**Foundations of Databases and SQL Programming (IT FDN 130 A)**

**Author:** Vamsi Krishna Kancharla

**Assignment 07: Functions (UDF, Built-In, Scalar, Inline, Multi-Statement)**

**Submission Date:** 05/26/2021

GitHub URL - <https://github.com/linuxgeek99/DBFoundations-Module07>

**Introduction:**

The purpose of this paper is to discuss about Module 7 on functions - User Defined Functions (UDF), Scalar and Multi-Statement functions.

1. Explain when you would use a SQL UDF.

A SQL UDF evaluates an arbitrary SQL expression and returns the result(s) of the expression. Built-in functions do not always offer the desired functionality. To customize to our specific needs, we need to create our own functions that can be used at various places.

1. Explain are the differences between Scalar, Inline, and Multi-Statement Functions.

**Scalar Function:** SQL Server scalar function takes one or more parameters and returns a single value. Scalar functions are called like a built-in function. An example of inline function listed in Figure 2.0.

CREATE FUNCTION sales.udfNetSale(

@quantity INT,

@list\_price DEC(10,2),

@discount DEC(4,2)

)

RETURNS DEC(10,2)

AS

BEGIN

RETURN @quantity \* @list\_price \* (1 - @discount);

END;

Figure 2.0 – Example of Scalar Function

**Inline Function:** Inline functions accept parameters, and these parameters must be passed to the functions to execute them. The body of the function will have only a single select statement prepared with the “RETURN” statement, table returned by the inline table-valued function in SQL Server can also be used in joins with other tables. An example of inline function listed in Figure 2.1.

CREATE FUNCTION [dbo].[udfGetProductList]

(@SafetyStockLevel SMALLINT

)

RETURNS TABLE

AS

RETURN

(SELECT Product.ProductID,

Product.Name,

Product.ProductNumber

FROM Production.Product

WHERE SafetyStockLevel >= @SafetyStockLevel)

Figure 2.1 – Example of Inline Function

Multi-Statement Function: The table structure can be defined in multi-statement function that the function will return, it can have Begin and End blocks. RETURN statement must also be used to return the output table. An example of Multi-Statement is shown in Figure 2.2.

CREATE FUNCTION udfContacts()

RETURNS @contacts TABLE (

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(255),

phone VARCHAR(25),

contact\_type VARCHAR(20)

)

AS

BEGIN

INSERT INTO @contacts

SELECT

first\_name,

last\_name,

email,

phone,

'Staff'

FROM

sales.staffs;

INSERT INTO @contacts

SELECT

first\_name,

last\_name,

email,

phone,

'Customer'

FROM

sales.customers;

RETURN;

END;

Figure 2.2 – example of Multi-Statement function

**Conclusion:** In this paper we have learned various SQL Server functions and their purpose. Creating UDF are beneficial as it helps with don’t repeat yourself (DRY) principal, where code is written once and re-used every time which saves time, creates more efficiency and productivity.