Melinda Hull

December 2, 2020

IT Foundations of Database Management

Assignment 07

https://github.com/Caprica8/DBFoundations-Module07

Function is as Function does

# Introduction

The SQL Server toolbox provides a myriad of tools for manipulating data, getting things done and pinpointing specific pieces of data or “characteristics” of data. We have previously focused on views as a tool for, as the name suggests, viewing data that has been stored in a data base and with the use of where, group by, order by and having clauses allow for filtering and manipulating of the underlying data. However, sometimes the developer/user wants to take this filtering and manipulation to another level and that is where functions can come in handy.

# User Defined Functions - A Tool for the Perpetually Unsatisfied

Microsoft SQL developers sat down and wrote out literally hundreds of Transact SQL built-in functions that are included in SQL Server in twenty-two different categories (yes, I counted them up!). However, SQL developers/users often look at this abundance of riches and say *“that’s all you got!?? I want this and this and this! And I want it* ***NOW****!”* Rather than the Microsoft SQL developers gritting their teeth and saying, “y*es, madame, right away madame”,* they say, “*here are the keys to the function tools, go do it yourself!”* Enter User Defined Functions or UDFs which allow developers/users to create their own functions in code snippets specifically tailored to the needs of a specific database and/or user.

Microsoft itself defines UDFs as “[L]ike functions in programming languages, SQL Server user-defined functions are routines that accept parameters, perform an action, such as a complex calculation, and return the result of that action as a value. The return value can either be a single scalar value or a result set” [1] and they further list the benefits of UDFs as (1) facilitation of modular programming; (2) faster execution; and (3) reduction of network traffic [2].

# UDFs Come in Different Flavors

UDFs functions can be created in three broad types: Scalar, Inline and Multi-Statement Functions or more simply put, “Small, Medium and Large”. A Scalar function is created when the developer wants to input one or more parameters and output a single parameter. Daniel Hutmacher at sqlsunday.com defines an Inline function as similar to a view whereby it can only can use a single Select statement and the columns in the Select statement implicitly define the columns of the returned table set of the function [3]. He contrasts the Inline function with the Multi-Statement function which acts more as a stored procedure that accepts arguments and returns a table recordset [4].

# Summary

Functions are an integral tool in SQL Server. Any developer who wants to be able to truly get the most out of any database will learn to use the built-in functions, but also learn to write and use UDFs in the form of Scalar, Inline and Multi-Statement Functions as described herein.

1. Microsoft, “*What are the SQL database functions?”* at https://docs.microsoft.com/en-us/sql/t-sql/functions/functions?redirectedfrom=MSDN&view=sql-server-ver15 accessed on December 1, 2020.
2. Microsoft, *ibid.*
3. Hutmacher D, “*Table value functions vs inline functions”* at sqlsunday.com <https://sqlsunday.com/2013/05/05/table-value-vs-inline-table-functions/> accessed on December 1, 2020.
4. Hutmacher D, *ibid.*