Kaiyue Xu
Mar 10, 2025
IT FDN 130 A
Assignment 07

# SQL UDF

## When to use a SQL UDF

When you need to write a custom function to carry out a certain task that isn't possible with conventional SQL functions, you would use a SQL UDF. Because UDFs enable modular, reusable code, queries become more manageable and succinct.

- Complex logic encapsulation: When intricate computations or text manipulations are required that are difficult to do with conventional SQL methods.
- Reusability: When a function's logic must be applied consistently across several queries.
- Enhancing readability: By isolating the logic within a function, a UDF may streamline complicated queries, making them simpler to read and comprehend.
- Performance optimization: By eliminating repetitious logic or the need to recalculate data in a query, UDFs can sometimes increase performance.

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

1. Scalar Functions:
   Depending on the input values, a scalar function yields a single value, or scalar. It may be applied anytime a value is required, including in `WHERE` clauses and `SELECT` queries.
2. Inline Functions:
   Inline functions just require one `SELECT` query and `return` a result set (table). In essence, they are perspectives with restrictions.
3. Multi-Statement Functions:
   A multi-statement function returns a single value or a table and can include several statements, such as loops, conditionals, and variable declarations. Because of the additional processing required, it might be slower than scalar or inline functions despite being more versatile.

Each type of function serves different purposes and has its own strengths depending on the complexity of the task and the performance requirements.