# IMP SQL FUNCTIONS LIST

- 1) SQL STRING FUNCTIONS
- 2) SQL DATE FUNCTIONS
- 3) SQL WINDOWS FUNCTIONS
- 4) SQL CONDITIONAL FUNCTIONS
- 5) SQL AGGREGATES FUNCTIONS
- 6) SQL ARITHMETIC FUNCTIONS
- 7) SQL USER DEFINED FUNCTIONS



# **IMP SQL String Functions**



NCHAR()

RANSFORMATION	NC	AL
SUBSTRING()		Extracts character from given string as per input position.
LTRIM()		Removes leading spaces from string.
RTRIM()		Removes trailing spaces from string.
LOWER()		Converts string to lower case.
UPPER()	X	Converts string to upper case.
CONCAT()		Combines 2 or more string to a single string.
REPLACE()		Replaces all occurrences of a substring with another string.
STUFF()		Deletes given section of string and replace with another string.
SPACE()		Returns a string with specified number of space characters.
REVERSE()	X	Reverse the sequence of the string.
REPLICATE()	X	Repeats the string with no. of times of given input parameter.
POSITIONAL		
LEN()	X	Returns the total length of the string.
CHARINDEX()	×	Returns the first position of given character.
RIGHT()	×	Extracts no of given string characters from right position.
PATINDEX()	×	Returns the position of pattern in the string.
CONVERSION		
ASCII()	X	Returns ASCII value of specific character.
CHAR()	X	Returns character based on ASCII code.
Annual Control of the		

UNICODE() Returns unicode value of first character of input expression.

Returns unicode character based on number code.

### **IMPORTANT SQL DATE FUNCTIONS**

GETDATE()

Returns current date and time of system's timezone

ISDATE(date)

Returns 1 if input parameter is a valid date else return 0

CURRENT\_TIMESTAMP

Returns current date and time of database's timezone

CURRENT TIMEZONE()

Returns timezone name

DATEADD(day,num,date)

It adds day, month or year in the date given based on number provided in second parameter. Negative number will subtract the days.

DATEDIFF(day,date1,date2)

It returns the day, month or year difference between the two dates provided. The output is integer.

DATEPART(year,date)

It returns the day, month or year of the date provided in input. The output is integer.

DAY(date)

It returns the day of the date provided in input. The output is integer.

MONTH(date)

It returns the month of the date provided in input. The output is integer.

YEAR(date)

It returns the Year of the date provided in input. The output is integer.

@minalbhatkar

# **SQL Window Functions**

ROW_NUMBER()	Assigns a unique sequential integer to each row within the result set, starting from 1.	
RANK()	Assigns a rank to each row in a result set based on specified column values, with tied values receiving the same rank.	
DENSE_RANK()	Assigns a rank to each row within a result set based on specified column values, with tied values receiving the same rank, but there are no gaps in the ranking.	
NTILE(n)	Divides the result set into 'n' roughly equal parts and assigns a group number to each row.	
LAG()	Accesses data from the previous row within result set.	
LEAD()	Accesses data from the next row within the result set.	
SUM() OVER()	Perform sum aggregation operations over a specified window or partition of rows.	
AVG() OVER()	Perform Average aggregation operations over a specified window or partition of rows.	
COLINIT() OVER()	Counts the number of rows within a specified window or	

partition.



# **SQL Conditional Functions**

CASE	The CASE statement lets you apply conditional logic to one or multiple conditions, resulting in diverse outcomes.
ISNULL	Functions designed to address NULL values by returning a default value when a column or expression is NULL.
COALESCE	Returns first non-null expression from a list of expressions.
NULLIF	Compares two expressions and returns null if they are equal; otherwise, it returns the first expression.
CHOOSE	Selects a value based on an index.
BETWEEN	Used in where clause for range comparisons.
LIKE	Used in where clause for pattern matching
EXISTS	It checks for the existence of rows in a subquery.
ANY/ALL	Used for comparison operators in subqueries.
IF-ELSE	Conditional construct to check condition and return subsequent value.



## **SQL** Aggregate and Arithmatic Functions

#### AGGREGATE FUNCTIONS

SUM()	Provides the cumulative sum of a numerical column	
JUIVIU	A Provides the cultiviative sum of a numerical column	

AVG() The function computes the avarage of a collection of	of values.
--	------------

THUS THE LOWEST VOLUE HOLL SIVEH COLUMN.	MIN()	Finds	the lowest	value from	given column.
--	-------	-------	------------	------------	---------------

MAX()	Finds th	ne highest	value from	n given column.
-------	----------	------------	------------	-----------------

#### **ARITHMATIC FUNCTIONS**

ABS()	Provides the absolute value of a given numeric argument.
-------	--

CEIL()	Rounds up positive or negative decimal value to next integer.
--------	---

FLOOR() Rounds up positive or negative decimal val to least integer.

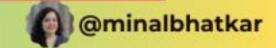
EXP() Computes e to the power of a numeric expression (n).

LN() Returns the natural logarithm of a positive number (n).

MOD() Function provides the remainder after division.

POWER() Computes first parameter raised second parameter.

SQRT() Calculates the square root of the provided argument.



# **USER DEFINED SQL FUNCTION**

#### Syntax:

```
CREATE FUNCTION [schema_name.]function_name
  (@parameter1 data_type, @parameter2 data_type, ...)
RETURNS return_data_type

AS
BEGIN
-- Function body: Define the logic of your function here
-- Use parameters and local variables as needed
```

-- Return a value or a result set END;

#### **Example:**

```
CREATE FUNCTION dbo.AddNumbers
(@num1 INT, @num2 INT)
RETURNS INT
AS
BEGIN
DECLARE @result INT;
SET @result = @num1 + @num2;
RETURN @result;
END;
```

