Nate Kopp
November 26th, 2025
IT FDN 130A Au 25
Assignment 07

SQL Functions

Intro

User-defined functions (UDFs) are routines that accept parameters and return a value or table based on inputs. They are useful tools to work efficiently by creating your own functions and avoid rewriting complicated code. This paper will explain the three main types of UDFs and why you would use them.

When Would You Use a SQL UDF?

You would use a UDF when you want to simplify your queries by condensing repeated or complex logic into a reusable function. UDFs help keep SQL cleaner and more consistent by centralizing calculations, business rules, or formatting into one place.

Comparing a Scalar, Inline and Multi-Statement Functions

Scalar, Inline, and Multi-Statement Functions all have the same core purpose of letting you quickly reuse logic without repeating multiple lines of code. They all use CREATE FUNCTION, however how they are structured and what they return differ.

- Scalar Functions – Returns a single value, a number or string, based on the inputs you define. Scalar functions are often used for repeated queries or to simplify complex calculations. They use BEGIN/END and are run row by row, which make them the least performance efficient of the three.
- Inline Table-Valued Functions (ITVF) – Much like scalar functions, ITVFs are a SQL function where you pass inputs to return query. However, instead of returning a single value, an ITVF returns a table. ITVFs are used to produce a parameterized view with reusable table logic. They are based on a single SELECT statement and are optimized to be the most performance efficient of the three.
- Multi-Statement Table-Valued Functions (MTVF) – Similar to ITVFs, MTVFs return a table, however they are built across multiple SELECT statements. They allow for more complex, multi-step, logic using INSERTs, UPDATEs, and conditional logic. Due to this, they are slower than ITVFs, but more efficient than scalar functions as not run row by row.

Summary

UDFs help make queries cleaner, reduce duplication, and ensure consistent results across reports and applications. UDFs come in different types, Scalar, Inline, and Multi-Statement, each designed for specific needs, such as returning a single value or producing a table of results. Overall, UDFs improve readability, maintainability, and reliability in SQL development.