**Built-in functions in SQL Server**

**Mathmetical functions**

**ABS ( numeric\_expression )** - ABS stands for absolute and returns, the absolute (positive) number.   
  
**For example**, Select ABS(-101.5) -- returns 101.5, without the - sign.

**CEILING ( numeric\_expression ) and FLOOR ( numeric\_expression )**  
**CEILING**and **FLOOR** functions accept a numeric expression as a single parameter. CEILING() returns the smallest integer value greater than or equal to the parameter, whereas FLOOR() returns the largest integer less than or equal to the parameter.   
  
**Examples:**  
Select CEILING(15.2) -- Returns 16  
Select CEILING(-15.2) -- Returns -15  
  
Select FLOOR(15.2) -- Returns 15  
Select FLOOR(-15.2) -- Returns -16  
  
**Power(expression, power)** - Returns the power value of the specified expression to the specified power.  
  
**Example**: The following example calculates '2 TO THE POWER OF 3' = 2\*2\*2 = 8  
Select POWER(2,3) -- Returns 8  
  
**RAND([Seed\_Value])** - Returns a random float number between 0 and 1. Rand() function takes an optional seed parameter. When seed value is supplied the   
  
RADN() function always returns the same value for the same seed.  
  
**Example:**  
Select RAND(1) -- Always returns the same value  
  
**If you want to generate a random number between 1 and 100**, RAND() and FLOOR() functions can be used as shown below. Every time, you execute this query, you get a random number between 1 and 100.  
Select FLOOR(RAND() \* 100)  
  
**The following query prints 10 random numbers between 1 and 100.**  
Declare @Counter INT  
Set @Counter = 1  
While(@Counter <= 10)  
Begin  
 Print FLOOR(RAND() \* 100)  
 Set @Counter = @Counter + 1  
End  
  
**SQUARE ( Number )** - Returns the square of the given number.  
  
**Example:**  
Select SQUARE(9) -- Returns 81  
  
**SQRT ( Number )** - SQRT stands for Square Root. This function returns the square root of the given value.  
  
**Example:**  
Select SQRT(81) -- Returns 9  
  
**ROUND ( numeric\_expression , length [ ,function ] )** - Rounds the given numeric expression based on the given length. This function takes 3 parameters.   
**1. Numeric\_Expression** is the number that we want to round.

**2. Length parameter**, specifies the number of the digits that we want to round to. If the length is a positive number, then the rounding is applied for the decimal part, where as if the length is negative, then the rounding is applied to the number before the decimal.  
**3. The optional function parameter**, is used to indicate rounding or truncation operations. A value of 0, indicates rounding, where as a value of non zero indicates truncation. Default, if not specified is 0.  
  
**Examples:**  
-- Round to 2 places after (to the right) the decimal point  
Select ROUND(850.556, 2) -- Returns 850.560  
  
-- Truncate anything after 2 places, after (to the right) the decimal point  
Select ROUND(850.556, 2, 1) -- Returns 850.550  
  
-- Round to 1 place after (to the right) the decimal point  
Select ROUND(850.556, 1) -- Returns 850.600  
  
-- Truncate anything after 1 place, after (to the right) the decimal point  
Select ROUND(850.556, 1, 1) -- Returns 850.500  
  
-- Round the last 2 places before (to the left) the decimal point  
Select ROUND(850.556, -2) -- 900.000  
  
-- Round the last 1 place before (to the left) the decimal point  
Select ROUND(850.556, -1) -- 850.000

**String Functions**

ASCII(Character\_Expression) - Returns the ASCII code of the given character expression.  
To find the ACII Code of capital letter 'A'  
 **Example:** Select ASCII('A')  
**Output:** 65  
  
CHAR(Integer\_Expression) - Converts an int ASCII code to a character. The Integer\_Expression, should be between 0 and 255.  
The following SQL, prints all the characters for the ASCII values from o thru 255  
  
Declare @Number int  
Set @Number = 1  
While(@Number <= 255)  
Begin  
 Print CHAR(@Number)  
 Set @Number = @Number + 1  
End  
  
**Note:** The while loop will become an infinite loop, if you forget to include the following line.  
Set @Number = @Number + 1  
  
**Printing uppercase alphabets using CHAR() function:**  
Declare @Number int  
Set @Number = 65  
While(@Number <= 90)  
Begin  
 Print CHAR(@Number)  
 Set @Number = @Number + 1  
End  
  
**Printing lowercase alphabets using CHAR() function:**  
Declare @Number int  
Set @Number = 97  
While(@Number <= 122)  
Begin  
 Print CHAR(@Number)  
 Set @Number = @Number + 1  
End  
  
  
**Another way of printing lower case alphabets using CHAR() and LOWER() functions.**  
Declare @Number int  
Set @Number = 65  
While(@Number <= 90)  
Begin  
 Print LOWER(CHAR(@Number))  
 Set @Number = @Number + 1  
End  
  
LTRIM(Character\_Expression) - Removes blanks on the left handside of the given character expression.  
  
**Example**: Removing the 3 white spaces on the left hand side of the '   Hello' string using LTRIM() function.  
Select LTRIM('   Hello')  
**Output**: Hello  
  
RTRIM(Character\_Expression) - Removes blanks on the right hand side of the given character expression.  
 **Example**: Removing the 3 white spaces on the left hand side of the 'Hello   ' string using RTRIM() function.  
Select RTRIM('Hello   ')  
**Output**: Hello  
  
**Example**: To remove white spaces on either sides of the given character expression, use LTRIM() and RTRIM() as shown below.  
Select LTRIM(RTRIM('   Hello   '))  
**Output**: Hello  
  
LOWER(Character\_Expression) - Converts all the characters in the given Character\_Expression, to lowercase letters.  
  
**Example**: Select LOWER('CONVERT This String Into Lower Case')  
**Output**: convert this string into lower case  
  
UPPER(Character\_Expression) - Converts all the characters in the given Character\_Expression, to uppercase letters.  
**Example**: Select UPPER('CONVERT This String Into upper Case')  
**Output**: CONVERT THIS STRING INTO UPPER CASE  
  
REVERSE('Any\_String\_Expression') - Reverses all the characters in the given string expression.  
**Example**: Select REVERSE('ABCDEFGHIJKLMNOPQRSTUVWXYZ')  
**Output**: ZYXWVUTSRQPONMLKJIHGFEDCBA  
  
LEN(String\_Expression) - Returns the count of total characters, in the given string expression, excluding the blanks at the end of the expression.  
  
**Example**: Select LEN('SQL Functions   ')  
**Output**: 13  
  
