Phuong Tran (Eric)

IT FDN 130 A

03/11/2025

Assignment 07

GitHub Repository: https://github.com/erictran03/DBFoundations-Module07

Understanding SQL User-Defined Functions

Introduction

In Assignment 07, I explored SQL User-Defined Functions (UDFs), an essential feature in SQL that allows users to encapsulate reusable logic into functions. Functions improve query efficiency, maintainability, and readability by abstracting repeated logic into modular components. This assignment involved reading about SQL functions, implementing them in a database, and answering specific business-related questions using SQL queries. The key focus was on understanding scalar, inline table-valued, and multi-statement table-valued functions while utilizing them in practical database scenarios.

When to Use a SQL UDF

SQL User-Defined Functions (UDFs) are used when we need to encapsulate logic that can be reused across multiple queries while improving maintainability. UDFs help simplify complex calculations and ensure consistency by standardizing computations in a central function rather than repeating the logic in multiple queries. They are particularly beneficial when performing transformations, aggregations, and formatting operations that require repeated application across datasets. For example, a business may use a UDF to format monetary values consistently or calculate inventory trends over time. Since UDFs do not allow direct modification of database data, they are generally safer than stored procedures when used in a read-only context.

Differences Between Scalar, Inline, and Multi-Statement Functions

SQL UDFs are categorized into three main types, each serving distinct purposes:

- Scalar Functions: These functions return a single value based on the input parameters. They are commonly used for data transformation, such as formatting strings or performing mathematical calculations. An example is a function that converts a price into a formatted currency string.
- Inline Table-Valued Functions (TVFs): These functions return a table result set and are defined using a single SELECT statement. They are often used to simplify queries by encapsulating complex joins or aggregations. Unlike scalar functions, inline TVFs can be directly referenced in a FROM clause, improving query performance.

- Multi-Statement Table-Valued Functions: These functions return a table but allow for multiple SQL statements to be executed within the function body. They are more flexible than inline TVFs but may have performance implications since they use table variables internally.

Summary

This assignment enhanced my understanding of SQL UDFs and their role in improving database efficiency. UDFs provide modularity, consistency, and reusability, making them valuable tools in SQL development. Additionally, understanding the differences between scalar, inline table-valued, and multi-statement table-valued functions has provided me with better insight into when to use each type for optimal database performance. Overall, this knowledge strengthens my ability to design efficient and maintainable SQL queries in real-world applications.