

Genevieve Lim

November 28, 2022

IT FDN 130 A Au 22: Foundations of Databases & SQL Programming

Assignment 07

GitHubURL: <https://github.com/genevivelim/DBFoundations-Module07>

SQL User Defined Functions

Introduction

The purpose of this document is to provide a basic overview of SQL User Defined Functions, also known as UDFs. This document will describe the purpose of a SQL User Defined Function, when a UDF is used, and the different types of User Defined Functions used in SQL.

What is a SQL UDF?

SQL Server already has many useful built-in functions, but database can create customized functions by storing SQL statements in the database. These customized functions are called User Defined Functions (UDFs). (Root, Randal. "Module07-Functions." IT Foundations 130, University of Washington, 23 Nov. 2022. Notes. Accessed 29 Nov. 2022.)

There are two basic types of SQL User Defined Functions: Scalar Functions and Table-valued Functions. These Function types will be described further in the following sections of this document. In SQL Server, Functions are created using the Create Function Statement and executed using a Select Statement as shown by the example code below. (Root, Randal. "Views, Functions, and Stored Procedures." IT Foundations 130, University of Washington, 9 Nov. 2022. Notes. Accessed 29 Nov. 2022.)

```
Create Function dbo.fProducts() # Using the dbo prefix is common in Microsoft SQL Server
```

```
Returns Table
```

```
AS
```

```
Return(
```

```
    Select ProductID, ProductName, CategoryId, Discontinued
```

```
        From Northwind.dbo.Products
```

```
);
```

```
go
```

```
Select * from dbo.fProducts();
```

```
go
```

When is a SQL UDF used?

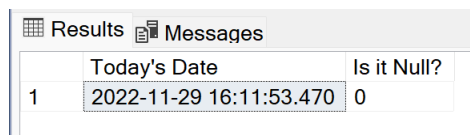
SQL User Defined Functions are used to store simple to complex SQL statements. It allows database users to write complex code once for repeated use, saving time and reducing code clutter. In addition, unlike SQL Views, Functions accepts parameters that result in customized queries. (Root, Randal. "Module07-Functions." IT Foundations 130, University of Washington, 23 Nov. 2022. Notes. Accessed 29 Nov. 2022.)

What is a Scalar Function?

A Scalar Function is a type of UDF that returns a single or scalar value as an expression. Scalar functions are sometimes used for Check constraints for when a user cannot otherwise reference a column in another table. Examples of scalar functions, GetDate() and IsNull(), are shown below. (Root, Randal. "Module07-Functions." IT Foundations 130, University of Washington, 23 Nov. 2022. Notes. Accessed 29 Nov. 2022.)

Select

```
[Today's Date] = GetDate(),  
[Is it Null?] = IsNull(null,0);
```



	Today's Date	Is it Null?
1	2022-11-29 16:11:53.470	0

Figure 1. Results from GetDate() and IsNull() functions from code above.

What is an Inline Function?

An Inline Function is a type of table-valued function that accepts parameters and returns a table of values based on a Select Statement. In an Inline Function, the user does not need to specify a table variable or define the columns and data types of the table. An example of an Inline UDF called fnFilmsByDuration() is shown by the code below, where the user specifies an integer parameter for FilmRunTimeMinutes when executing the function with a Select Statement. ("Inline and multi-statement table-valued functions." WiseOwl, <https://www.wiseowl.co.uk/blog/s347/in-line.htm>. Accessed 29 Nov. 2022.) (External Site)

```
CREATE FUNCTION fnFilmsByDuration(  
    @duration int  
)  
RETURNS TABLE  
AS  
-- function to return all films lasting  
-- more than N minutes  
RETURN  
    SELECT  
        FilmId,
```

```

        FilmName,
        FilmRunTimeMinutes
FROM
    tblFilm
WHERE
    FilmRunTimeMinutes >= @duration

-- show all of the films lasting more than 3 hour 10 minutes
SELECT * from dbo.fnFilmsByDuration(190)

```

What is a Multi-Statement Function?

A Multi-Statement Function is a second type of table-valued function that accepts parameters and returns a table of values. In contrast to an Inline Function, a Multi-Statement Function allows for more complex combinations of Select Statements. It can perform multiple queries within the function, aggregate results from various tables, and return a table with all the combined data.

A Multi-Statement Function requires additional specifications in its syntax. The user must define a table variable, along with columns and associated data types, in the Create Function statement. Also, the Select Statement is enclosed within a Begin/End block. In the example code below, fnPeopleBornYear is a function that returns the names of actors and directors born in a certain year from a films database. The user has to define the table variable, @people and define its columns and the associated data types. In addition, two Select Statements enclosed within a Begin/End block are used to pull data from two different tables, which are then inserted into one resulting table. ("Inline and multi-statement table-valued functions." WiseOwl, <https://www.wiseowl.co.uk/blog/s347/multi-statement.htm>. Accessed 29 Nov. 2022.) (External Site)

```

CREATE FUNCTION fnPeopleBornYear(
    @BirthYear int
)
RETURNS @people TABLE (
    PersonName varchar(50),
    PersonRole varchar(50),
    Dob datetime
)
AS
-- all code lies in a BEGIN / END block
BEGIN
-- insert the actors born in this year into table
INSERT INTO @people (
    PersonName,
    PersonRole,
    Dob
)
SELECT
    ActorName,
    'Actor',
    ActorDob

```

```

FROM
    tblActor
WHERE
    Year(ActorDob) = @BirthYear
-- now add the directors born in this year
INSERT INTO @people (
    PersonName,
    PersonRole,
    Dob
)
SELECT
    DirectorName,
    'Director',
    DirectorDob
FROM
    tblDirector
WHERE
    Year(DirectorDob) = @BirthYear
-- return the results
RETURN
END

```

Summary

Database users can create customized functions called User Defined Functions (UDFs) which are saved in the database. SQL User Defined Functions store simple to complex SQL statements. UDFs allow database users to write complex code once for repeated use, saving time and reducing code clutter. In addition, unlike SQL Views, Functions accept parameters that result in customized queries.

There are two basic types of SQL UDFs: Scalar Functions and Table-valued Functions. A Scalar Function returns a single or scalar value as an expression, while a Table-valued Function returns a table of values. Table-valued Functions are further broken down into two types: Inline Table-valued Functions and Multi-Statement Table-valued Functions. In a simpler Inline Function, a table of data is generated by a Select Statement. The user does not need to specify a table variable or define the columns and data types of the results table. In a Multi-Statement Function, the user can perform multiple queries within the function, aggregate results from various tables, and return a table with all the combined data.