**John Winslow**

**11/30/2022**

**IT FDN 130**

**Assignment 07**

**<https://github.com/jwins66/DBFoundations>-Module07**

**Functions**

**Introduction:**

This module was about learning how to use SQL Functions to retrieve information from a database. The question to answer in the assignment are listed below:

**When do you use a SQL UDF (User Defined Function)?**

## User-defined functions are routines that accept parameters, perform an action, and return the result as a single scalar (single) value or a result set. The purpose of scalar functions is to encapsulate a piece of code into the single database object, and in this way, it enables us to use functions in the queries without code repetition.

## Benefits of user-defined functions

Why use user-defined functions (UDFs)?

* **Modular programming.** You can create the function once, store it in the database, and call it any number of times in your program. User-defined functions can be modified independently of the program source code.
* **Faster execution.** Similar to stored procedures, Transact-SQL user-defined functions reduce the compilation cost of Transact-SQL code by caching the plans and reusing them for repeated executions. This means the user-defined function doesn't need to be reparsed and reoptimized with each use resulting in much faster execution times.

CLR functions offer significant performance advantage over Transact-SQL functions for computational tasks, string manipulation, and business logic. Transact-SQL functions are better suited for data-access intensive logic.

* **Reduce network traffic.** An operation that filters data based on some complex constraint that can't be expressed in a single scalar expression can be expressed as a function. The function can then be invoked in the WHERE clause to reduce the number of rows sent to the client.

[**https://learn.microsoft.com/en-us/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-ver16**](https://learn.microsoft.com/en-us/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-ver16)

## Specify parameters

A user-defined function takes zero or more input parameters and returns either a scalar value or a table. A function can have a maximum of 1024 input parameters. When a parameter of the function has a default value, the keyword DEFAULT must be specified when calling the function to get the default value. This behavior is different from parameters with default values in user-defined stored procedures in which omitting the parameter also implies the default value. User-defined functions don't support output parameters.

[**https://learn.microsoft.com/en-us/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-ver16**](https://learn.microsoft.com/en-us/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-ver16)

**What are the differences between Scalar, Inline and Multi-Statement Functions:**

As referenced in the answer to question one, Scalar refers to a single value or result set each time the user-defined function is invoked. Scalar functions contain the source code for the user-defined function within the user-defined function definition.

An Inline Statement allows you to group multiple SQL statements (Compound) into an optionally atomic block (combination of several operations in one operation) in which you can declare variables, and condition handling elements. Therefore, when one operation has failed in the block, other operations will also be failed.

A Multi-Statement Function is a user-defined function that combines the scalar function’s capability to contain complex code with the inline table-valued function’s capability to return a result set. This type of function creates a table variable and then populates it within the code. The table is then passed back from the function so that it may be used within SELECT statements. The primary benefit of the multi-statement table-valued user-defined function is that complex result sets may be generated within code and then easily used with a SELECT statement. <https://www.oreilly.com/library/view/microsoft-sql-server/9781118282175/c18_level1_3.xhtml>

**Summary:**

As referenced by w3resource, a function is a predefined formula which takes one or more arguments as input then processes the arguments and returns an output. This can be done simply with a scalar function. Or, more complicated with aggregate functions that operate on many records and produce a summary. Using GROUP BY with aggregate functions versus non-aggregate functions operate on each record independently.

<https://www.w3resource.com/sql/sql-functions.php>