

SQL Built-in functions

Agenda

- SQL Built-in functions
- String Functions
- More String Functions and Examples

SQL Built-in function

- MySQL ships with built-in functions.
- You can categorize them based on the type of data they operate on, e.g. strings, dates, and numeric built-in functions.

SQL Built-in function

- There are different types of SQL built - in functions, which are mentioned below:
 - String
 - Numeric
 - Date

SQL Built-in function

- We use the following Employee table to understand the built-in function:

ID	NAME	AGE	ADDRESS	SALARY
1	Kellie	32	California	2000
2	Pete	25	Texas	1500
3	Popy	23	Boston	2000
4	Sam	25	Florida	6500
5	Jhon	27	Hawaii	10000

String functions

This file is meant for personal use by maneel.chauhan@coforge.com only.
Sharing or publishing the contents in part or full is liable for legal action.

String functions

- String functions operate on an input string and return an output string.
- The following string functions are defined in SQL.

Functions	Meaning
ASCII	The ASCII code value of a character
CHAR	To convert an ASCII value to a character
CHARINDEX	The position of a substring within a string starting from a specified location is returned

More String functions

Functions	Meaning
CONCAT	Connects multiple strings into one string
LEFT	A given number of characters can be extracted from a character string starting at the left
LEN	The number of characters in a character string
LOWER	Changing a string to lowercase
LTRIM	A new string is returned after removing all leading blanks from a specified string

More String functions

Functions	Meaning
REPLACE	In a string, replace all occurrences of a substring with another substring.
REVERSE	The reverse order of a character string
RIGHT	A given number of characters can be extracted from a character string starting from the right
RTRIM	Removes all trailing blanks from a string and gives a new string.

More String functions

Functions	Meaning
SUBSTRING	From a specified location, extract a substring with a specified length
TRIM	Return a new string after removing all leading and trailing blanks from a specified string
UPPER	To convert a string to uppercase

String function - Example 1

LENGTH(): A word's length can be determined using this function.

Syntax:

```
SELECT length('MySQL') as  
CHAR_LEN;
```

Output:

CHAR_LEN
5

```
1 • SELECT length('MySQL') as CHAR_LEN;
```

Result Grid	
CHAR_LEN	
5	

String function - Example 2

CONCAT(): This function connects multiple strings into one string.

Syntax:

```
SELECT CONCAT( 'great',
'learning') as CONCAT_STR;
```

Output:

CONCAT_STR

great_learning

```
1 •   SELECT CONCAT( 'great', 'learning') as CONCAT_STR;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
CONCAT_STR				
greatlearning				

String function – Example 3

LCASE(): The function converts a string to lowercase.

Syntax:

```
SELECT LCASE ("Greatlearning To  
Learn")as LCASE;
```

Output:

LCASE
greatlearning to learn

12 • SELECT LCASE ("Greatlearning To Learn")as LCASE;
100% 19:12
Result Grid Filter Rows: <input type="text"/> Search Export:
LCASE ▶ greatlearning to learn

String function - Example 4

REPLACE(): This function shows all occurrences of one substring in a string should be replaced with another substring

Syntax:

```
SELECT REPLACE("Great Lakes",
    "Lakes", "Learning") as Replaced;
```

66 • SELECT REPLACE("Great Lakes", "Lakes", "Learning") as Replaced;	
	Result Grid Filter Rows: [] Export: [] Wrap Cell Content: []
	Replaced
►	Great Learning

Output:

Replaced

| Great Learning |

String function - Example 5

TRIM(): The function removes unwanted characters from a string.

Syntax:

```
SELECT TRIM( 'Great' from 'Great  
Learning') AS TrimmedString;
```

Output:

TrimmedString
Learning

Result Grid	Filter Rows:	Export:	Wrap Cell Content:		
88 • SELECT TRIM('Great' from 'Great Learning') AS TrimmedString;					
<table border="1"><thead><tr><th>TrimmedString</th></tr></thead><tbody><tr><td>Learning</td></tr></tbody></table>			TrimmedString	Learning	
TrimmedString					
Learning					

String function - Example 6

SUBSTR(): A substring is extracted from a string using this function.

Syntax:

