

String Functions in SQL

1. **ASCII:** Returns the ASCII code value of the leftmost character of a character expression.

Syntax: ASCII (character_expression)

Example: SELECT ASCII('A')

2. **CHAR:** Returns the single-byte character with the specified integer code, as defined by the character set and encoding of the default collation of the current database.

Syntax: CHAR (integer_expression)

Example: Select CHAR(97)

3. **LEFT():** Returns the left part of a character string with the specified number of characters.

Syntax: LEFT (character_expression , integer_expression)

Example: SELECT LEFT('pandeyGuruji',3)

Select LEFT(FirstName,3) as First4Letter , * from DimEmployee

4. **RIGHT():** Returns the right part of a character string with the specified number of characters.

Syntax: RIGHT (character_expression , integer_expression)

Example: SELECT right('PandeyGuruji',4)

Select right(FirstName,4) , * from DimEmployee

5. **LTRIM():** Returns a character expression after it removes leading blanks.

Syntax: LTRIM (character_expression)

Example: SELECT LTRIM(' PandeyGuruji')

6. **RTRIM():** Returns the right part of a character string with the specified number of characters.

Syntax: RTRIM (character_expression , integer_expression)

Example: SELECT RTRIM(' PandeyGuruji ')

SELECT LTRIM(RTRIM(' PandeyGuruji '))

7. REPLACE(): Replaces all occurrences of a specified string value with another string value.

Syntax: REPLACE (string expression , string pattern , string replacement)

Example: SELECT REPLACE('My name is Mukesh','Mukesh', 'Vansh')

```
Select REPLACE(EmailAddress,'adventure-works.com','Graphy.com') as  
NewEmailAddress,EmailAddress,* from DimEmployee
```

8. STUFF: The STUFF function inserts a string into another string. It deletes a specified length of characters in the first string at the start position and then inserts the second string into the first string at the start position.

Syntax: STUFF (character_expression , start , length , replaceWith_expression)

Example: SELECT STUFF(Phone,3,8,'****'),Phone, * from DimEmployee

9. REVERSE(): Returns the reverse order of a string value.

Syntax: REVERSE (string_expression)

Example: Select REVERSE(FirstName),* from DimEmployee

10. CHARINDEX: This function searches for one character expression inside a second character expression, returning the starting position of the first expression if found.

Syntax: CHARINDEX (expressionToFind , expressionToSearch [, start_location])

Example: SELECT CHARINDEX('SQL', 'Microsoft SQL Server')

```
select CHARINDEX('@',EmailAddress),EmailAddress, * from DimEmployee
```

```
select CHARINDEX('@',FirstName), * from DimEmployee
```

11. PATINDEX: Returns the starting position of the first occurrence of a pattern in a specified expression, or zeros if the pattern is not found, on all valid text and character data types.

Syntax: PATINDEX ('%pattern%' , expression)

Example: Select PATINDEX('%CD%','ABCDhhjhighighjhjh')

```
Select PATINDEX('%ter%', 'interesting data')
```

12. LEN: Returns the number of characters of the specified string expression, excluding trailing spaces.

Syntax: LEN (string_expression)

Example: select LEN(FirstName).* from DimEmployee

```
select LEN>Title).* from DimEmployee
```

13. SUBSTRING: Returns part of a character

Syntax: SUBSTRING (expression ,start , length)

Example: SELECT SUBSTRING('SQL SERVER is USEFUL', 5, 3)

Select SUBSTRING(EmailAddress,1,CHARINDEX('@',EmailAddress)) as NTID,EmailAddress, * from DimEmployee

14. LOWER: Returns a character expression after converting uppercase character data to lowercase.

Syntax: LOWER (character_expression)

Example: Select LOWER(FirstName),* from DimEmployee

15. UPPER: Returns a character expression with lowercase character data converted to uppercase.

Syntax: UPPER (character_expression)

Example: Select UPPER(FirstName),* from DimEmployee

16. Space: Returns a string of repeated spaces.

Syntax: SPACE (integer_expression)

Example: Select (FirstName + Space(1) + LastName) as FullName, * from DimEmployee

17. Concat: This function returns a string resulting from the concatenation, or joining, of two or more string values in an end-to-end manner.

Syntax: CONCAT (string_value1, string_value2 [, string_valueN])

Example: Select CONCAT(FirstName,Space(1),LastName) as FullName, * from DimEmployee