Diana Wong

11/23/2021

Foundations Of Databases & SQL Programming

Assignment 07

https://github.com/dwong1uw/DBFoundations-Module07

Functions

Introduction

The purpose of this document is to learn about the SQL user defined function. We'll learn about what a UDF is, when are they good to use and the differences between scalar, inline and multi-statement functions.

SQL UDF

A user defined function (UDF) allows customization in functions to return values as a single value or as a table. This allows a user to define their own calculations and formatting. Once a function is created, it can be reused, which is very helpful when complex codes do not have to be re-written each time. Scalar, inline and multi-statement functions are a few versions of UDFs.

Scalar Functions

With scalar functions, it returns a single value where schema names are required. Parameters can also be used with scalar functions.

Inline Functions

The simplest UDF is said to be inline functions that is used as a table function and can return a single set of rows. Parameters in table functions are not as useful when it is used in scalar functions.

Multi-Statement Functions

Similar to inline functions that returns as a table function is multi-statement function. An added capability is that multiple select statements can be performed so various sources can be used to build a customized table. Figure 1 shows some similar constraints of both scalar functions and table-valued functions (https://www.wiseowl.co.uk/blog/s347/limitations.htm) (External Site).

Limitations of table-valued functions

I use table-valued functions throughout my systems, but it's as well to know of a few limitations they share with normal scalar functions. These are as follows:

Limitation	Notes
No "side-effects" allowed	Functions that you write can't insert, delete or update permanent tables.
Calling stored procedures is not possible	Functions can't call normal stored procedures (although they can call extended stored procedures and other functions).
Non-deterministic functions are barred	You can not use certain non-deterministic system functions such as **RAND** (a function to generate random numbers).
No temporary tables	You can not use temporary tables within a user-defined function.
Limited Error Trapping	You can not use **TRY** / **CATCH** blocks in user-defined functions.

Figure 1 - Table-Valued Functions

Summary

We reviewed the great benefit of allowing customization through user defined functions and examples of three different types. Whether we have an output of a single value or as a table of values, UDFs provides a user more capability and flexibility in calculations and formatting. While there are still limitations to each, it is still a good tool.