```
SELECT SUBSTR("Great Learning", 1,  
5) AS ExtractedString;
```

Output:

ExtractedString
Great

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
87 • SELECT SUBSTR("Great Learning", 1, 5) AS ExtractedString;			
ExtractedString			
Great			

Summary

This slide explains the multiple string functions including the basic like CONCAT and REPLACE of string values using the queries.

The syntax of the query is also explained, as well as how to use the functions in the query.

SQL Built-in functions

Agenda

- SQL Built-in functions
- Numeric Functions
- More Numeric Functions and Examples

SQL Built-in function

- MySQL ships with built-in functions.
- Built-in functions are categorized by the type of data they operate on, i.e. strings, dates, and numerics.

SQL Built-in function

- There are different types of SQL built - in functions, which are mentioned below:
 - String
 - Numeric
 - Date

SQL Built-in function

- We use the following Employee table to understand the built-in function:

ID	NAME	AGE	ADDRESS	SALARY
1	Kellie	32	California	2000
2	Pete	25	Texas	1500
3	Popy	23	Boston	2000
4	Sam	25	Florida	6500
5	Jhon	27	Hawaii	10000

Numeric functions

This file is meant for personal use by maneel.chauhan@coforge.com only.
Sharing or publishing the contents in part or full is liable for legal action.

Numeric function - Syntax

- To perform any mathematical operation in a query, use the following syntax.

Syntax:

```
SELECT numerical_expression as OPERATION_NAME  
[FROM table_name WHERE CONDITION] ;
```

Numeric function - Example 1

Numerical expressions are used for expressing mathematical expressions and formulas. The following example illustrates how to use SQL numeric expressions .

Syntax:

```
SELECT (25 + 7) AS ADDITION
```

Output:

ADDITION
32

The screenshot shows a database interface with the following details:

- Toolbar:** Includes icons for file operations, search, and other database functions, along with a "Limit to 50" button.
- Query Editor:** Displays the SQL query:

```
1 •  SELECT (25 + 7) AS ADDITION
2
```
- Result Grid:** Shows the output of the query in a tabular format:

ADDITION
32
- Zoom and View Options:** Includes a zoom slider set to 100% and a view mode indicator showing 5:1.
- Filter Row:** A search bar labeled "Search" with a magnifying glass icon.

Numeric function - Example 2

numerical_expressions must contain a mathematical expression or formula. The following example illustrates the use of numerical expressions in SQL.

Syntax:

```
SELECT COS(1) As Cos;
```

Output:

Cos
0.5403023058681398

4 •	SELECT COS(1) As Cos;
100%	23:4
Result Grid	Filter Rows:
Cos	
0.5403023058681398	

Numeric function - Example 3

numerical_expressions must contain a mathematical expression or formula. The following example illustrates the use of numerical expressions in SQL.

Syntax:

```
SELECT 20 DIV 6 As DIVISION;
```

Output:

DIVISION
3

6 • **SELECT 20 DIV 6 As DIVISION;**

00% 17:6

Result Grid Filter Rows: Search

DIVISION
3

More numeric functions

- Following are the numeric functions defined in SQL:

Functions	Meaning
ABS	Provides absolute value of a number
ACOS	Provides the cosine of a number
ASIN	Provides the arc sine of a number
ATAN	It provides the arc sine of a numerical value.
CEIL	Provides the smallest integer value that is greater than or equal to a number
CEILING	Provides the smallest integer value that is greater than or equal to a number

More numeric functions

Functions	Meaning
COS	Provides the cosine of a number
COT	Provides the cotangent of a number
DEGREES	Provides a radian value into degrees
DIV	Generally used for the integer division
EXP	The result is e raised to the power of the number
FLOOR	Provides the largest integer value that is less than or equal to a number

More numeric functions

Functions	Meaning
GREATEST	The greatest value in a list of expressions is returned
LEAST	The smallest value in a list of expressions is returned
LN	The natural logarithm of a number is returned
LOG10	The base-10 logarithm of a number is returned
LOG2	The base-2 logarithm of a number is returned
MOD	The remainder of n divided by m is returned

More numeric functions

	Meaning
PI	It gives the value of PI (up to six decimal places).
POW	It gives m^n .
RADIANS	It changes the degrees into radians.
RAND	It returns a random number
ROUND	It gives the rounded number to the specific decimal place.
SIGN	An indication of a number's sign is returned.

More Numeric Functions

Functions	Meaning
SIN	The sine of a number is returned
SQRT	The square root of a number is returned
TAN	The tangent of a number is returned
ATAN2	The arctangent of the x and y coordinates, as an angle and expressed in radians is returned
TRUNCATE	This doesn't work for SQL Server. It returns 7.53635 truncated to 2 places right of the decimal point

Summary

This slide explains the multiple numeric functions including the basic like addition and subtraction of values using the queries

The syntax of the query is also explained, as well as how to use the functions in the query.

SQL Built-in functions

Agenda

- SQL Built-in functions
- CAST Function
- COALESCE Function
- Sorting Query Result

CAST function

This file is meant for personal use by maneel.chauhan@coforge.com only.
Sharing or publishing the contents in part or full is liable for legal action.

CAST function - Syntax

- **CAST()** → This function converts any type of value into the specified datatype.

Syntax:

