Name: Alex Pirio
Date: Mar 1, 2022
Course: IT FDN 130

Assignment #7-Foundation Assignment

Introduction

UDFs are very important to all programming languages and SQL is no different. UDFs allow the user to create a function that they can then recall later to minimize the amount of rewritten code in the program. UDFs can come in Scalar, Inline and Multi-Statement forms. They vary in terms of their parameters and what they return but all of them streamline the process and allow cleaner mor concise code.

Topic: Explain when you would use a SQL UDF:

SQL programmers use UDFs to allow for modular programming. You can create a function once and then store it for later use in your program. This allows the use to call the specified UDF instead of rewriting the functions code. This is much more efficient and leads to cleaner code.

Topic: Explain what are the differences between Scalar, Inline and Multi-Statement Functions:

All of them are UDFs, but they vary on what they return and what parameters they take. Scalar functions return one value, and they accept a wide variety of parameters. Since I have a background in programming this is the type of function in SQL, I'm most familiar with. Inline Functions return a table and unlike a view they can take parameters because they are functions. Multi-Statement Functions can do more than a standalone SELECT statement can. This function type returns a table, but it creates a virtual version of it. Multi-Statement Functions can also increase the functionality of the other two UDFs because they can add more functions to them creating a complete package.

Summary

UDFs are important for a SQL programmer to master because they will make their lives to much easier. UDFs allow reuse of a function without writing the same chunk of code again and again for multiple calls of it. This can make sharing code amongst colleagues easier, and it can help you understand your code better. It also minimizes the mistakes when you only write the function once, then all your errors with to function (assuming it works right) are most likely due to improperly used parameters.