Madeline Miller
May 29, 2024
Foundations Of Databases & SQL Programming
Assignment 07

User Defined Functions (UDFs) in SQL

Introduction

User-defined functions (UDFs) are customized functions in SQL. This document reviews the purposes of UDFs and the different types of UDFs: Scalar, Inline, and Multi-Statement.

When to use a SQL UDF

Microsoft details three main benefits[1] for the use of UDFs, summarized as follows:

- **Modular Programming:** functions are reusable building blocks that allow for calling the same set of complicated procedures while writing out the steps only once.
- **Faster Execution:** Storing a single set of steps means that functions can allow code to run faster.
- **Reduce Network Traffic:** Functions can be used as a filter in a where clause to reduce the total amount of records sent to the client via the network.

For our purposes, we'll focus on modular programming, which is the most easily understood benefit of creating UDFs in cases where execution and network traffic may not be relevant.

Functions are useful for repeatedly performing a set of actions, simplifying the code needed to return results. Built-in SQL functions, like FORMAT(), allow you to enter parameters but don't let you define the set of operations or output types. UDFs are beneficial because they allow users to create a customized block of code where they can define what input is needed, the procedures, and what will be returned.

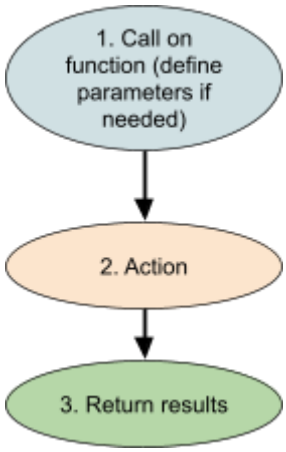

Figure 1: Function Steps

[1] (Microsoft Learn, https://learn.microsoft.com/en-us/sql/relational-databases/user-defined-functions/user-defined-functions?view=sql-server-ver16, 2024)(External Site)

Scalar, Inline, and Multi Statement Functions

Scalar Functions

Scalar functions return a single value. These can be used in a variety of cases such as formatting and calculations.

```
560  CREATE FUNCTION  CelsiusToFahrenheit(@Celsius FLOAT)
561  RETURNS FLOAT
562  AS
563      BEGIN
564          RETURN(@Celsius * 9/5 + 32)
565      END
566  GO
567
568  SELECT dbo.CelsiusToFahrenheit(12) AS Fahrenheit
569
```

Results | Messages

	Fahrenheit
1	53.6

Figure 2: Example Scalar function

Table-Valued Functions (TVFs)

Table valued functions (TVFs) return a table. There are two types of table valued functions, inline and multi statement.

Inline TVFs

Inline tvfs return a table. They don't use a 'begin' and 'end' statement. These are useful for returning tables based on input parameters.

```
541  CREATE FUNCTION fProductInventoriesWithPreviousMonthCountsWithKPIs(@KPIvalue INT)
542  RETURNS TABLE
543  AS
544      RETURN(
545          SELECT TOP 10000000000000
546              ProductName,
547              InventoryDate,
548              [Count],
549              PrevMonthCount,
550              MonthKPI
551          FROM vProductInventoriesWithPreviousMonthCountsWithKPIs
552          WHERE MonthKPI = @KPIvalue
553          ORDER BY ProductName, CAST(InventoryDate AS DATE)
554      );
555  go
```

Figure 3: Example inline function

Multi Statement TVFs

Multi statement TVFs also return a table, but can contain multiple statements and more complicated logic. Like scalar functions, they use a ‘begin’ and ‘end’ statement to contain their logic. These are useful for building table results with more complex logic.

```sql
CREATE FUNCTION dbo.GetSalesSummary (@Year INT)
RETURNS @SummaryTable TABLE
(
    Month INT,
    TotalSales DECIMAL(18, 2)
)
AS
BEGIN
    DECLARE @Month INT = 1;

    WHILE @Month <= 12
    BEGIN
        INSERT INTO @SummaryTable (Month, TotalSales)
        SELECT @Month, SUM(SalesAmount)
        FROM Sales
        WHERE YEAR(SalesDate) = @Year AND MONTH(SalesDate) = @Month;

        SET @Month = @Month + 1;
    END

    RETURN;
END;
```

Figure 4: Example multi statement function written by chatgpt. (OpenAI ChatGPT, https://chatgpt.com/c/, May 2024)

Summary

- UDFs are useful for building custom functions where you can define what parameters go in, what actions are performed, and what results are returned
- Scalar Functions
  - Return a single value as the result
  - Use ‘begin’ and ‘end’ statements
- Table-Valued Functions
  - Return a table as the result
  - Two types
    - Inline
      - Basic - do not use ‘begin’ and ‘end’ statements
      - Only use a return
    - Multi Statement
      - Can build tables with multiple statements and complex logic
      - Use ‘begin’ and ‘end’ statements