

Kristen Doehlert

May 31, 2022

IT FDN 130 A

Assignment 07

<https://github.com/kdoehlertUW/DBFoundations-Module07>

SQL Functions

Introduction

In module 7 we learned how to use SQL Functions. In this paper I am going to discuss when you would use a SQL User Defined Function (UDF). I will also cover the differences between Scalar, Inline, and Multi-Statement functions.

Using SQL UDFs

You can create custom functions, called UDFs, in addition to the stored functions already available in SQL Server. After creating the function once, it is stored in the database so you can call it any number of times in your code. You can use UDFs to return an individual value or a table. Like Views, UDFs are saved Select statements. However, parameters can be added to UDFs, which is useful when a value needs to be calculated. You can also use UDFs when creating Check constraints when you cannot otherwise reference a column in another table.

Scalar, Inline, and Multi-Statement Functions

A SQL function can either be Scalar, Inline, or Multi-Statement. Scalar functions return a single value, where Inline and Multi-Statement functions return a table. Inline functions contain a single select statement, and Multi-Statement functions contain multiple select statements. Figure 1 is an Inline UDF that returns a table of values that matches the KPI value that is entered in the function.

Figure 1. Inline UDF.

```
Create Function dbo.fProductInventoriesWithPreviousMonthCountsWithKPIs(@KPI Int)
Returns Table
AS
Return
    Select Top 10000000
        ProductName,
        InventoryDate,
        InventoryCount,
        PreviousMonthCount,
        CountVsPreviousCountKPI
    From vProductInventoriesWithPreviousMonthCountsWithKPIs
    Where CountVsPreviousCountKPI = @KPI
    Order by ProductName, Cast(InventoryDate As Date);
```

Summary

In this paper I discussed when you would use SQL UDFs. I also described the differences between Scalar, Inline, and Multi-Statement functions.