

Sergio Govoni



#sqlsat871



 PASS
SQLSATURDAY
SARDEGNA | 18 MAY 2019

Set-based vs Iterative programming

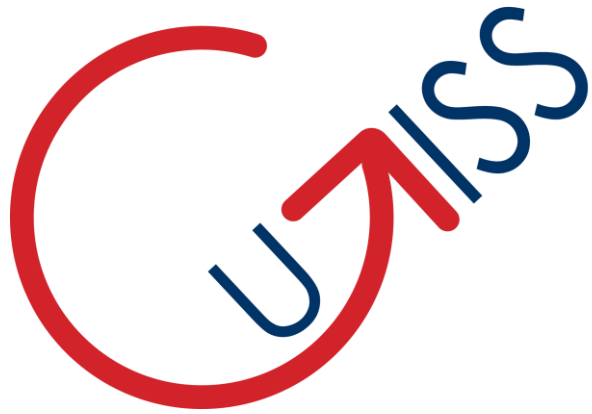
Sponsors



#sqlsat871



Organizational Sponsors



www.ugiss.org



www.unica.it



www.pass.org



Sergio Govoni

CTO at Centro Software – The ERP Company



Microsoft Data Platform MVP since 2010



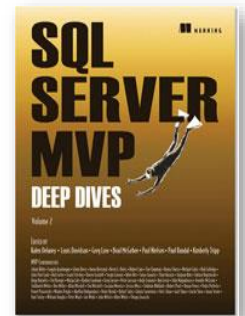
Vice President at UGISS



Co-Author

MVP Deep Dives Volume 2

<http://www.manning.com/delaney/>



#sqlsat871



Agenda

- When can I say “This query works properly”?
- Rewrite you query, T-SQL is a declarative language!
- Set-Based or Iterative code?
- Real-world scenarios
 - Barcode check-digit
 - Running total
 - Bill of materials

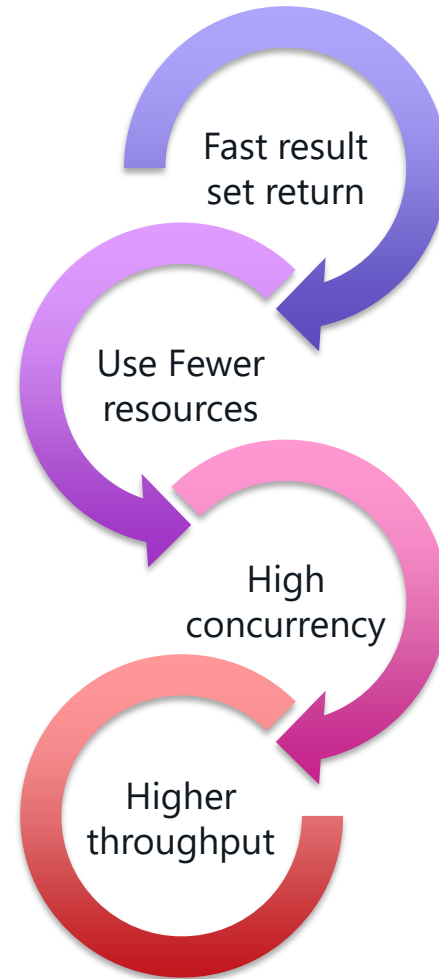


Test Environment

- Download and install the last version of SQL Server Management Studio
 - <https://bit.ly/2K7Jzgu>
- Connect to Azure SQL Database
 - **Server:** sqlsat871.database.windows.net
 - **User:** UserSqlSat871
 - **Password:** L9A5EKCyBbQ4BjT2NMWgfhVaZ
 - **Database:** AdventureWorks2017
- Enjoy 😊



When can I say "This query works properly"?



T-SQL language

- T-SQL, compared to the other languages, it's not difficult to learn and it can be very tolerant
- Many people start with SELECT, INSERT, UPDATE and DELETE
- Few people realize that it's a declarative language
 - Declare only what do you want and leave the rest to the Query Optimizer! 😊



T-SQL language

- The code wrote isn't completely wrong 😊
- It has been influenced by
 - The company's strategy
 - The person's mood
 - The technology used and how it's known
- If you know all the new language functionalities, you could use the right one at the right time!



Set Calculations

- Calculations applied to a set of rows
- Type
 - Ranking
 - Groupings
 - Offset
 - Distributions



Set-Based or Iterative Code?

- They are two approaches to develop T-SQL solutions
 - Each Developer knows them
 - Iterative approach is the most used
- Why is the set-based approach the recommended one?
- If it is the recommended one, why do many Developers use the iterative approach?



