Zachary Larmer
IT FDN 130
11/26/2024
Assignment 07
https://github.com/ZachLarmer/DBFoundations/tree/main

SQL Functions

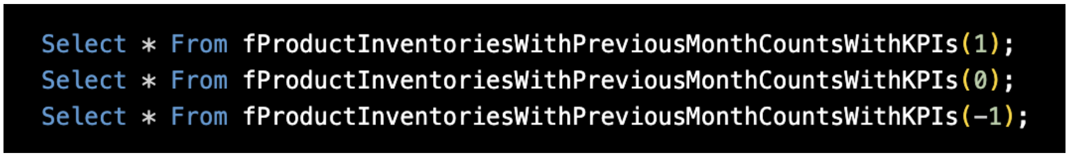
Introduction

Functions in SQL allow complex code to be stored in reusable easily accessible pieces of code. Each SQL program has many built in functions to assist the user. Aggregate functions being the most used of these help transform data by performing mathematical computations on selected data. Built in to SQL is also the ability for a user to create their own custom functions. These functions are a useful way to easily access data without writing out all of the code needed each time. Functions can be split into three categories, scalar, inline, or multi statement.

Using User Defined Functions (UDFs)

User defined functions or UDFs are exactly what they sound like. They are functions created by a user to assist in streamlining the usage of the database. UDFs are typically used when a specific query is used regularly and is either too complex or tedious to want to recreate every time it is used. By creating a UDF much like views or procedures, code can be called or selected quickly. This creates a useful level of abstraction that simplifies database usage.

Further, functions have the added utility of being able to pass parameters through to the underlying code. Adding parameters to a function allows multiple permutations of queries to be run quickly without having to copy and paste, and edit full SQL scripts (see figure 1). UDFs can then be used in other queries to perform complex actions, such as comparing parameters as a check constraint.



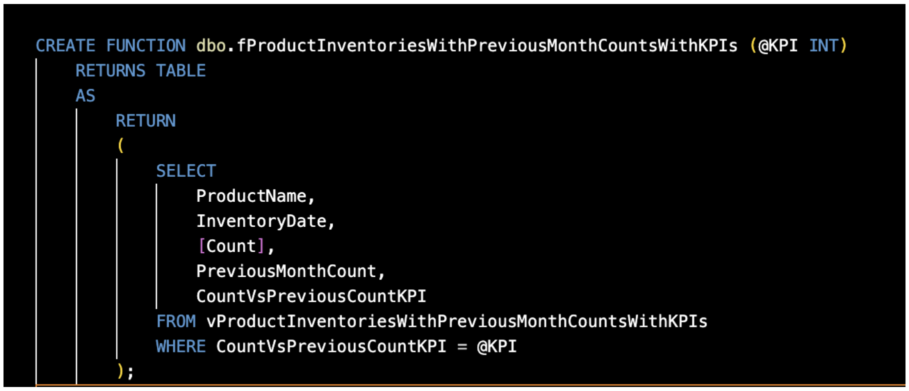
```
Select * From fProductInventoriesWithPreviousMonthCountsWithKPIs(1);
Select * From fProductInventoriesWithPreviousMonthCountsWithKPIs(0);
Select * From fProductInventoriesWithPreviousMonthCountsWithKPIs(-1);
```

Figure 1: Selecting a user defined function with parmeters

Scalar, Inline, & Multi Statement Functions

Scalar functions in their simplest form are functions that produce a single value when selected. These user defined functions output a single value. They can include parameters to complete actions similar to that of aggregate functions. Scalar functions are often used in a select statement to create a column in the output of a query showing additional information.

Inline functions output a table similar to view. Also known as a table-valued function, these functions encapsulate a select statement within the function for easy access. They often also include a parameter to help filter data as needed. The parameter is quite often used to alter the where clause to help filter data as needed (See figure 2).



```
CREATE FUNCTION dbo.fProductInventoriesWithPreviousMonthCountsWithKPIs (@KPI INT)
    RETURNS TABLE
    AS
        RETURN
        (
            SELECT
                ProductName,
                InventoryDate,
                [Count],
                PreviousMonthCount,
                CountVsPreviousCountKPI
            FROM vProductInventoriesWithPreviousMonthCountsWithKPIs
            WHERE CountVsPreviousCountKPI = @KPI
        );
```

Figure 2: Inline function with parameter in the where clause

Multi statement functions are the most complex of the UDFs. These functions return a table with a structure defined by the functions creator. Then within the function an query is written that will then insert selected values into the user defined table the function will output. These functions can contain very complex queries, and even multiple queries within the same function. Like the other types of UDFs, multi statement functions provide a useful level of abstractions.

Summary

Functions provide powerful tools for manipulating data within a table. They help aggregate and format data as needed for reporting and analyzing data. They also allow complex code to be saved and reused. Like views and stored procedures, functions help ease the use of a database and provide layers of abstraction to simplify complex uses of SQL queries.