Julia York
23-NOV-2025
Foundations of Databases and SQL Programming
Assignment 07
https://github.com/juliayork2/DBFoundations/tree/DBFoundations-Module07

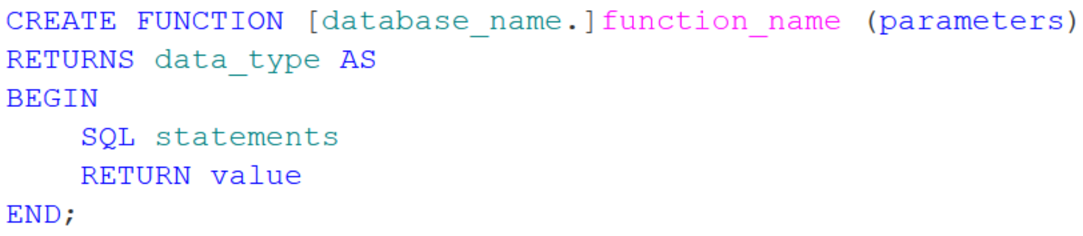
User-Defined Functions in SQL

Introduction

Functions are used in SQL to compute and manipulate values using parameters. Functions include built-in functions within SQL Server, such as sum and average, and user-defined functions which are created by the user. Functions can be written to return results as a table or to return a single (or scalar) value. When returning results in a table, the function may be written as inline or as a multi-statement function.

User-Defined Functions

Although SQL contains many built-in functions, there are times when creating a custom function is beneficial. User-defined functions (UDFs) are custom functions created by the user. UDFs are useful because they can be formatted to your exact specifications and then reused as many times as needed. Having the ability to save multiple lines of code into a single UDF saves time when writing code and reduces coding errors. UDFs are created using the CREATE FUNCTION statement (Figure 1). Parameters are defined within the CREATE FUNCTION statement. The results can be returned as either a scalar value or a table. The result type is specified using the RETURNS keyword. For scalar UDFs, a BEGIN…END block is used to specify the code that will be carried out by the function in conjunction with the RETURN keyword to specify what the function will return. After creating a UDF, it can be used in queries like a built-in function. However, the schema name must be included when calling UDFs in SQL.



```
CREATE FUNCTION [database_name.]function_name (parameters)
RETURNS data_type AS
BEGIN
    SQL statements
    RETURN value
END;
```

Figure 1. SQL syntax used to create a scalar user-defined function (SQLShack, https://www.sqlshack.com/learn-sql-user-defined-functions/, 2025).

Differences Between Scalar, Inline, and Multi-Statement Functions

Scalar functions are functions that return a single value. For example, the AVG() function is a scalar function that returns the average of the specified values. There can only be one average for each set of values used in the function, so only one value will be returned. In contrast, table-valued functions return a table of values. Table-valued functions can be inline table-valued functions or multi-statement table-valued functions. An inline table-valued function is defined through a single SELECT statement, whereas a multi-statement table-valued function (MSTVF) contains one or more table variables (@return_variable) used to store and accumulate the results that will be returned (Microsoft Learn, https://learn.microsoft.com/en-us/sql/t-sql/statements/create-function-transact-sql?view=sql-server-ver17, 2025). Inline table-valued functions are used when the desired results can be obtained using a single SELECT statement without additional processing. MSTVFs, on the other hand, are used when multiple steps are needed to determine the final output. As such, the results table of an MSTVF may contain aggregate data.

The syntax used to create inline table-valued functions is similar to that of scalar UDFs. However, there is no BEGIN…END block, and a single SELECT statement is used to specify the columns to be returned in the results table (Figure 2).

```
CREATE FUNCTION [schema_name. ] function_name (@parameter_name data_type)
RETURNS table
AS
   RETURN
      (SELECT statement);
```

Figure 2. Basic syntax used to create inline table-valued functions (Adapted from Microsoft Learn, https://learn.microsoft.com/en-us/sql/t-sql/statements/create-function-transact-sql?view=sql-server-ver17, 2025).

The syntax for MSTVFs is also similar to that of scalar UDFs, but MSTVF syntax includes one or more table variables (Figure 3). These variables are calculated as specified in the function body and reported in the results table. The structure of the results table may be defined by the user (SQLShack, https://www.sqlshack.com/sql-server-multi-statement-table-valued-functions/, 2025).

```
CREATE FUNCTION [schema_name. ] function_name (@parameter_name data_type)
RETURNS @return_variable table
AS
   BEGIN
      function_body
    RETURN
   END;
```

Figure 3. Basic syntax used to create multi-statement table-valued functions (Adapted from Microsoft Learn, https://learn.microsoft.com/en-us/sql/t-sql/statements/create-function-transact-sql?view=sql-server-ver17, 2025).

Summary

SQL Server includes built-in functions which may be used to manipulate data. Users can also create custom functions to supplement the built-in functions. These user-defined functions (UDFs) allow users to save commonly-used code as a function that can be called at will. UDFs increase efficiency by eliminating the need to rewrite code, which also decreases coding errors. Three common types of UDFs are scalar, inline, and multi-statement. Scalar functions are used to return a single value. Inline and multi-statement functions are both table-valued functions, meaning they return a table as the result.

Inline table-valued functions are used for less complex queries that can be accomplished using a single SELECT statement. Multi-statement table-valued functions are used for complex queries that may require aggregation. Multi-statement table-valued functions use table variables to store and report aggregated values in the results table.