

Madeleine McDonald
August 10, 2024
IT FDN 130
Assignment 7: Functions
<https://github.com/MMcDonald04/DBFoundations-Module7>

Introduction

Using relational databases to extract data based on specific requirements is easy. However, performing a repetitive task could become burdensome if one had to copy and paste code over and over for similar parameters. Using functions streamlines this process by cutting down any redundant code. Besides the default functions available to SQL, there are user defined functions (UDFs) that can be scalar, inline or multi-statement.

UDF's

If there is a custom query that needs to be ran multiple times for a set of parameters, then UDFs are the preferred method of execution. A simple example of when a UDF might be used is for a business determining sales price based on custom discounts for each product. If columns "unit price" and "discount" exists in a table, the default method of finding the new "sales price" might be to multiply one unit price by its custom discount. However, creating a UDF with variables for "unit price" and "discount" means the code to perform the calculation can be reused and saved as a function. One could also apply this function to a new row in the table to compare "unit price" with the new "sales price".

Scalar, Inline and Multi-Statement Functions

The three types of UDFs are scalar, inline and multi-statement. Each type can handle progressively more complex demands. Scalar returns a single value and inline returns a table. Scalar uses begin/end block in the Return section of function, while inline uses the common syntax of a select statement. Multi-statement includes both aspects of scalar and inline, begin/end block and select statement syntax.

Conclusion

Performing queries back-to-back for various parameters is made simpler through the use of functions. UDFs provide a more customizable approach to this method, by allowing the database user to use scalar, inline or multi-statement functions. Utilizing these features increases the value in storing data in relational tables.