Jourieh Hage
November 29th 2021
Foundations of Databases & SQL Programming
Assignment 07
https://github.com/J-Hage/DBFoundations-Module07

User Defined Functions

Introduction

In this paper we will learn about SQL (Structured Query Language) User Defined Functions (UDFs), what they are, and when we would use them. In addition, we will look at the differences between Scalar, Inline, and Multi-Statement functions.

User Defined Functions

SQL has a variety of built-in functions, such as aggregate, analytic, ranking, conversion, and more. SQL also allows us to create custom functions known as *User Defined Functions* (UDFs). Two types of UDFs are: functions that return a single value and functions that return a table of values.

UDFs are used for several different reasons, including standard programming and faster execution. Once we create a UDF function, it is stored in the database and we can use it in other coding throughout our database. UDFs create faster execution as they do not have to be reoptimized every time a UDF is run since it has already been saved in the database. Finally, UDFs can filter down data by using WHERE clauses that reduce the amount of data returned in the report.

Scalar Functions

Scalar function is one type of UDF. Scalar functions return as an expression a single data value such as a string of text, a number, or a date. Parameters within the scalar function can define the type of data returned. Scalar functions can accept a number of parameters but return only one value.

As shown in Figure 1: Scalar Functions, the function ‘dbo.MultiplyValues’ was created to take the expression of ‘value 1’ multiplied by ‘value 2’ to return a single value. Then, when applied into a SELECT statement the function returns only the ‘Extended Price’ values in the result set.

```
Create Function dbo.MultiplyValues(@Value1 Float, @Value2 Float)
 Returns Float
 As
  Begin
   Return(Select @Value1 * @Value2);
  End
go
-- Calling the function
Select Tempdb.dbo.MultiplyValues(4, 5);
go
```

```
Select
 SalesId
,SalesLineItemId
,ProductId
,SalesPrice
,SalesQty
,dbo.MultiplyValues(SalesPrice,SalesQty) as ExtendedPrice
From dbo.SalesDetails
```

Here are the results:

Results | Messages

	SalesId	SalesLineItemId	ProductId	SalesPrice	SalesQty	ExtendedPrice
1	1	1	100	9.99	10	99.9
2	1	2	200	1.00	5	5

Figure 1: Scalar Functions

Inline Functions

Inline functions are another type of UDF. Inline functions return the result set as a table. As with scalar functions, inline functions can accept a number of parameters. In contrast to scalar functions, inline functions return a table set of results.

As shown in Figure 2: Inline Functions, the function 'fnFilmsByDuration' returns a table of results with the parameter of the 'FilmRunTimeMinutes' being greater than or equal to a specified value. When the function is applied in a SELECT statement specifying the parameter to be 190 minutes, it returns a table result of all the films with run times greater than or equal to 190 minutes.

```
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)
```

FilmId	FilmName	FilmRunTimeMinutes
6	Titanic	194
73	The Lord of the Rings: Return of the King	201
188	Schindler's List	195
260	Seven Samurai	190

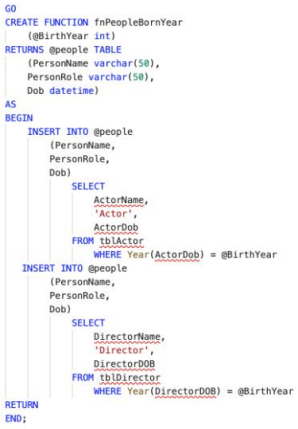
Figure 2: Inline Functions

[Inline and Multi-Statement Table Valued Functions](#) **(External Site)**

Multi-Statement Function

Multi-Statement functions are a third type of UDF. Similar to inline functions, multi-statement functions also return a table of results. However, in contrast to inline functions, multi-statement functions require additional processing of a temporary table that is directly constructed in the script.

As shown in Figure 3: Multi-Statement Functions, the function 'fnPeopleBornYear' has a parameter defined as '@BirthYear'. The output of the function is established in a temporary table '@people', where the structure of the table and data types are defined. Next, the function begins by inserting the correct data. When the function is applied in a SELECT statement, specifying the parameter to be the birth year of 1945, it returns a table result showing all actors and directors born in the year 1945.



```
GO
CREATE FUNCTION fnPeopleBornYear
    (@BirthYear int)
RETURNS @people TABLE
    (PersonName varchar(50),
    PersonRole varchar(50),
    Dob datetime)
AS
BEGIN
    INSERT INTO @people
        (PersonName,
        PersonRole,
        Dob)
            SELECT
                ActorName,
                'Actor',
                ActorDob
            FROM tblActor
                WHERE Year(ActorDob) = @BirthYear
    INSERT INTO @people
        (PersonName,
        PersonRole,
        Dob)
            SELECT
                DirectorName,
                'Director',
                DirectorDOB
            FROM tblDirector
                WHERE Year(DirectorDOB) = @BirthYear
RETURN
END;
```

```
-- show actors and directors born in 1945
SELECT
  PersonName AS Who,
  PersonRole AS What,
  CONVERT(char(10),Dob,103) AS DOB
FROM
  dbo.fnPeopleBornYear(1945)
ORDER BY
  Dob
```

Who	What	DOB
Roger Spottiswoode	Director	05/01/1945
John Lithgow	Actor	19/10/1945
Nicholas Meyer	Director	24/12/1945
Henry Winkler	Actor	30/10/1945

Figure 3: Multi-Statement Functions

[Inline and Multi-Statement Table Functions](#) **(Link to External Site)**

Summary

In summary, UDFs are customized functions that can have parameters which return a single value or a table of results. A scalar function returns a single value. An inline function returns a table set of results. Finally, a multi-statement function also returns a table set of results but uses a temporary table constructed within the function to return the complex results.

References

Root, Randy. Professor at University of WA. Teaching Foundations of Databases & SQL Programming course.

Scalar UDF Inlining. (n.d.) Microsoft SQL Docs. Retrieved from external site. https://docs.microsoft.com/en-us/sql/relational-databases/user-defined-functions/scalar-udf-inlining?view=sql-server-ver15 (External Site Link).

Wise Owl Training. *Inline and multi-statement table-valued functions Blog.* (n.d.) Retrieved from external site. https://www.wiseowl.co.uk/blog/s347/table-valued-functions.htm (External Site Link).