIME()	2025-03-18 14:30:45.1234567
CURRENT_TIMESTAMP	Returns the current date and time (similar to GETDATE()).	SELECT CURRENT_TIMESTAMP	2025-03-18 14:30:45
DATEPART()	Extracts a part of a date (year, month, day, etc.).	SELECT DATEPART(YEAR, GETDATE())	2025
YEAR()	Extracts the year from a date.	SELECT YEAR(GETDATE())	2025
MONTH()	Extracts the month from a date.	SELECT MONTH(GETDATE())	3

<code>DAY()</code>	Extracts the day from a date.	<code>SELECT DAY(GETDATE())</code>	18
<code>DATENAME()</code>	Returns a specific part of the date as a string.	<code>SELECT DATENAME(MONTH, GETDATE())</code>	March
<code>DATEADD()</code>	Adds a specified interval to a date.	<code>SELECT DATEADD(DAY, 10, GETDATE())</code>	2025-03-28
<code>DATEDIFF()</code>	Returns the difference between two dates.	<code>SELECT DATEDIFF(YEAR, '2000-01-01', GETDATE())</code>	25
<code>EOMONTH()</code>	Returns the last day of the month.	<code>SELECT EOMONTH(GETDATE())</code>	2025-03-31
<code>FORMAT()</code>	Formats the date as a string.	<code>SELECT FORMAT(GETDATE(), 'yyyy-MM-dd')</code>	2025-03-18

#### Example Query

sql

Copy

Edit

```
SELECT
    GETDATE() AS CurrentDateTime,
    YEAR(GETDATE()) AS CurrentYear,
    MONTH(GETDATE()) AS CurrentMonth,
    DAY(GETDATE()) AS CurrentDay,
    DATEADD(DAY, 10, GETDATE()) AS TenDaysLater,
    DATEDIFF(YEAR, '2000-01-01', GETDATE()) AS YearsSince2000;
```

## Summary

- ✓ **Mathematical Functions** → Used for calculations (e.g., `ROUND()`, `SQRT()`, `ABS()`).
- ✓ **List (Aggregate) Functions** → Used for aggregation (e.g., `SUM()`, `AVG()`, `COUNT()`).
- ✓ **Date Functions** → Used for working with dates and times (e.g., `GETDATE()`, `DATEADD()`).

Would you like practice exercises or examples for these? 🚀

wow now tell me diff between single row function and multirow function with restrictions . show restrictions inside of query for bettr understanding. also at end in one line tell me answer of this " does list functions and group functions are same ? "

Great enthusiasm! Now, let's break this down step by step. 🚀

## Difference Between Single-Row and Multi-Row (Group) Functions in SQL Server

### 1 Single-Row Functions

- Operate on a **single** row and return **one** result per row.
- Can be used in the `SELECT`, `WHERE`, and `ORDER BY` clauses.

### Examples of Single-Row Functions

👉 Mathematical, String, and Date Functions fall under this category.

sql

Copy

Edit

```
SELECT
    UPPER(Name) AS UpperCaseName,
    ROUND(Salary, 2) AS RoundedSalary,
    GETDATE() AS CurrentDate
FROM Employees;
```

✓ Each row gets a transformed output without affecting other rows.



## 2 Multi-Row (Group) Functions

- Work on **multiple rows** and return a **single aggregated value**.
- Require the `GROUP BY` clause when used with column names.

### Examples of Multi-Row Functions

✎ Aggregate functions like `SUM()`, `AVG()`, `COUNT()`, `MAX()`, `MIN()`.



```
sql Copy Edit

SELECT
    Department,
    AVG(Salary) AS AvgSalary
FROM Employees
GROUP BY Department;
```

✓ Each department gets only one aggregated value for salary.



### ⛔ Restrictions of Multi-Row Functions (GROUP Functions)

- Cannot be used in the `WHERE` clause directly.
- Must use `HAVING` instead of `WHERE` to filter aggregated results.

