



T-SQL Functions Problem and Solution

Andy Novick

www.NovickSoftware.com

anovick@novicksoftware.com

Read More: <http://www.novicksoftware.com/problem-user-defined-functions-solution/>

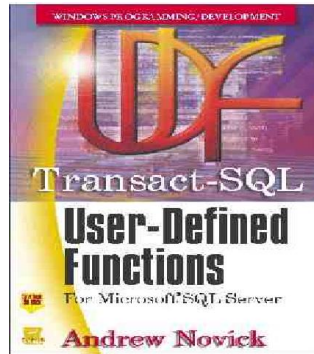
<http://bit.ly/1virMpE>

Types of Functions

- Scalar
 - Returns a single scalar value
- Table Valued
 - Returns a table
- Inline
 - A view with parameters that returns a table



T-SQL UDFs the Book Free PDF



<http://www.novicksoftware.com/transact-sql-user-defined-functions-pdf-available-free/>

<http://bit.ly/1rnsVN2>

Exponentially Weighted Moving Average

$$EMA_{Today} = (Value_{Today} * \left(\frac{Smoothing}{1 + Days}\right)) \\ + EMA_{Yesterday} * \left(1 - \left(\frac{Smoothing}{1 + Days}\right)\right)$$

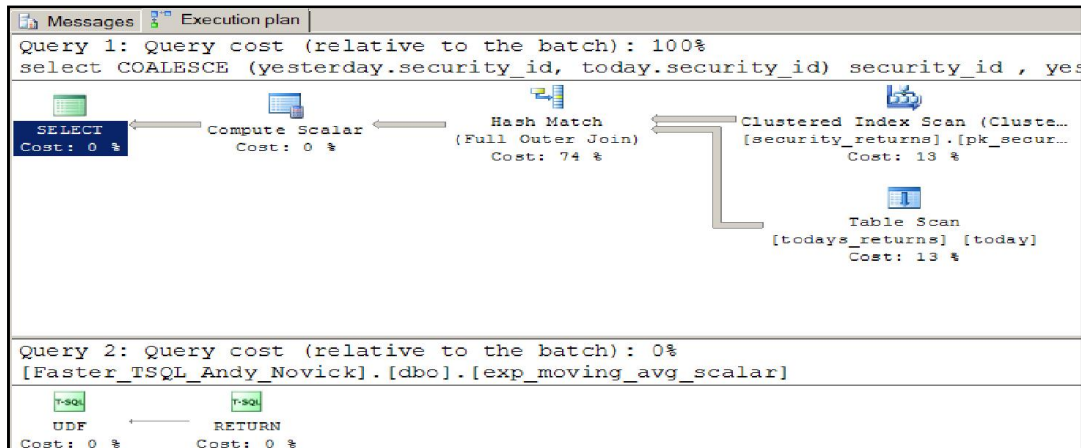
Example Scalar UDF

```
CREATE FUNCTION [dbo].[exp_moving_avg_scalar] (  
    @ema_yesterday FLOAT  
    , @value_today FLOAT  
    , @days INT  
    , @smoothing FLOAT = 2.0  
  
    ) RETURNS FLOAT  
AS BEGIN  
  
    RETURN CASE WHEN @ema_yesterday IS NULL  
        THEN @value_today  
        ELSE (@value_today * (@smoothing/(1.0+@days)))  
            + (@ema_yesterday  
                * (1.0 - (@smoothing/(1.0+@days))))  
        END  
END
```

The Problem with Scalar UDFs!

- Row-by-Row processing
 Row by Agonizing Row - RBAR
- Inhibits Parallelism

Query Plan with A Scalar



Solution – Inline UDF

- A view with parameters
- The SQL optimizer merges the UDF's **SELECT** with the rest of the query
- Works in SQL 2000 through SQL 2014

Example Inline UDF

```

CREATE FUNCTION [dbo].[exp_moving_avg_inline] (
    @ema_yesterday FLOAT
    , @value_today FLOAT
    , @days INT
    , @smoothing FLOAT = 2.0
) RETURNS TABLE
AS RETURN

SELECT CASE WHEN @ema_yesterday IS NULL
    THEN @value_today
    ELSE (@value_today * (@smoothing/(1.0+@days)))
        + (@ema_yesterday
            * (1.0 - (@smoothing/(1.0+@days))))
    END
    as ema_today

```

Example Scalar UDF

```

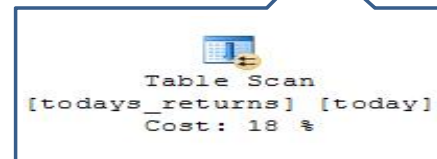
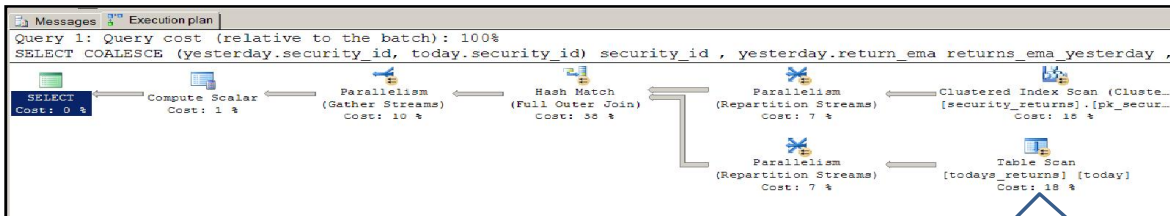
CREATE FUNCTION [dbo].[exp_moving_avg_scalar] (
    @ema_yesterday FLOAT
    , @value_today FLOAT
    , @days INT
    , @smoothing FLOAT = 2.0
) RETURNS FLOAT
AS BEGIN

RETURN CASE WHEN @ema_yesterday IS NULL
    THEN @value_today
    ELSE (@value_today * (@smoothing/(1.0+@days)))
        + (@ema_yesterday
            * (1.0 - (@smoothing/(1.0+@days))))
    END

END

```

Query Plan with Inline UDF



Is Using UDFs OK?

```
SET @new_returns_ema =
    dbo.exp_moving_avg_scalar(@yesterday_return_ema
        , @returns_today
        , 21
        , 2.0
    )
SELECT dbo.exp_moving_avg_scalar(yesterday.return_ema
    , today.returns_today, 21, 2.0) new_ema
FROM security_returns yesterday
FULL OUTER JOIN today's_returns today
    on yesterday.security_id = today.security_id
WHERE dbo.return_confidence(returns_today) > 3.42
```



Yes, It is OK to use a UDF, when:

- There are a small number of rows involved
- Scalar UDF is converted to an Inline-UDF



T-SQL Functions Problem and Solution

Andy Novick
www.NovickSoftware.com
anovick@novicksoftware.com

Read More: <http://www.novicksoftware.com/problem-user-defined-functions-solution/>

<http://bit.ly/1virMpE>