Bryan Mon

February 24, 2021

IT FDN 130 A Wi 21: Foundations of Databases & SQL Programming

Assignment 07

<https://github.com/bmon1216/DBFoundations-Module07.git>

**SQL Functions**

# Introduction

In our previous lessons, we were introduced to some of the built in SQL functions such as an aggregate function. As we discovered, we are able to parse data through any defined calculations and have it return a result. Our current week assignment builds on this lesson by introducing the user defined function (UDF). We learned about the different types of UDF’s and how to create them in SQL server.

# User Defined Functions

A UDF is a custom function which accepts a parameter, does an action, and then returns the result of this action. Because a UDF is created by us, we can store the function in the database where it can be called any number of times. The customization of these functions allows for flexibility which can provide detail reporting along with improvements to the database performance. UDF’s also range in complexity. A simple design, such as seen in figure 7.1, can be used to perform a basic multiplication action. Note that the “dbo.” is included in the function name. UDF’s require the schema name when being defined.

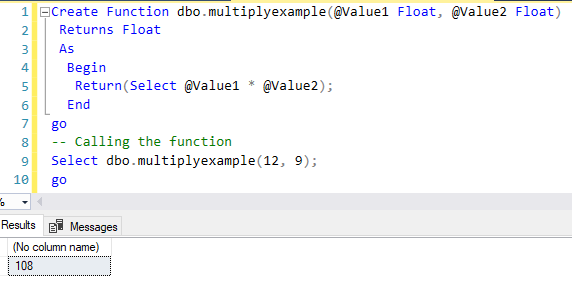


Figure 7.1 – UDF for multiplication, where selected input is 12 & 9

# Scalar, Inline, and Multi-Statement Functions

There are three types of UDF’s in SQL. The first type is a Scalar function, which is a UDF that accepts zero or more parameters and then returns a single value.

The second type of UDF is the Inline function. The Inline function will return a table of data instead of just a single value. One important note about the Inline Function is that it is limited to only using a single Select statement. This means that the returned table’s definitions will be based on the one Select statement.

The last type of UDF is the Multi-Statement Function. A Multi-Statement Function is similar to the Inline function but uses a Return syntax which explicitly specifies the structure of the returned table. By declaring a table variable, we can specify how the table result will be displayed: how many columns, their names, data types, and other returned variables. This also means that a Multi-Statement function is often slower in performance compared to the other two UDF’s.

# Summary

The UDF custom functions come in three different types in SQL. They are Scalar, Inline, and Multi-Statement functions, and they all provide unique purposes. A UDF is a useful tool to help tune reporting results as well as the performance of our database.