#### ✗ Incorrect Query (Will Cause an Error)

```
sql Copy Edit

SELECT Department, AVG(Salary)
FROM Employees
WHERE AVG(Salary) > 50000 -- ✗ ERROR: Cannot use aggregate function in WHERE
GROUP BY Department;
```



#### ✓ Correct Query (Use HAVING Instead)

```
sql Copy Edit

SELECT Department, AVG(Salary) AS AvgSalary
FROM Employees
GROUP BY Department
HAVING AVG(Salary) > 50000;
```



## Final Answer: Are List Functions and Group Functions the Same?

No, they are not exactly the same!

✎ **List Functions** = All types of aggregate functions including `COUNT()`, `SUM()`, `AVG()`, etc.



✎ **Group Functions** = Subset of list functions used with `GROUP BY` (like `SUM(Salary) GROUP BY Department`).

🚀 Hope this clears everything up! Let me know if you need more examples. 😊

ok so give me list of group functions . also enlighten me about use of where do we use having , group by , orderby etc with query

Great! Now let's go step by step. ✖



## 📌 List of Group Functions in SQL Server

Group functions, also called **aggregate functions**, perform calculations on multiple rows and return a single value per group.

Function	Description	Example
<code>SUM()</code>	Adds up all values in a column.	<code>SELECT SUM(Salary) FROM Employees;</code>
<code>AVG()</code>	Calculates the average value.	<code>SELECT AVG(Salary) FROM Employees;</code>
<code>MIN()</code>	Finds the smallest value.	<code>SELECT MIN(Salary) FROM Employees;</code>
<code>MAX()</code>	Finds the largest value.	<code>SELECT MAX(Salary) FROM Employees;</code>
<code>COUNT()</code>	Counts the number of rows.	<code>SELECT COUNT(*) FROM Employees;</code>
<code>COUNT(DISTINCT column)</code>	Counts distinct values in a column.	<code>SELECT COUNT(DISTINCT Department) FROM Employees;</code>
<code>STRING_AGG()</code>	Concatenates column values separated by a delimiter.	<code>SELECT STRING_AGG(Name, ', ') FROM Employees;</code>

✅ Group functions work with `GROUP BY` to aggregate data.

## 📌 Where Do We Use GROUP BY, HAVING, ORDER BY, and WHERE?

Each of these clauses has a specific role in structuring SQL queries.

Clause	Purpose	Can Use Aggregate Functions?
<code>WHERE</code>	Filters rows <b>before</b> aggregation.	✖ No
<code>GROUP BY</code>	Groups rows <b>before</b> performing aggregation.	✖ No
<code>HAVING</code>	Filters groups <b>after</b> aggregation.	✅ Yes
<code>ORDER BY</code>	Sorts the result set.	✅ Yes

### 💡 Understanding with an Example

sql

📄 Copy 🛠 Edit

```
SELECT Department, AVG(Salary) AS AvgSalary
FROM Employees
WHERE Salary > 20000 -- ✅ Filters data before grouping
GROUP BY Department -- ✅ Groups employees by department
HAVING AVG(Salary) > 30000 -- ✅ Filters groups after aggregation
ORDER BY AvgSalary DESC; -- ✅ Sorts the result in descending order
```

### Step-by-Step Execution of SQL Query

- 1 `WHERE` → Filters employees where `Salary > 20000`.
- 2 `GROUP BY` → Groups remaining employees **by Department**.
- 3 `HAVING` → Removes groups where `AVG(Salary) <= 30000`.
- 4 `ORDER BY` → Sorts final result in descending order of `AvgSalary`.

## 📌 When to Use Each Clause?

- Use `WHERE` → When filtering individual rows **before** aggregation.
- Use `GROUP BY` → When you need aggregation based on a column (e.g., `SUM()`, `AVG()`).
- Use `HAVING` → When filtering **aggregated** values (like `AVG(Salary) > 30000`).
- Use `ORDER BY` → When sorting the final output.


Use ORDER BY when sorting the final output.


 Quick Trick:


✓ Always use WHERE before GROUP BY and HAVING after GROUP BY !


Let me know if you need more clarity! 😊

      Compare with Claude 



 Search

 Reason




ChatGPT can make mistakes. Check important info.



Remove City Distance Duplicat

Delete All Records SQL

Query Evaluation and Alternati

 Upgrade plan  
More access to the best models