

Dayana Stroshine

September 7, 2021

Foundations Of Databases & SQL Programming

Assignment 07

## SQL FUNCTIONS

### INTRODUCTION

Functions are very helpful resources in SQL. There are a variety of built-in functions that a developer can use such as aggregate functions (MAX, MIN, SUM, etc.) as well as DATE functions or NULL. If these functions aren't able to provide what the developer needs, there also exists the option to create UDFs or User Defined Functions. These functions are custom built by the user and can result in anything from a single result to a table of results. This brief summary will review these custom functions and provide example use cases of each.

### WHEN WOULD ONE USE A SQL UDF?

A User Defined Function is a function that is custom built by a user. These functions are used when a developer needs to perform a function that is not available in the built-in functions. There are two types of UDFs, functions that return a table of values(table-valued) and functions that return a single value (scalar).

### WHAT ARE THE DIFFERENCES AND SIMILARITIES BETWEEN A SCALAR, INLINE, AND MULTI-STATEMENT FUNCTION?

As mentioned above, all three of these functions are UDFs and can be adjusted by the developer as needed. A scalar function returns a single value of any data type. As seen in figure 1 below, scalar functions need BEGIN and END statements in order to run and they do not select FROM a specific table. This example returns a single value of 9.

```
-- create the function
CREATE FUNCTION dbo.fFunctionName(@value1 INT, @value2 INT)
RETURNS INT
AS
BEGIN
    RETURN(SELECT @value1 + @value2);
END
GO
-- call the function
SELECT dbo.fFunctionName(2,7)
```

Figure 1

An inline function and a multi-statement function are both considered table-valued functions, which means they return a table of values and both select FROM a table or tables. Inline functions and multi-statement functions differ in what they return. Inline functions return a single set of rows, while multi-select functions return values from multiple tables. In figure 2, we see an example of an inline function. Inline functions take a value input, similar to scalar functions, but return a table rather than a data type. The table consists of a SELECT statement that encompasses the declared value in its logic.

```
GO
CREATE FUNCTION fFunctionName
(
    @value MONEY
)
RETURNS TABLE
AS
RETURN
    SELECT
        ProductName
        ,UnitPrice
    FROM
        dbo.vProducts
    WHERE
        UnitPrice > @value
GO
-- call the function
SELECT * FROM fFunctionName(25)
```

	ProductName	UnitPrice
1	Uncle Bob's Organic Dried...	30.0000
2	Northwoods Cranberry Sauce	40.0000
3	Mishi Kobe Niku	97.0000
4	Ikura	31.0000

Figure 2

Lastly, multi-select statements are used when a developer wants to incorporate additional logic into a function and return a new table or value. In figure 3 below, the table @Threshold is returned with the values 'above' and 'below' in the Threshold field to identify the threshold on each row based on the unit price. Multi-select functions use the INSERT INTO statements to bring new fields based on logic using the declared value.

```

CREATE FUNCTION fFunctionName(
    @value INT
)
RETURNS @Threshold TABLE(
    ProductName VARCHAR(50)
    ,UnitPrice VARCHAR(50)
    ,Threshold VARCHAR(50)
)
AS
BEGIN
    INSERT INTO @Threshold (
        ProductName
        ,UnitPrice
        ,Threshold
    )
    SELECT
        ProductName
        ,UnitPrice
        , 'Above'
    FROM vProducts
    WHERE UnitPrice > @value
    INSERT INTO @Threshold (
        ProductName
        ,UnitPrice
        ,Threshold
    )
    SELECT
        ProductName
        ,UnitPrice
        , 'Below'
    FROM vProducts
    WHERE UnitPrice < @value
    RETURN
END
GO
-- calling the function
SELECT * FROM fFunctionName(25) order by productName

```

	ProductName	UnitPrice	Threshold
1	Alice Mutton	39.00	Above
2	Aniseed Syrup	10.00	Below
3	Boston Crab Meat	18.40	Below
4	Camembert Pierrot	34.00	Above
5	Carnarvon Tigers	62.50	Above
6	Chai	18.00	Below

Figure 3

## SUMMARY

In summary, functions are a helpful tool to use when needing to manipulate, dynamically filter, or change data. The built-in functions are great at helping to aggregate, standardize, and organize data and if but if they do not help to solve the immediate problem, using UDFs is also an excellent way to create custom functionality for all purposes.