

Anna McGrady

March 1, 2022

IT FDN 130 A Wi 22: Foundations of Databases & SQL Programming

Assignment 07: Functions

[GitHub Repository Link](#) [EXTERNAL WEBSITE]

Saving Time and Energy with SQL Functions

Introduction

As you're working in SQL, you may find that there are times where you have an action you want to execute multiple times, or perhaps save for later. Microsoft SQL Server allows you to encapsulate defined actions in Functions. Saving a segment of code as a user-defined function or utilizing Microsoft's set of system functions gives you the ability to run a pre-designed procedure according to any input values and receive an output as a scalar or tabular data.

Utilizing User-Defined Functions in SQL

Microsoft SQL Server has two types of functions: system functions and user-defined functions (UDFs). System functions are built into SQL Server and can perform standard tasks ranging from aggregation to identifying null values to concatenating strings of text. User-Defined Functions take the concept of a function (Defined Action Based on Parameters > Optional Parameters > Defined Scalar or Tabular Output) and set it as a flexible concept you can use whenever you want to execute an action multiple times.

Consider this example of when you might create a User-Defined Function: You have complex SQL statement that joins an Employee Info table, an Employee Location table, a Sales table, and a Manager Table to get the Monthly Sales Count by Employee with their Location and Manager name. If you need to run this query each month for each employee, you're going to get sick of writing all that SQL code. Instead of writing new code each time, you can save your joined SELECT statement as a function with the Employee ID as a parameter. Then whenever you need to see an employee's sales count, you just call your function and use their ID as the function's parameter.

The Different Types of SQL Functions

There are three primary types of SQL functions – Scalar, Inline, and Multi-Statement – which are differentiated based on what kind of data they return and how that data is structured within the function itself. These three types of functions can apply to a multitude of situations,

each focused on the core value add of optimizing the re-usability and customization of your SQL code.

Scalar Functions

Scalar functions return a single – scalar – value. While it only returns one value, scalar functions can have none to many elements as parameters. For scalar functions, the function is created by defining the type of data returned for the single value.

Inline Functions

Inline Functions (Inline Table Valued Functions or ITVFs) return tabular data and are created by returning the results of a single SELECT statement. This means that inline functions rely on the underlying structure of the tables or views called in the SELECT statement (so no actual tables are being created or defined within the function). As with scalar functions, inline functions can accept none to many elements as parameters.

Multi-Statement Functions

Multi-Statement Functions (Multi-Statement Table Valued Functions or MSTVFs) are like inline functions in that they return tabular data, but they differ in that you define and insert that data within the function. The multiple statements within MSTVFs refer to the first statement that builds and defines the returned table structure and the second statement that inserts the defined data into the table created above.

As MSTVFs include INSERT INTO statements, they must also include BEGIN and END clauses after the data insertion. One of the benefits of MSTVFs over ITVFs is that by building the table within the function, you can define data types and any constraints that may not be otherwise included in the underlying raw data or views, as well as perform any rollbacks or error messaging as part of your insertion batch.

Summary

The ability to functionalize your work means that you don't have to do the same work twice. Microsoft SQL Server offers multiple options for utilizing functions within your SQL code, allowing you to create actions that can be saved and applied again in the future. There are different types of functions, which are differentiated by their input and output. Scalar Functions return a single value based on 0 to many elements in the parameters. Inline Table Valued Functions return a table of values based on a SELECT statement and 0 to many elements in the parameters. Multi-Statement Table Valued Functions return a table of values that are built, defined, and populated within the statements of the function based on 0 to many elements in the parameters. By properly functionalizing your SQL code, you can future-proof your processes and save time.

Resources for Background Research

“Create User-Defined Functions (Database Engine).” *SQL Server / Microsoft Docs*, Microsoft, 22 Dec. 2020, <https://docs.microsoft.com/en-us/sql/relational-databases/user-defined-functions/create-user-defined-functions-database-engine?view=sql-server-ver15#TVF>. [EXTERNAL WEBSITE]

D., Josh. “Multi Statement Table Valued Function: The Ultimate Guide for Beginners.” *Simple SQL Tutorials*, 29 Oct. 2021, <https://simplesqltutorials.com/multi-statement-table-valued-functions-guide-for-beginners/>. [EXTERNAL WEBSITE]

Drkusic, Emil. “User-Defined Functions.” *SQL Shack*, 17 May 2021, <https://www.sqlshack.com/learn-sql-user-defined-functions/>. [EXTERNAL WEBSITE]

Erkec, Esat. “SQL Server Multi-Statement Table-Valued Functions.” *SQL Shack*, 11 Sept. 2019, <https://www.sqlshack.com/sql-server-multi-statement-table-valued-functions/>. [EXTERNAL WEBSITE]

Root, Randal. “Module 07 – Functions.” Randal Root, 22 Feb. 2022.

“User-Defined Functions.” *SQL Server / Microsoft Docs*, Microsoft, 14 Dec. 2020, <https://docs.microsoft.com/en-us/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-ver15>. [EXTERNAL WEBSITE]