

Customizing Output: An Overview of User-Defined Functions

Objective

Write a one-page document answering the following prompts about user-defined functions.

Explain when a SQL UDF would be used.

A user-defined function (or UDF) in SQL is just that - a user-generated block of code which can be stored and subsequently recalled and reused across the database. Without going too far into their precise construction, from a high level UDFs have a number of applications across industries and sectors. Say, for example, we are a data analytics team tasked with generating and maintaining issue resolution metrics for a large organization's IT department. Instead of manually accessing and generating reports for each issue or service area on a week-by-week and month-by-month basis, we could easily create and employ an array of functions capable of tabulating our metrics *ad hoc* - giving us valuable time to gather insights from, rather than endlessly calculate, our department's performance. In doing so, not only will our metrics be standardized; we will also be able to assign referential values to and otherwise perform calculations on the results of those functions as well.

Explain the differences between Scalar, Inline, and Multi-Statement Functions.

There are several types of UDF, all of which serve the same ultimate purpose of simplifying and recycling calculations and other actions performed on the database. A scalar function, for example, is a UDF that returns a single value each time it is invoked. Though scalar functions are limited in output, they are powerful. Imagine a cashier having to manually total each sale at their register and apply a corresponding tax: each transaction would, in theory, take several minutes to complete. However, by storing that calculation and recalling it repetitively, receipts are totaled - with tax - almost instantaneously, with no variation in their result. These predictable, working calculations in practice have very wide applications.

That said, not every task merits a simple solution. As opposed to scalar functions, some UDFs are able to return a table of data - collectively, these are called table-valued functions (TVF). Two of the most common TVFs are inline and multi-statement functions. In the case of inline TVFs, the values returned are defined through a single SELECT statement. While said SELECT statement can be relatively complex, by nature an inline function is doing just that - selecting and returning values - and therefore cannot account for variables. In this way inline TVFs are very similar to views. Due to the precise manner in which they are executed, inline functions are, from a performance standpoint, quite efficient as well.

Multi-statement functions, however, are much closer in execution to a stored procedure. Unlike inline functions, MSTVFs allow the user to input variables that will alter the resulting table. Depending on the needs of the user, this may be ideal; instead of returning a static set of results, MSTVFs empower the user to store and recall complex calculations which can then be further defined and refined at the time of use, rendering them much more flexible and customizable than a standard inline or scalar function.

Resources

Microsoft, 2016, *User-Defined Functions*, accessed 22 August 2021, <<https://docs.microsoft.com/en-us/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-ver15>>.

GitHub Documentation

<https://github.com/emleggett/DBFoundations-Module07>