

SQL Functions

Introduction

Functions are a named collection of SQL programming code. All RDMS include built-in functions, and some even let you create your own. These are often called User Defined Functions or just UDFs.

When you would use a SQL UDF

In addition to SQL Server's built-in functions, you can create custom functions. These are often called User Defined Functions or just UDFs. There are two basic types of functions: functions that return a table of values and functions that return a single value.

- Modular programming.

You can create the function once, store it in the database, and call it any number of times in your program. User-defined functions can be modified independently of the program source code.

- Faster execution.

Similar to stored procedures, Transact-SQL user-defined functions reduce the compilation cost of Transact-SQL code by caching the plans and reusing them for repeated executions. This means the user-defined function doesn't need to be reparsed and reoptimized with each use resulting in much faster execution times.

CLR functions offer significant performance advantages over Transact-SQL functions for computational tasks, string manipulation, and business logic. Transact-SQL functions are better suited for data-access intensive logic.

- Reduce network traffic.

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 client.

The differences between Scalar, Inline, and Multi-Statement Functions

- Scalar Functions

A scalar function returns a single value of the data type referenced in the RETURNS clause of the CREATE FUNCTION statement. The returned data can be of any type except text, ntext, image, cursor, or timestamp.

- **Inline Table-Valued Functions**

An inline table-valued function returns a variable of data type table whose value is derived from a single SELECT statement. Since the return value is derived from the SELECT statement, there is no BEGIN/END block needed in the CREATE FUNCTION statement. There is also no need to specify the table variable name (or column definitions for the table variable) because the structure of the returned value is generated from the columns that compose the SELECT statement. Because the results are a function of the columns referenced in the SELECT, no duplicate column names are allowed and all derived columns must have an associated alias.

- **Multi-Statement Table-Valued Functions**

The multi-statement table-valued function is slightly more complicated than the other two types of functions because it uses multiple statements to build the table that is returned to the calling statement. Unlike the inline table-valued function, a table variable must be explicitly declared and defined.

Summary

There are three types of user-defined functions in SQL Server:

1. Scalar Functions (Returns A Single Value)
2. Inline Table Valued Functions (Contains a single TSQL statement and returns a Table Set)
3. Multi-Statement Table Valued Functions (Contains multiple TSQL statements and returns Table Set)