

**Name:** Hyungu Kim

**Date:** NOV 30, 2022

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

## Assignment 7 – Functions

### Introduction

UDF stands for User-Defined Function (UDF), which lets the users to create a function by using a SQL expression. A UDF accepts columns of input, performs actions on the input, and returns the result of those actions as a value. The users should use UDFs when they have a calculation they will repeat throughout the database. Scalar, Inline, and Multi-Statement functions are UDFs. Scalar function accepts any number of parameters and returns one value. Inline Function returns a result set, much like a view. However, unlike a view, functions can accept parameters. Multi-Statement functions can be used to do some unique things outside the context of a standard “select” statement.

### Topic

The main purpose of the assignment was to understand the use of “function” statement at the end of the questions. However, throughout the course of questions, I realized how important it is to pay attention to detail when writing codes.

```
CREATE VIEW vProductInventoriesWithPreviousMonthCountsWithKPIs
AS
    SELECT TOP 1000000
        ProductName, InventoryDate, InventoryCount, [PreviousMonthCount], [CountVsPreviousCountKPI] = ISNULL(CASE
            WHEN InventoryCount > [PreviousMonthCount] THEN 1
            WHEN InventoryCount = [PreviousMonthCount] THEN 0
            WHEN InventoryCount < [PreviousMonthCount] THEN -1
            END, 0)
    FROM vProductInventoriesWithPreviousMonthCounts
    ORDER BY 1, Month(InventoryDate), 3;
GO

-- Important: This new view must use your vProductInventoriesWithPreviousMonthCounts view!
-- Check that it works: SELECT * FROM vProductInventoriesWithPreviousMonthCountsWithKPIs;
SELECT * FROM vProductInventoriesWithPreviousMonthCountsWithKPIs;
GO
```

**Figure 1. When writing wrong code**

In the figure above in the question 7, I have made a typing mistake for the long statement called “vProductInventoriesWithPreivousMonthCountsWithKPIs”. As you can see it is a very long title which the users can easily make mistakes writing it. As the result of the typing mistake, I had to spend long time figuring out why question 8’s codes were not working. Therefore, the users must be careful and pay more attention when writing codes especially when codes are connected throughout the entire database.

```

Create Function fProductInventoriesWithPreviousMonthCountsWithKPIs (@KPIValue int)
Returns Table
As
Return Select
    ProductName, InventoryDate, InventoryCount, [PreviousMonthCount], [CountVsPreviousCountKPI]
From vProductInventoriesWithPreviousMonthCountsWithKPIs
Where [CountVsPreviousCountKPI] = @KPIValue;
go

/* Check that it works:
Select * From fProductInventoriesWithPreviousMonthCountsWithKPIs(1);
Select * From fProductInventoriesWithPreviousMonthCountsWithKPIs(0);
Select * From fProductInventoriesWithPreviousMonthCountsWithKPIs(-1);
*/
Select * From fProductInventoriesWithPreviousMonthCountsWithKPIs(1);
Select * From fProductInventoriesWithPreviousMonthCountsWithKPIs(0);
Select * From fProductInventoriesWithPreviousMonthCountsWithKPIs(-1);

```

**Figure 2. Functions**

As you can see in the above figure 2. The function statement allows the users to calculate with different set of numbers without writing multiple different codes. This is a useful function in real world which can be applied into our everyday life such as mortgage calculations and tax calculations.

## Summary

UDF, user defined functions enables the users to create a function which allows the users to use the functions without excessive amount of codes. Thus, the users should use UDFs when they have a calculation they will repeat throughout the database, which can be really helpful in any sort of career fields. However, the users must be more attentive when using this functions and writing excessively long codes which may result in further consequences.