Anh Phan

May 25, 2021

Foundations of Databases & SQL Programming

Assignment07

<https://github.com/anhphan1101/DBFoundations>

Module 7: Assignment

# Introduction

Through out the course of the 7th module, I learned the basic concepts and practical uses of User Defined Functions (UDF) available in the SQL. This function addresses the similarities and differences between Scalar, Inline and Multi-statement Functions. Please find the detailed examples in the SQL file.

# Content

1. Explain when you would use a SQL View. Explain when you would use a SQL UDF (User Defined Functions)

I would use SQL View when I need to show certain tables to other teams without compromise the showing all the tables that currently store in the current department.

Here is why I would use a SQL UDF. A lot of shops will like to use UDFs to hid code complexity and make their queries look prettier and the scalar part means that one value is returned. The main benefits of UDC including supporting modular programming, reduce the completion cost of T-SQL code, and reduce the network traffic.

1. Explain are the differences between Scalar, Inline, and Multi-statement Functions.

A scalar function is to return a single value. For example, a function that bring back an aggregate either sum of items on the previous month, or to check inventory level in the store db.

An inline function also returns a single select statement in the form of table function. If a table value function could be said to work like a stored procedure, a line function is similar to a view. This means that a inline function can only contain a single Select statement and the columns in the Select statement implicitly define the columns of the returned table set of the function.

Multi-statement Functions returns a table as output and this output table structure can be defined by the user. A multi-statement function contains multiple SQL statement enclosed in Begin-end blocks. For example, a production engineers want to analyze product scrap quantities and they determine a critical value in order to degrade the cost of the products. If the scrap quantities exceed this value, the query has to indicate these product’s scrap status deems as critical. So to do that we normally use multi-statement tables to declares a parameter call @ScrapComRatio. This parameter will help us to determine the crab level of material and whether or not it would deem as critical or regular. The @Resultbl would store the values like ProductName. ScrapQty, etc… which would give us some description about the part/ product.

# Summary

Overall, I really enjoy functions and would explore more about it. It does optimize my SQL script performance.