

Assignment Documentation Writeup
Course: IT FDN 130 A

Document: Doug Dunbar Assignment07 Writeup
Revision Number: 00
Revision Date: 11/27/21
Document Author: Doug Dunbar

SQL Functions Why & When

Introduction:

In this essay I will be reviewing the types of functions available in SQL coding. Starting with UDF's otherwise known as User Defined Functions, then moving to Scalar and Inline Functions and finally covering the basics of MSTVF's better known as Multi-Statement Table-Valued Functions, one of the poorest named SQL functions.

Overview:

1. Explain when you would use a SQL UDF.
2. Explain are the differences between Scalar, Inline, and Multi-Statement Functions.

When to Use UDF's

User-defined functions allow programmers to create their own routines and procedures that the computer can follow; it is the basic building block of any program also very important for modularity and code reuse since a programmer could create a user-defined function which does a specific process and simply call it every time it is needed (1). User defined functions are highly specific but if designed well can be used across multiple databases. For our purposes in this class the UDF's we have seen are specific to the point of context. That is, without the context of the databases we have been working in, the UDF's probably wouldn't provide much insight.

Types of SQL Functions

Scalar Functions: A scalar function is a UDF that accepts any number of parameters and returns one value. The term scalar differentiates a single, "flat" value from more complex structured values, such as arrays or result sets (2). Simple examples of Scalar Functions include: LCASE(), UCASE(), MID(), ROUND() etc.

Inline Table-Valued Functions: This type of functions returns a result set, much like a view. However, unlike a view, functions can accept parameters this is a key functionality difference. The inline function's syntax is quite simple. In the function definition, the return type is set to a table. A return statement is used with a select query in parenthesis (2).

Multi-Statement Table-Valued Functions: Multi-Statement functions can be used to do some unique things outside the context of a standard SELECT statement. This type of function returns a table-type result set, but the table is explicitly constructed in script. This can be used to accomplish one of two things: either to process some unique business logic by assembling a virtual table on the fly, or to duplicate the functionality of an inline function in a more verbose and compiled way. In short, if you

need to select records from an existing a result set, use an inline table-valued function (2). The syntax of an MSTVF is as follows (3):

```
CREATE FUNCTION MultiStatement_TableValued_Function_Name
(
    @param1 DataType,
    @param2 DataType,
    @paramN DataType
)
RETURNS
@OutputTable TABLE
(
    @Column1 DataTypeForColumn1 ,
    @Column2 DataTypeForColumn2
)
AS
BEGIN
    --FunctionBody
RETURN
END
```

MSTVF's essentially take many questions regarding a database that are not readily selectable and outputs a new table of values to answer those questions. This is slightly different from Inlined and Scalar value functions in that they can produce many results to provide more context to the data.

Conclusions & Lessons Learned:

Functions are complex... Maybe it is the turkey brain from thanksgiving, but I will need to review these concepts and use them several more times before I understand what is going on.

Sources:

1. *Technopedia*, <https://www.techopedia.com/definition/3888/user-defined-function-udf> (External Site)
2. *Microsoft Social*, <https://social.msdn.microsoft.com/Forums/sqlserver/en-US/e8aaefdb-0269-437f-b352-f2682a2c8055/difference-between-scalar-inline-and-table-valued-functions?forum=transactsql> (External Site)
3. *SQL Shack*, <https://www.sqlshack.com/sql-server-multi-statement-table-valued-functions/> (External Site)
- 4.

<https://github.com/DougDunbar5/Module07->