```
CAST (value AS datatype)
```

Argument:

Parameter	Description
value	Required. The selected value we want to convert
datatype	Required. The datatype to convert to

CAST function - Example 1

- Converting a value to a CHAR data type:

Syntax:

```
SELECT CAST(150 AS CHAR);
```

Output:

Number of Records: 1
CAST(150 AS CHAR)
150

CAST function - Example 2

- Converting a value to a TIME datatype:

Syntax:

```
SELECT CAST("14:06:10" AS TIME);
```

Output:

Number of Records: 1
CAST("14:06:10" AS TIME)
14:06:10

COALESCE function

This file is meant for personal use by maneel.chauhan@coforge.com only.
Sharing or publishing the contents in part or full is liable for legal action.

COALESCE function - Syntax

- COALESCE() → Provides first non-null values in the list.

Syntax:

```
COALESCE(value1, value2, . . . , value_n)
```

Parameter Values:

Parameter	Description
<i>value1, value2, till value_n</i>	It is required because these are the values we want to test.

COALESCE function - Example

- Return the first non-null value in a list

```
SELECT COALESCE(NULL, 1, 2, 'greatlearning.in');
```

Output:

Number of Records: 1
COALESCE(NULL, 1, 2, 'greatlearning.in')
1

The first value after the NULL value is '1' , so the output is 1

Sorting Query result

This file is meant for personal use by maneel.chauhan@coforge.com only.
Sharing or publishing the contents in part or full is liable for legal action.

Sorting Query results

Consider the following table for the study:

UserID	Name	City	Salary
1	Atanu Chakraborty	Kolkata	15000
2	Hriday Bharadwaj	Lucknow	27000
3	Birendra Maity	Hyderabad	68000
4	Raj Patil	Bangalore	71000
22	Bikram Ganguly	Kolkata	51000
31	Tapan Chatterjee	Kolkata	25000
33	Jai Prakash Bhatt	Jaipur	81000
34	Keith Kosta	Mumbai	77000
35	Mark Tailor	Bangalore	52000
36	Christopher Swan	Null	98000
39	Paul Grant	Mumbai	67000

Sorting Query results

- Using the ***order by*** clause, the tuples in the result of a query are sorted.
- Let us write a query which will show the users according to the ascending order of their salary.

```
SELECT UserID, Name, Salary FROM tblUser ORDER BY Salary
```

Output:

UserID	Name	Salary
1	Atanu Chakraborty	15000
31	Tapan Chatterjee	25000
2	Hriday Bharadwaj	27000
22	Bikram Ganguly	51000
35	Mark Tailor	52000
39	Paul Grant	67000
3	Birendra Maity	68000
4	Raj Patil	71000
34	Keith Kosta	77000
33	Jai Prakash Bhatt	81000
36	Christophar Swan	98000

The ***order by*** clause lists items in ascending order by default.

Sorting Query results

- To specify the sort order we used: **desc** → descending order or **asc** → ascending order.
- Now let us see an example to show the users according to the descending order of their salary.

```
SELECT UserID, Name, Salary FROM tblUser ORDER BY Salary desc
```

Output:

UserID	Name	Salary
36	Christophar Swan	98000
33	Jai Prakash Bhatt	81000
34	Keith Kosta	77000
4	Raj Patil	71000
3	Birendra Maity	68000
39	Paul Grant	67000
35	Mark Tailor	52000
22	Bikram Ganguly	51000
2	Hriday Bharadwaj	27000
31	Tapan Chatterjee	25000
1	Atanu Chakraborty	15000

Sorting Query results

- Furthermore ordering can be performed on multiple attributes.
- Suppose we wish to order the entire tblUser relation in ascending order of City and then descending order of amount. Then we express this query in SQL as follows.

```
SELECT UserID, Name, Salary FROM tblUser ORDER BY City desc, Salary asc
```

Output:

UserID	Name	City	Salary
39	Paul Grant	Mumbai	67000
34	Keith Kosta	Mumbai	77000
2	Hriday Bharadwaj	Lucknow	27000
1	Atanu Chakraborty	Kolkata	15000
31	Tapan Chatterjee	Kolkata	25000
22	Bikram Ganguly	Kolkata	51000
33	Jai Prakash Bhatt	Jaipur	81000
3	Birendra Maity	Hyderabad	68000
35	Mark Tailor	Bangalore	52000
4	Raj Patil	Bangalore	71000
36	Christophar Swan	Null	98000

Summary

This slide explains the cast function and the coalesce function with its usage in queries by following certain examples.

Sorting queries is also explained by using the *orderby* clause.

The syntax of the query is also explained, as well as how to use the functions in the query.