Jonathan Owen
May 31, 2022
IT FDN 130 A Sp 22
Assignment 07
https://github.com/jono3028/DBFoundations-Module07

Introduction

A built-in system function allows a database user to perform common tasks that could potentially be useful in any database. However, when data within a specific database must be manipulated in a bespoke way, potentially repetitively, a User Defined Function (UDF) may be created to simplify the action. This paper will discuss the UDFs and the different types that may be used.

Using a SQL UDF

User-Defined Functions are reusable pieces of code that are stored within a database. There are multiple types of UDFs that are primarily determined by the type of object returned; these types will be discussed later in this paper. UDFs are an excellent choice when a calculation or complex query needs to be performed. A benefit of UDFs is that they accept input parameters, allowing the function to return dynamic results. Within a database, a UDF allows for the reduction of duplicate code and an increase in re-usability.

Types of Function

A UDF may be described as being a specific type of function; these will be outlined here.

Scalar

Scalar functions may accept zero to many input parameters but only return a single value. This form of UDF may be used to perform customized calculations or provide custom formatting of a column's value. A simple example would be a function to find the difference between two integers:

```
CREATE FUNCTION dbo.fnIntegerDiff(@value1 int, @value2 int)
RETURNS int
AS
    RETURN @value1 - @value2;
```

Inline Tabular

An inline UDF is a function that returns a table of values but contains only one select statement. A function like this may come in handy if there is a commonly used complex query with dynamic input parameters. An example of an inline UDF would be if a user had a table of users and a common query is to retrieve N users from a specific state.

```
CREATE FUNCTION dbo.fnGetUsersByState(@state nvarchar(2), @userCount int)
RETURNS table
AS
    RETURN(
        SELECT * FROM users WHERE state = @state LIMIT @userCount;
    )

-- To retreive the first 10 users from Washington
SELECT * FROM dbo.fnGetUsersByState('WA', 10);
```

Multi-Statement Tabular

Multi-statement UDFs are very much like an inline UDF, except that they contain multiple select statements. The statements occurring later in the function may be dependent on earlier ones. This form of UDF allows for some of the most complex functions to be created. An example of a multi-statement function would be taking data from two unrelated tables and returning them within one table. A function like this could be created by using a temporary table and then selecting the values from the desired tables before inserting the selected values into the selected table. The temporary table would then be returned.

Summary

User-Defined Functions are flexible tools within a database and allow for the simplification of complex processes into an easy-to-understand command. UDFs also allow for the reduction of duplicate code and improved database code management through the ability to store frequently used snippets of code in a reusable package that may be shared throughout a database.