

Name: **Devaki Deshpande**

Date: **February 28, 2024**

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

Assignment No: **07**

Functions

Introduction:

SQL User-Defined Functions (UDFs) are essential tools for improving code and database efficiency. This paper explores when and how to use SQL UDFs effectively, focusing on their practical applications and distinctions between Scalar, Inline, and Multi-Statement Functions.

1. When to Use a SQL UDF:

SQL User-Defined Functions (UDFs) are handy for encapsulating complex logic or calculations into reusable components within SQL queries. They're great for repetitive calculations or when built-in SQL functions fall short. By creating a UDF, you enhance code readability, reusability, and maintenance, especially for repetitive tasks or custom logic.

2. Differences Between Scalar, Inline, and Multi-Statement Functions:

Scalar, inline, and multi-statement functions are all SQL UDFs, but they vary in structure and behavior

- **Scalar functions:** These return a single value based on input parameters and are ideal for simple calculations within queries.
- **Inline functions:** These are optimized scalar functions, expanded directly into the calling query, offering better efficiency.
- **Multi-statement functions:** These can contain multiple SQL statements and are useful for complex logic, although they're less efficient due to overhead.

In short, scalar functions are simple, inline functions are optimized, and multi-statement functions handle more complex tasks, each serving different needs based on efficiency and complexity.

Summary:

SQL User-Defined Functions (UDFs) are crucial for organizing complex logic in SQL queries. This paper breaks down their importance and differences, from simple to complex functions, aiming to help developers optimize queries and enhance database performance.