Douglas Neri

March 4, 2021

IT FDN 130 A Wi 21: Foundations of Databases & SQL Programming

Assignment -07

[dougn15/DBFoundations-Module07 (github.com)](https://github.com/dougn15/DBFoundations-Module07) (external link)

Functions

Introduction

This paper will introduce you to function in SQL to provide you with an overview of the general concept. Functions, which are also known as User Defined Functions (UDF) are routines that accept parameters, perform actions, and return the result of that action. The return value can either be scalar (single) or in table form and unlike some other routines in SQL, must have a return value. Functions as a whole are an integral part of using SQL and it will greatly increase the programs’ usability the more you are familiar with them.

Explain when you would use a UDF

The simplest explanation of when to use a UDF is when you would like to repeat a single action. If you would like to repeat multiple actions a Stored Procedure is a better routine (see Assignment06). UDF’s provide a mechanism for modular programming, faster execution, and a reduction in network traffic1. Modular programming is the technique of breaking up the program into smaller interchangeable ‘modules’ for easier adaptation. This not only makes it easier to work on a team, but it also helps with reusing functions amongst multiple SQL databases. The way in which function increase performance is by caching the code and reusing it. Within the context of this paragraph, you should lean into using functions when memory is a commodity or when speed is a primary driver of what you are assembling. The reduction in network traffic has a specific caveat to it, that you use the UDF in the WHERE clause. This reduces the number of rows transmitted and thus reduces traffic. As a note, it is also assumed that the information put into the WHERE clause is appropriate and not attempting to force the clause to do something it is not intended to do (like arithmetic).

Example UDF2

CREATE FUNCTION fnNameOfFunction (@param DATATYPE, @param DATATYPE, etc)

RETURNS datatype (table, int, etc)

AS (sql expression)

The differences between Scalar, Inline, and Multi-Statement functions.

A scalar function is a function that returns a single value. Inline and Multi-Statement functions are a description of how many statements the function has within it, inline being a single Select statement and a multi-statement being multiple Select statements. One caveat to all of this is that the return type cannot be text, ntext, image, cursor, or timestamp.

Conclusion

UDF’s and Scalar functions can be exceptionally helpful while querying a database. A few notes to point out is to not mix up a User Defined Function and a System Defined Function, and their sub-types. Each has a Scalar function with slightly different aspects to them, so be aware as you are navigating your query!

1 - <https://docs.microsoft.com/en-us/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-ver15> (external link)

2 - [Simple (in-line) table-valued functions (wiseowl.co.uk)](https://www.wiseowl.co.uk/blog/s347/in-line.htm)