Why do developers use the Iterative approach?

- The Set-Based approach is not intuitive 😞
- To build Set-Based solution you have to know T-SQL deeply
- Set-Based thinking is especially difficult for those people who have procedural and iterative programming background
 - Example: Working with files



Why do developers use the iterative approach?

- Our brain thinks data in this way
 - Ordered
 - Handled one record at a time
- Cursor solutions, through fetch operations, guarantee
 - Ordered result sets return
 - Handled one record at a time



Set Definition

By a "set" we mean any collection M into a whole of definite, distinct objects m (which are called the "elements" of M) of our perception or of our thought

Georg Cantor



A large, stylized teal graphic element on the left side of the slide, consisting of several overlapping, curved, parallel lines that form a shape reminiscent of a stylized arrow or a modern letter 'R'.

Real-world scenarios

Calculate barcode check-digit using T-SQL

- Suppose that your boss asked you to implement a T-SQL object, in an SQL Server database, that it is able to calculate the check-digit for a given EAN code
- How can you translate the standard algorithm to T-SQL code?



<https://en.wikipedia.org/wiki/Barcode>



Calculate barcode check-digit using T-SQL

- The calculation algorithm requires that every digit in the code is enumerated from right to left
- After you have enumerated each digit of the given code:
 1. Add up the digits in the even positions
 2. Multiply the result of the previous step by three
 3. Add up the digits in the odd positions
 4. Add up the results obtained in steps two and three
 5. Subtract the upper multiple of 10 from the result obtained in step 4
- If the result of the four step is a multiple of ten, the check-digit will be equal to zero, otherwise the check-digit will be the result of the fifth step



Calculate barcode check-digit using T-SQL

- One possible solution could be implemented by a user-defined scalar-valued function that receives an EAN code as input, and gives out the check-digit calculated for the given code
- In reading the algorithm step-by-step, the most natural and obvious solution is procedural.. Let's try!!





DEMO

Dynamic products stock level

The dynamic product stock level is requested by you boss to justify the **warehouse stock level** of a certain product on a certain date

ProductID	TransactionID	TransactionType	TransactionDate	sQuantity	StockLevel
1	104764	P	2007-09-05	3	3
1	106415	P	2007-09-12	3	6
1	107547	P	2007-09-16	3	9
1	109947	P	2007-09-26	3	12
1	113417	P	2007-10-03	3	15
1	115439	P	2007-10-10	3	18
1	117122	P	2007-10-17	3	21
1	124939	P	2007-11-07	2	23
1	124951	S	2007-12-07	-8	15



Procedural Thinking

When you image a running total, as a first time, you think a dynamic calculation that increase, for each row, a temporary value variable

	ProductID	TransactionID	TransactionType	TransactionDate	Quantity	StockLevel
1	1	104764	P	2007-09-05 00:00:00.000	3	3
2	1	106415	P	2007-09-12 00:00:00.000	3	6
3	1	107547	P	2007-09-16 00:00:00.000	3	9
4	1	109947	P	2007-09-26 00:00:00.000	3	12
5	1	113417	P	2007-10-03 00:00:00.000	3	15
6	1	115439	P	2007-10-10 00:00:00.000	3	18
7	1	117122	P	2007-10-17 00:00:00.000	3	21



Set-based Thinking

In this example there are, at the same time, seven windows, each row in the result set have the own window

	ProductID	TransactionID	TransactionType	TransactionDate	Quantity	StockLevel
1	1	104764	P	2007-09-05 00:00:00.000	3	3
2	1	106415	P	2007-09-12 00:00:00.000	3	6
3	1	107547	P	2007-09-16 00:00:00.000	3	9
4	1	109947	P	2007-09-26 00:00:00.000	3	12
5	1	113417	P	2007-10-03 00:00:00.000	3	15
6	1	115439	P	2007-10-10 00:00:00.000	3	18
7	1	117122	P	2007-10-17 00:00:00.000	3	21



The diagram illustrates seven separate window functions for ProductID 1. Each window starts at a different StockLevel value (3, 6, 9, 12, 15, 18, 21) and increases by 3 for each transaction. This represents a set-based approach where each row in the result set has its own window.

