Name: Caroline Truong

Date: March 1, 2023

Course: IT FDN 130 – Foundations of Databases and SQL Programming

Assignment: Knowledge Document for Assignment 7

Github URL: https://github.com/caroltru/DBFoundations-Module07.git

Introduction: The writeup topics for Assignment 7 request explanations for (a) when one would use a SQL UDF, or user-defined function; (b) the differences and similarities between a Scalar, Inline, and Multi-Start Functions.

Topic (a): UDF's are often considered "routines" in SQL. They generally provide efficiency to perform specific tasks like complex calculations and data processing. They can be relied on for faster execution, modular programming and network traffic reduction. While using UDF's, faster execution means that one can repeat the same procedure with efficiency and speed. Like stored procedures, Transact -SQL UDF's reduce the compilation cost of Transact-SQL code by caching the plans and recycling them for repeated executions. This means the UDF does not need to be reoptimized with each use resulting in much faster execution times. Modular programming means one can construct UDF and then save it; can store function in the database and call it in any program as often as desired with comfort that the source code will not be affected. Network traffic reduction means that at the time of data transmission to the use, complex constraints can be expressed by UDF's. This way the amount of content displayed or sent to the user is controlled. An operation that filters data based on some complex constraint that can't be expressed in a single scalar expression can be expressed as a function. The function can then be invoked in the WHERE clause to reduce the number of rows sent to the user. UDF's also have the following limitations: (a) functions that are written cannot insert, delete or update permanent tables; (b) functions cannot call normal stored procedures although they can call extended stored procedures and other functions; (c) one cannot use temporary tables within a UDF; (d) one cannot use TRY/CATCH blocks in UDF's.

Topic (b): Scalar functions return a single value of a data type and are defined in the return clause. They are comprised of inline and multi-statement functions. An inline scalar function indicates that the returned scalar value is a result of a single statement. A multi-statement scalar function indicates that the returned scalar value is a combination of a series of statements. The function body can contain a series of Transact-SQL statements that return the single value.

Summary:

Assignment 7 questions relate to when a SQL UDF would be utilized as well as the primary differences and similarities of the scalar, inline and multi-start functions.