

Emi Okikawa
May 28, 2024
IT FND 130 A Sp 24
Assignment 07
<https://github.com/eokikawa/DBFoundations-Module07>

Understanding SQL User Defined Functions

Introduction

This paper examines SQL User Defined Functions (UDFs) and their pivotal role in database queries. Beyond standard built-in functions, UDFs allow users to tailor custom functions, catering to specific reporting requirements. The paper explores three types of UDFs: Scalar, Inline, and Multi-Statement.

Explain when you would use a SQL UDF.

In addition to built-in functions, you can also create custom functions to build the report you want. These are called User Defined Functions (UDFs), and there are two basic types: functions that return a table of values and functions that return a single value. You can also use UDFs, specifically scalar functions, as check constraints to be able to reference a column in another table.

UDFs are powerful tools for streamlining the functionality of code within a database. Since they accept parameters, they allow users to swap values into the function for quicker processing. They also support various data types and can encapsulate a set of SQL statements, making them versatile for data processing. UDFs provide an efficient means for promoting code reuse and simplifying complex operations without having to type out lengthy or repetitive queries.

Explain the differences between Scalar, Inline, and Multi-Statement Functions.

Scalar: A Scalar Function returns a single value based on the input parameters. It is commonly used for calculations, string manipulations, or date operations. Scalar functions return a specific value, and use a Begin/End block.

```
CREATE FUNCTION dbo.AddTwoNumbers(@num1 INT, @num2 INT)
RETURNS INT
AS
BEGIN
    RETURN @num1 + @num2;
END;
```

Inline: An In-line (single statement) table function returns a table of data. Unlike Scalar and Multi-Statement functions, it does not use a Begin/End block.

```
CREATE FUNCTION dbo.ExampleFunction(@num1 INT)
RETURNS TABLE
AS
RETURN (
    SELECT ExampleID, ExampleName
    FROM ExampleDatabase
    WHERE NumberID = @num1
);
```

Multi-Statement: A Multiple-Statement function also returns a table of data, but the query can contain many complex SELECT statements. These functions differ from inline table-valued functions by declaring the return table structure, using a Begin/End block, using a Return operator, and can involve multiple statements.

```
CREATE FUNCTION MultiStatement_TableValued_Function_Name
(
    @param1 DataType,
    @param2 DataType,
    @paramN DataType
)
RETURNS
@OutputTable TABLE
(
    @Column1 DataTypeForColumn1 ,
    @Column2 DataTypeForColumn2
)
AS
BEGIN
    --FunctionBody
    RETURN
END
```

Summary

SQL User Defined Functions (UDFs) offer a flexible approach to enhancing the functionality of database queries beyond built-in functions. UDFs in SQL serve as tools for optimizing code efficiency within databases, accommodating the use of parameters in data processing and supporting various data types. Scalar functions perform single-value computations and manipulations, while inline functions facilitate the retrieval of tabular data. Multi-statement functions offer versatility for intricate data operations, enabling the execution of multiple SQL statements within a defined function scope, contributing to enhanced code organization and reusability.