Dynamic products stock level: Solution 1

- Traditional Set-Based solution with joins
- In certain data distribution it can't be well optimized

```
-- Traditional set-based solution with joins
SELECT
    T.ProductID
    ,T.TransactionDate
    ,T.TransactionType
    ,CASE (T.TransactionType)
        WHEN 'S' THEN (T.Quantity * -1)
        ELSE (T.Quantity)
    END AS Quantity
    ,SUM(
        CASE (T1.TransactionType)
            WHEN 'S' THEN (T1.Quantity * -1)
            ELSE (T1.Quantity)
        END
    ) AS StockLevel
FROM
    Production.TransactionHistory AS T
JOIN
    Production.TransactionHistory AS T1
    ON (T.ProductID = T1.ProductID)
    AND (T1.TransactionID <= T.TransactionID)
GROUP BY
    T.ProductID
    ,T.TransactionDate
    ,T.TransactionType
    ,T.Quantity
    ,T.TransactionID
ORDER BY
    T.ProductID, T.TransactionID;
GO
```



Dynamic products stock level: Solution 2

- Cursor-based solution
- Very verbosity

```
-- Cursor-based solution
BEGIN
DECLARE
    @StockLevelTab AS TABLE
    (
        ProductID INTEGER NOT NULL
        ,TransactionID INTEGER NOT NULL
        ,TransactionType NCHAR(1) NOT NULL
        ,TransactionDate DATETIME NOT NULL
        ,Quantity INTEGER NOT NULL
        ,StockLevel INTEGER NOT NULL
    );

DECLARE
    @ProductID INTEGER
    ,@TransactionID INTEGER
    ,@PrevProductID INTEGER
    ,@Quantity INTEGER
    ,@StockLevel BIGINT
    ,@TransactionDate DATETIME
    ,@TransactionType NCHAR(1);

-- Declare cursor
DECLARE StockByProduct CURSOR FAST_FORWARD FOR
SELECT
    ProductID
    ,TransactionID
    ,CASE (TransactionType)
        WHEN 'S' THEN (quantity * -1)
        ELSE (Quantity)
    END
    ,TransactionDate
    ,TransactionType
FROM
    Production.TransactionHistory
ORDER BY
    ProductID
    ,TransactionID;

OPEN StockByProduct;

FETCH NEXT FROM StockByProduct INTO @ProductID, @TransactionID, @Quantity, @TransactionDate, @TransactionType;

SELECT @PrevProductID = @ProductID, @StockLevel = 0;

WHILE (@@FETCH_STATUS = 0)
BEGIN
    IF (@PrevProductID <> @ProductID)
        SELECT @PrevProductID = @ProductID, @StockLevel = 0;

    SET @StockLevel = @StockLevel + @Quantity;

    INSERT INTO @StockLevelTab
    (
        ProductID
        ,TransactionID
        ,TransactionDate
        ,TransactionType
        ,Quantity
        ,StockLevel
    )
    VALUES
    (
        @ProductID
        ,@TransactionID
        ,@TransactionDate
        ,@TransactionType
        ,@Quantity
        ,@StockLevel
    );

    FETCH NEXT FROM StockByProduct
    INTO @ProductID, @TransactionID,
        @Quantity, @TransactionDate, @TransactionType;
END

CLOSE StockByProduct;

DEALLOCATE StockByProduct;

SELECT * FROM @StockLevelTab ORDER BY ProductID, TransactionID;
GO
```



Dynamic products stock level: Set-based solution

- SUM quantity OVER each partition of ProductID ordered by TransactionID
- For each row: consider all previous rows and current row

```
-- Dynamic products stock level
SELECT
    ProductID
    ,TransactionID
    ,TransactionType
    ,TransactionDate
    ,CASE (TransactionType)
        WHEN 'S' THEN (Quantity * -1)
        ELSE (Quantity)
    END AS Quantity
    ,SUM(CASE (TransactionType)
        WHEN 'S' THEN (Quantity * -1)
        ELSE (Quantity)
    END)
    OVER(PARTITION BY ProductID
        ORDER BY TransactionID
        ROWS BETWEEN UNBOUNDED PRECEDING
            AND CURRENT ROW) AS StockLevel
FROM
    Production.TransactionHistory
ORDER BY
    ProductID, TransactionID;
GO
```





Demo

Links

- UGISS
 - www.ugiss.org
- SQL Saturday
 - www.sqlsaturday.com
- 24 Hours of PASS
 - www.pass.org/AttendanEvent/OnlineEvents/24HoursofPASS.aspx



Books

- By Itzik Ben-Gan
 - SQL Server T-SQL Fundamentals
 - tsql.solidq.com/books/tf3/
 - T-SQL Querying
 - tsql.solidq.com/books/tq3/
- SQL Server MVP Deep Dives Volume 2
 - www.manning.com/delaney



Ask your Questions





Thank you very much for attending
SQL Saturday Sardegna 2019!

