

**Name:** MariiaK

**Date:** 6/6/2024

**Course:** Foundations Of Databases & SQL Programming

**GitHub:** <https://github.com/mustbekot/DBFoundations-Module07>

# Assignment 7 - SQL User-Defined Functions

## Introduction

This document provides an overview of SQL User-Defined Functions (UDFs), explaining their usage and the differences between Scalar, Inline, and Multi-Statement Functions. Understanding these UDFs is essential for efficient and modular SQL code management.

## When to Use a SQL UDF

SQL UDFs encapsulate reusable logic for use within queries, promoting code reusability, modularity, and consistency. They are ideal for abstracting complex calculations, ensuring consistent logic application, and optimizing performance.

## Differences Between Scalar, Inline, and Multi-Statement Functions

### Scalar Functions

Scalar functions return a single value of a specified data type and are used for calculations or operations that produce one result. They are suitable for mathematical operations, string manipulations, and date calculations.

### Inline Table-Valued Functions (Inline TVFs)

Inline TVFs return a table and are defined using a single **SELECT** statement, making them efficient and similar to views. They are used to return a set of rows based on input parameters.

### Multi-Statement Table-Valued Functions (Multi-Statement TVFs)

Multi-Statement TVFs return a table constructed through multiple **BEGIN . . . END** statements, suitable for more complex logic. They can be less performant due to the overhead of managing the table variable and complex operations.

## Summary

SQL UDFs, including Scalar, Inline TVFs, and Multi-Statement TVFs, enable reusable and efficient SQL code. Each type serves different scenarios: Scalar for single values, Inline TVFs for table results with simple logic, and Multi-Statement TVFs for complex logic returning tables. Understanding their use enhances SQL code efficiency and maintainability.