Joanne Lee

Nov. 27, 2022

DB Foundation

Assignment 07

<https://github.com/Hsinjou99/DBFoundations-Module07>

Functions

# Introduction

This week I learned about various SQL functions, what they do, and how to use them. I will be going over the User Defined Functions (UDF) and when you would use them.

# User Defined Functions

UDFs are custom functions that get stored inside of a database. You can define these functions to return a single value or even a whole table. UDFs are useful because you can use it to store code and call it instead of writing more code to get the same result. Unlike views, you can set parameters in functions, and this allows you to change the output of the function depending on what you put in the input.

# Scalar Functions

Scalar functions only return one value even when given any number of parameters. When creating a scalar function, the schema name must be included in the name of the function, and you must specify which variable will be returned.

# Inline Functions

Inline functions store a SELECT statement that returns a table. You can include parameters in these functions but when called, will still return a table just like a view or SELECT statement will. Inline function parameters can be used to give the same results as when using the WHERE clause in a SELECT statement.

# Multi-Statement Functions

Multi-statement functions also returns a table but are slightly more complicated. Unlike inline functions, they require you to declare the columns of the table that will be returned. It also requires BEGIN and END to insert values into the table. You can use more than one SELECT statements in multi-statement functions unlike with inline functions where you can only include one.

# Summary

UDFs allow you to create and store your own functions with its own use so that you don’t have to rewrite SELECT statements in the future. There are several types of UDFs that you can make, including scalar, inline, and multi-statement functions. Each have their own properties and can be used in different situations.