Wendy Olson
11/28/23
IT Foundations of Database
Management & SQL Programming
Assignment07

Functions

Introduction – This document briefly describes the different types of Functions and when they are used. In general, functions are special operations or expressions that can perform various tasks on data. Functions can take one or more parameters as input, process the input, and return a result.

What are SQL User Defined Functions (UDF), and when are they used?

An SQL User-Defined Function (UDF) is a custom function created by the user to perform a specific task. Unlike built-in or system functions provided by the database management system, UDFs are stored procedures built by the user that can simplify complex queries, enhance reusability, and promote modular code design. Types of UDFs include scalar functions, which return a single value based on the input parameters, or table-valued functions, which return a table.

What are the differences between Scalar, Inline, and Multi-statement Functions?

Scalar, inline, and multi-statement functions are different types of user-defined functions (UDFs), each with its own characteristics and use cases.

Scalar Functions take input parameters and return a single scalar value (e.g., integer, string, date), and therefore are suitable for tasks that involve calculations or transformations on a single value (Figure 1).

```
-- Single Value (Scalar) Functions
Create Function dbo.AddValues(@Value1 Float,@Value2 Float)
 Returns Float
 As
  Begin
   Return(Select @Value1 + @Value2);
  End
Go
-- Calling the function
Select dbo.AddValues(4, 5);
Go
```

Figure 1. Example of a Scalar Function returning a single value. (RRoot, Mod7 Notes UW IT Database Management).

Inline Table-Valued Functions return a table result. They are defined using a single RETURN statement with a SELECT query. They are typically used for tasks requiring a table result based on input parameters (Figure 2).

```
-- Simple Table Value (Tabluar) Functions
Create Function dbo.ArithmeticValues(@Value1 Float, @Value2 Float)
 Returns Table
 As   --Begin << Cannot use Begin
  Return( Select [Sum] = @Value1 + @Value2,
 [Difference] = @Value1 - @Value2,
    [Product] = @Value1 * @Value2,
   [Quotient] = @Value1 / @Value2 );
 --End << Or End with Simple Table Value Functions
go
Select * FRom ArithmeticValues(4,5)
```

Figure 2. Example of an Inline-Table-Valued Function returning a Table. (RRoot, Mod7 Notes UW IT Database Management).

Wendy Olson
11/28/23
IT Foundations of Database
Management & SQL Programming
Assignment07

Multi-Statement Table-Valued Functions also return a table result set; however, they are defined using multiple BEGIN and END blocks with intermediate processing steps. They can have local variables and perform more complex logic than inline functions, typically used when the function's logic requires multiple steps or when local variables are needed (Figure 3).

```
-- Complex Table Value (Tabluar) Functions with Multiple Statements
Create Function dbo.fArithmeticValuesWithFormat(@Value1 Float, @Value2 Float, @FormatAs char(1))
 Returns @MyResults Table
   ( [Sum] sql_variant
   , [Difference] sql_variant
   , [Product] sql_variant
   , [Quotient] sql_variant  )
 As
  Begin --< Must use Begin and End with Complex table value functions
   If @FormatAs = 'f'
    Insert Into @MyResults
   Select Cast(@Value1 + @Value2 as Float)
        ,Cast(@Value1 - @Value2 as Float)
       ,Cast(@Value1 * @Value2 as Float)
       ,Cast(@Value1 / @Value2 as Float)
   Else If @FormatAs = 'i'
    Insert Into @MyResults
   Select Cast(@Value1 + @Value2 as int)
        ,Cast(@Value1 - @Value2 as int)
       ,Cast(@Value1 * @Value2 as int)
       ,Cast(@Value1 / @Value2 as int)
  Else
    Insert Into @MyResults
   Select Cast(@Value1 + @Value2 as varchar(100))
        ,Cast(@Value1 - @Value2 as varchar(100))
       ,Cast(@Value1 * @Value2 as varchar(100))
       ,Cast(@Value1 / @Value2 as varchar(100))
  Return
  End
go
```

Figure 3. Example of a Multi-Statement Table-Valued Function.
(RRoot, Mod7 Notes UW IT Database Management).

Summary – In summary, scalar functions return a single value, while inline and multi-statement functions return table result sets, but how they are defined differs. Scalar functions have a straightforward structure with a single RETURN statement, while inline functions have a more straightforward structure with a single RETURN statement containing a SELECT query. Multi-statement functions have a more complex structure with multiple BEGIN and END blocks, allowing for more complex logic that requires multiple steps, local variables, or intermediate processing.