Programming SQL Server Database Triggers and Functions

SCALAR AND MULTI-STATEMENT TABLE-VALUED FUNCTIONS

Ryan Booz
AUTHOR AND SPEAKER

@ryanbooz https://www.softwareandbooz.com

Overview

Review Multi-Statement User-Defined Functions

Create examples of Scalar and Multi-Statement Table-Valued Functions

Demonstrate how to use them in SELECT statements and WHERE clauses

Complications with SARGability and Multi-Statement functions

Discuss improvements in SQL Server 2017 and 2019 that improve Multi-Statement Function performance

A Closer Look at Multi-Statement User-Defined Functions

They are an essential part of productive SQL Server development

Function Review

- Many built-in functions that we use every day
- Can have inputs parameters only, including DEFAULT values
- Must return a value
- They can query and modify data locally, but no DML or DDL modifications are allowed
- Two broad categories of Multi-Statement and Single-Statement Functions
- Until recent versions of SQL Server, Multi-Statement functions cannot be “inlined”

Multi-Statement Scalar Functions

All scalar functions must have a BEGIN and END

One or more statements that must return a value of the specified type

Can be used as a value in DML statements, some schema definitions, and WHERE clauses

Multi-Statement Scalar Function

```
CREATE OR ALTER FUNCTION dbo.SuperAdd_scaler(@a INT, @b INT)
RETURNS INT
WITH SCHEMABINDING
AS
BEGIN
    RETURN @a + @b;
END;
```

Multi-Statement Scalar Function

```
CREATE OR ALTER FUNCTION dbo.SuperAdd_scaler(@a INT, @b INT)
RETURNS INT
WITH SCHEMABINDING
AS
BEGIN
    DECLARE @SumTotal INT = @a + @b;
    RETURN @SumTotal;
END;
```

```
SELECT MONTH('2019-01-01');
```

```
SELECT MONTH(OrderDate) FROM
    Sales.Orders;
```

```
SELECT * FROM Sales.Orders
    WHERE YEAR(OrderDate) = 2019
```

- Scalar functions with direct input parameter
- Scalar functions in DML statements utilizing table data
- Scalar functions as WHERE clause predicates
- Not a preferred method of using scalar functions

Multi-Statement Table-Valued Functions

Declare the table and columns that will be returned from the function

Must have a BEGIN and END

One or more statements that must populate and return the declared table

Can be used anywhere a table input can be used

Multi-Statement Table-Valued Function

```
CREATE OR ALTER FUNCTION dbo.SuperAdd_tvf(@a INT, @b INT)
RETURNS @SumValueTable TABLE
( SumValue INT )
WITH SCHEMABINDING
AS
BEGIN
    INSERT INTO @SumValueTable (SumValue)
        SELECT @a + @b;

    RETURN;
END;
```

```
SELECT * FROM
    dbo.CustSaleByYear();
```

- MSTVF as independent table

```
SELECT * FROM Customers C
    INNER JOIN
dbo.CustSaleByYear() ACS
    ON C.CustID = ACS.CustID
```

- MSTVF in DML statements utilizing table data

```
SELECT CustID FROM Customers C
CROSS APPLY
(SELECT * FROM
    dbo.CustSaleByYear(2019)
    WHERE CustID = C.CustID) ACS
```

- MSTVF filtered using CROSS APPLY

Challenges with Multi-Statement Functions

Multi-Statement Scalar and Table-Valued Functions suffer from known performance problems...

...until SQL Server 2017 & 2019

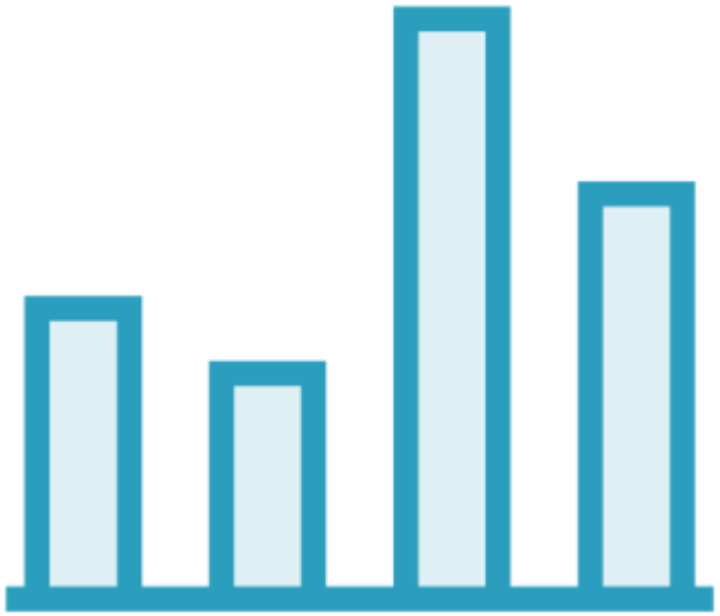
SARGability

SARG = Search Arguments

Used to identify indexes to use, which helps SQL Server find data faster

If we use any function on the left-hand side of the predicate, SQL Server can no longer use indexes

Consider using pre-computed data in some way that can have SARGable indexes applied



Bad Statistics

Multi-Statement Table-Valued functions are always given an estimate of 100 rows starting with SQL Server 2014

Severely limits the Query Optimizer's ability to correctly find the best plan

Performance Improvements in SQL Server 2017 and 2019

Interleaved Execution for MSTVF's

Introduced in SQL Server 2017

Allows SQL Server to pause execution of the main query, execute the MSTVF to get proper statistics, and then continue

The MSTVF query plan is cached and used to better adapt the overall query

Subsequent queries get access to the cached plan

Even Query Store can be used to track and display information about the cached plan

Interleaved Execution for MSTVF's

Interleaved Eligability

Must be on Compatability Level 140 or higher

Referencing statements must be read-only and not part of a data modification operation

MSTVF's cannot be the target of a CROSS APPLY.

Scalar UDF Inlining

Introduced in SQL Server 2019

Similar to Interleaved Execution but for Scalar Functions

Transforms the Scalar UDF into an expression that can be "inlined" to the calling query

Essentially removes the Scalar UDF Operator in the query plan that you see now

Scalar UDF Inlining

Inlining Scalar UDF Requirements

- **Must be on Compatability Level 150 or higher**
- **Cannot use time-dependent functions (GETDATE, NEWID)**
- **Must execute as CALLER**
- **Various restrictions around GROUP BY and ORDER BY clauses**
- **Check the “Is_Inlinable” column of ‘sys.sql_modules’**

https://docs.microsoft.com/en-us/sql/relational-databases/user-defined-functions/scalar-udf-inlining

Summary

- Reviewed Multi-Statement Function basics
- Created Scalar UDFs and Multi-Statement Table-Valued UDFs
- Discussed the problem with SARGable predicates when using functions
- Improvements with SQL Server 2017 and 2019