Michaela Orr
11/30/2022
Foundations of Databases and SQL Programming
Assignment 07
GitHub Repository: https://github.com/morr3/DBFoundations-Module07

Introduction

This writeup will cover different types of SQL user-defined functions and some of their use cases.

When to use a SQL UDF

User-defined functions are another type of abstraction layer for working with a database. Unlike views however, they allow a user to drive the results of the query by inputting different parameters into the function.

There are plenty of built-in functions associated with every SQL syntax, but UDFs allow an end user to create more custom functions suited to their specific workflows. They are particularly useful in cases when a user would otherwise need to perform the same type of complicated calculation repeatedly - baking that into a UDF can make code more legible and save users time moving forward.

Different Types of Functions

- Scalar functions return only one value, and the data type for that value must be specified in RETURNS clause. They can be used within a SELECT statement, which can be particularly useful for applying the function to every row with a results set.
- In-line functions are a type of tabular function, meaning they return a table of results that can be called like any other table/view once the function is created. A simple in-line function only contains one SELECT statement within the RETURN clause, which can be somewhat limited when additional processing is needed.
- Multi-statement functions are another type of tabular function. Unlike simple in-line functions, they can add layers of additional processing in the form of multiple INSERT/SELECT statements. Once created, the results of these functions can also be called like a table or view.

Summary

Though there are a wide variety of useful system-defined functions available, UDFs come in handy when a user needs to create a more bespoke function for future reference.