Daniel Hively

August 22, 2021

Foundations of Databases & SQL Programming

Assignment 07

Github: https://github.com/danjhively/DBFoundations

**Functions Written Assignment**

**Introduction**

We expanded on last week's introduction to views, functions, and stored procedures to focus more exclusively on functions in this week's module. We were shown a sampling of built-in functions, but in this week's written assignment I will be covering User-Defined Functions (UDFs). This includes when to use them, and what are the differences between the three types (scalar, inline, multi-statement).

**When to use a SQL UDF**

There are many reasons to implement a User-Defined Function (though there is overlap between views, functions, and stored procedures, this will just look at functions). If you want to store a set of SQL statements that won't add/alter/delete data then a UDF would be reasonable to use. Additionally if the desired output can conform to a scalar or table then a UDF would be useful. They also allow the use of input parameters into the set of SQL statements. Another practical use is in implementing a Check constraint on table fields.

**Scalar vs. Inline vs. Multi-Statement Functions**

* Scalar Function
  + Returns a scalar value
  + Can accept parameters
* Inline Function
  + Returns a table value
  + Can accept parameters
  + Doesn't define the returned table
  + Structure is primarily inside the Return statement where the table values get selected
* Multi-Statement Function
  + Returns a table value
  + Can accept parameters
  + Defines the returned table
  + Structure is primarily outside the Return statement where the created table gets filled in

**Conclusion**

Last week we touched on the utility of these tools, and this week we've expanded even further on that. While the built-in functions are handy, the UDFs were shown to be extremely powerful tools that are valuable in many scenarios. Additionally, in understanding the differences between the assorted types of functions, we see the varying strengths each type has so we understand when/where we would want to implement them.