Julia Nagle

November 29, 2022

Foundations of Databases & SQL Programming

Assignment 07

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

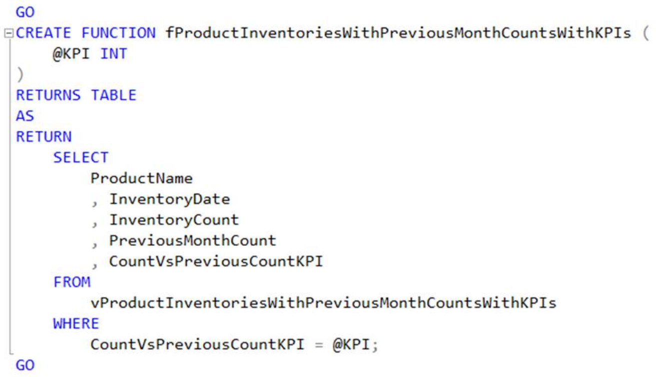
Functions

Introduction

This module focused on Functions. There are different types of functions with different purposes. Many of the functions that we have already used are System Functions – they are built into the SQL Server system and can't be modified. There are some Aggregate functions built in, such as COUNT, SUM, MAX, MIN, and AVG. There are some conversion functions, such as CAST and CONVERT, and many other useful functions which are part of the SQL system. We learned how to use a lot of these, but we also learned to create our own functions when there was no ready-made built-in Function to do the task needed.

When to Use a User Defined Function

User Defined Functions are very useful when you might want to reuse some code that can accept parameters. Views are also useful for re-use, but can't take parameters, so that is where UDFs come in handy. UDFs can either be Scalar (returning a single value) or Tabular (returning a table of values). They can take one or more parameters, and they can be used inside other SQL scripts, stored procedures, and functions.

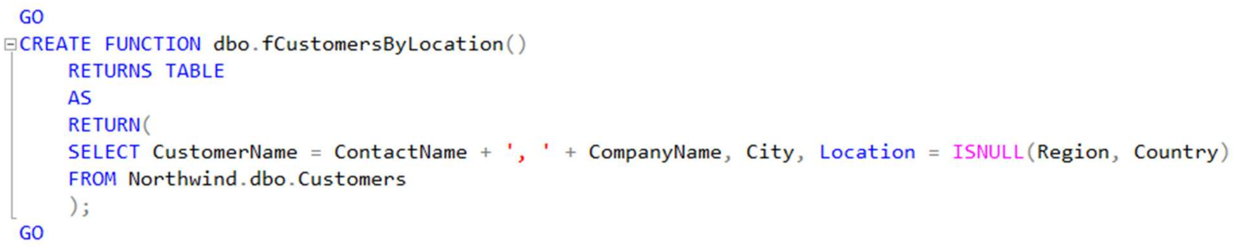


```
GO
CREATE FUNCTION fProductInventoriesWithPreviousMonthCountsWithKPIs (
    @KPI INT
)
RETURNS TABLE
AS
RETURN
    SELECT
        ProductName
        , InventoryDate
        , InventoryCount
        , PreviousMonthCount
        , CountVsPreviousCountKPI
    FROM
        vProductInventoriesWithPreviousMonthCountsWithKPIs
    WHERE
        CountVsPreviousCountKPI = @KPI;
GO
```

Figure 1. Creating a User Defined Tabular Function which takes a parameter, "@KPI"

Differences between Scalar, Inline, and Multi-Statement Functions

A scalar Function will return only a single value. You can use parameters in your function, but the result will still be a single value. Inline Tabular UDFs return a table, but they only use one Select statement. Again, they can take parameters as well, and they will return a table of results, not a single value.



```
GO
CREATE FUNCTION dbo.fCustomersByLocation()
    RETURNS TABLE
    AS
    RETURN(
    SELECT CustomerName = ContactName + ', ' + CompanyName, City, Location = ISNULL(Region, Country)
    FROM Northwind.dbo.Customers
    );
GO
```

Figure 1: Creating an Inline Tabular function with no parameters.

Multi-Statement Table Valued Functions return tables as well, but they are more complicated than the Inline functions. For these, you have to define the new table that your function is creating, and you must use the “BEGIN” and “END” code around your statements. Although parameters are not required, they’re unlikely to be missing from a MSTVF.

Summary

Functions range from the simple built-in system functions that do very common and useful things to Multi-Statement Table Valued User Defined functions that take multiple parameters. They are extremely useful for returning scalar values or tables using parameters, but unlike stored procedures they can’t actually modify data.