What are scalar functions

SQL Server scalar function takes one or more parameters and returns a single value.

The scalar functions help you simplify your code. For example, you may have a complex calculation that appears in many [queries](https://www.sqlservertutorial.net/sql-server-basics/sql-server-select/). Instead of including the formula in every query, you can create a scalar function that encapsulates the formula and uses it in each query.

Creating a scalar function

To create a scalar function, you use the CREATE FUNCTION statement as follows:

CREATE FUNCTION [schema\_name.]function\_name (parameter\_list)

RETURNS data\_type AS

BEGIN

statements

RETURN value

END

Code language: SQL (Structured Query Language) (sql)

In this syntax:

* First, specify the name of the function after the CREATE FUNCTION keywords. The schema name is optional. If you don’t explicitly specify it, SQL Server uses dbo by default.
* Second, specify a list of [parameters](https://www.sqlservertutorial.net/sql-server-stored-procedures/sql-server-stored-procedure-parameters/) surrounded by parentheses after the function name.
* Third, specify the data type of the return value in the RETURNS statement.
* Finally, include a RETURN statement to return a value inside the body of the function.

The following example creates a function that calculates the net sales based on the quantity, list price, and discount:

CREATE FUNCTION sales.udfNetSale(

@quantity INT,

@list\_price DEC(10,2),

@discount DEC(4,2)

)

RETURNS DEC(10,2)

AS

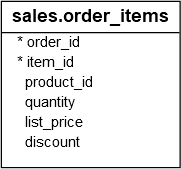
BEGIN

RETURN @quantity \* @list\_price \* (1 - @discount);

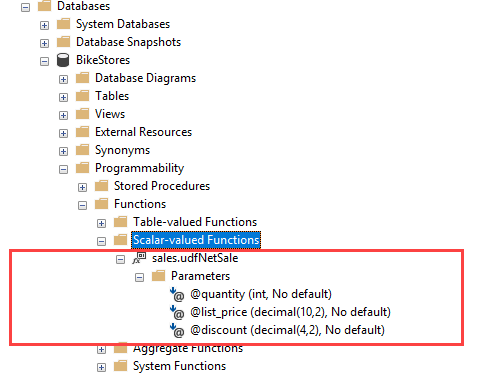
END;

Code language: SQL (Structured Query Language) (sql)

Later on, we can use this to calculate net sales of any sales order in the order\_items from the [sample database](https://www.sqlservertutorial.net/sql-server-sample-database/).



After creating the scalar function, you can find it under **Programmability > Functions > Scalar-valued Functions** as shown in the following picture:



Calling a scalar function

You call a scalar function like a built-in function. For example, the following statement demonstrates how to call the udfNetSale function:

SELECT

sales.udfNetSale(10,100,0.1) net\_sale;

Code language: SQL (Structured Query Language) (sql)

Here is the output:

SQL Server Scalar Function example

The following example illustrates how to use the sales.udfNetSale function to get the net sales of the sales orders in the order\_items table:

SELECT

order\_id,

SUM(sales.udfNetSale(quantity, list\_price, discount)) net\_amount

FROM

sales.order\_items

GROUP BY

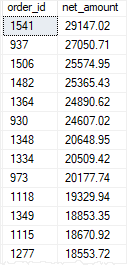
order\_id

ORDER BY

net\_amount DESC;

Code language: SQL (Structured Query Language) (sql)

The following picture shows the partial output:



Modifying a scalar function

To modify a scalar function, you use the ALTER instead of the CREATE keyword. The rest statements remain the same:

ALTER FUNCTION [schema\_name.]function\_name (parameter\_list)

RETURN data\_type AS

BEGIN

statements

RETURN value

END

Code language: SQL (Structured Query Language) (sql)

Note that you can use the CREATE OR ALTER statement to create a user-defined function if it does not exist or to modify an existing scalar function:

CREATE OR ALTER FUNCTION [schema\_name.]function\_name (parameter\_list)

RETURN data\_type AS

BEGIN

statements

RETURN value

END

Code language: SQL (Structured Query Language) (sql)

Removing a scalar function

To remove an existing scalar function, you use the [DROP FUNCTION](https://www.sqlservertutorial.net/sql-server-user-defined-functions/sql-server-drop-function/) statement:

DROP FUNCTION [schema\_name.]function\_name;

Code language: SQL (Structured Query Language) (sql)

For example, to remove the sales.udfNetSale function, you use the following statement:

DROP FUNCTION sales.udfNetSale;

Code language: SQL (Structured Query Language) (sql)

SQL Server scalar function notes

The following are some key takeaway of the scalar functions:

* Scalar functions can be used almost anywhere in T-SQL statements.
* Scalar functions accept one or more parameters but return only one value, therefore, they must include a RETURN statement.
* Scalar functions can use logic such as [IF](https://www.sqlservertutorial.net/sql-server-stored-procedures/sql-server-if-else/) blocks or [WHILE](https://www.sqlservertutorial.net/sql-server-stored-procedures/sql-server-while/) loops.
* Scalar functions cannot [update](https://www.sqlservertutorial.net/sql-server-basics/sql-server-update/) data. They can access data but this is not a good practice.
* Scalar functions can call other functions.