Steven Sparks

March 3, 2021

Foundations of Databases & SQL Programming

Assignment 07

<https://github.com/ssparky77/DBFoundations-Module07.git> (external site)

More on Functions in Microsoft SQL Server

# Introduction

This paper discusses using Functions in a relational database like Microsoft SQL Server. SQL User-Defined Functions (“UDFs”) are routines that accept parameters, perform actions, and return the result as a value, just like Functions in other programming languages. Scalar, Inline, and Multi-Statement Functions can return values as a single result or in tables, and each has some unique syntax.

# When to Use a UDF in SQL

User-Defined Functions allow for more efficient programming, faster execution, and reduced network traffic than other Microsoft SQL Server features that yielding similar results.

1. **Efficiency.** Developers can create and store a UDF once, call it back an unlimited amount of times in the program, and modify the UDF independently of the database source code.
2. **Speed.** UDFs reduce the cost of compiling SQL code by parsing and optimizing the plans, then caching and using them for each repeated execution resulting in significantly faster execution times.
3. **Reduced Traffic.** Developers can express complex operations as a Function and then call them the Where clause to lessen the number of individual rows sent to the server client.

(Microsoft website, <https://docs.microsoft.com/en-us/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-ver15> (external site), 2021.03.02).

# Scalar, Inline, and Multi-Statement Functions: Differences

Scalar Functions return a single value of the data type the programmer defines in the Returns clause. Scalar Functions work on each record independently and base their results on user input. For Inline Functions, the returned value results from a single Select statement. Inline Functions are similar to Views because they can contain only a single Select statement and the columns in the Select statement become the column names in the returned table. A Multi-Statement function can include several Select statements that return a single value. Multi-Statement Functions require the Begin and End syntax and generally execute more slowly than Inline Functions. The data type specified in the Returns clause of these Functions can be any except Text, Ntext, Image, Cursor, and Timestamp. (Database.Guide website, <https://database.guide/difference-between-multi-statement-table-valued-functions-inline-table-valued-functions-in-sql-server/> (external site), 2021.03.02). See figures 1 through 3 below for a code snippet highlighting the differences between each of the Functions mentioned above visually.

### Select

### ProductName

### ,[UnitPrice] = Format(UnitPrice, 'C')

### From vProducts

### Order by ProductName;

### go

### (Figure 1: Sample Scalar Function Code.)

Select

stor\_id

,qty

,[store min was] = min(qty) over(partition by stor\_id)

,[store max was] = max(qty) over(partition by stor\_id)

From Pubs.dbo.sales

Order By stor\_id;

Go

### (Figure 2: Sample Inline Function Code.)

Select

ProductName

,[OrderYear]

,YearlyTotalQty

,PreviousYearlyTotalQty

,[QtyChangeKPI] = Case

When YearlyTotalQty > PreviousYearlyTotalQty Then 1

When YearlyTotalQty = PreviousYearlyTotalQty Then 0

When YearlyTotalQty < PreviousYearlyTotalQty Then -1

End

From vProductOrderQtyByYear

Go

### (Figure 3: Sample Multi-Statement Function Code.)

# Summary

Microsoft SQL Server contains powerful tools used to store complex calculations for repetitious use. User-Defined Functions allow programmers to create and store their own custom Functions within the database files rather than the script. Programmers must know the varying syntax differences and use cases for developers to fully realize the capabilities and efficiencies Functions offer in Microsoft SQL Server.