

Using Window Aggregate Functions



Kathi Kellenberger

@aunkathi | aunkathisql.com

Window Aggregates



Introduced in 2005

Your favorite aggregate functions

Not an aggregate query

No GROUP BY

Don't lose detail

Supported Aggregate Functions

SUM

AVG

COUNT

COUNT_BIG

MIN

MAX

CHECKSUM_AGG

STDEV

STDEVP

VAR

VARP

Syntax

AggFunction(<expression>) OVER()

COUNT(*) OVER()

AggFunction(<expression>) OVER(PARTITION BY <expression>)

COUNT(*) OVER(PARTITION BY CustomerID)

~~AggFunction(<expression>) OVER(ORDER BY <expression>)~~

Aggregate Example

CustID	OrderID	Total	SUM(Total) OVER()
1	101	100	
2	102	40	
2	103	11	
3	104	432	
1	105	2000	
1	106	300	
4	107	674	
5	108	76	
4	109	234	
4	110	889	
5	111	234	

Aggregate Example

CustID	OrderID	Total	SUM(Total) OVER()
1	101	100	4990
2	102	40	4990
2	103	11	4990
3	104	432	4990
1	105	2000	4990
1	106	300	4990
4	107	674	4990
5	108	76	4990
4	109	234	4990
4	110	889	4990
5	111	234	4990

Aggregate Example

CustID	OrderID	Total	SUM(Total) OVER(PARTITION BY CustID)
1	101	100	
1	105	2000	
1	106	300	
2	102	40	
2	103	11	
3	104	432	
4	107	674	
4	109	234	
4	110	889	
5	108	76	
5	111	234	

Aggregate Example

CustID	OrderID	Total	SUM(Total) OVER(PARTITION BY CustID)
1	101	100	2400
1	105	2000	2400
1	106	300	2400
2	102	40	51
2	103	11	51
3	104	432	432
4	107	674	1797
4	109	234	1797
4	110	889	1797
5	108	76	310
5	111	234	310

Real World Example

Percent of Sales

Special Situations

Aggregate Queries

Distinct

Aggregate Example

CustID	OrderID	Total
1	101	100
1	105	2000
1	106	300
2	102	40
2	103	11
3	104	432
4	107	674
4	109	234
4	110	889
5	108	76
5	111	234

GROUP BY CustID

CustID

1

2

3

4

5

Σ

Any column in the **SELECT LIST** or **ORDER BY** clause must appear in the **GROUP BY** or be part of an aggregate expression

Add Window Aggregate to Aggregate Query

Custom Functions

```
CREATE FUNCTION dbo.AddTwoIntegers
    (@FirstNumber INT, @SecondNumber INT)
RETURNS INT AS
BEGIN
    RETURN @FirstNumber + @SecondNumber;
END
GO
SELECT dbo.AddTwoIntegers(2, 2);
```

Custom Aggregate Functions



CLR integration in 2005

Any .NET language

User defined aggregate functions

Can be used as window functions

Performance



Traditional methods may be better

Worktables in tempdb

Pre-aggregate if possible

Often a “good-enough” solution

Performance Testing



CTE, subquery, and window aggregate

Covering index

One, two, and three functions

No results displayed

Cold cache

Beefy laptop!

Performance Comparison Query: CTE

```
WITH Totals AS (  
    SELECT ProductID, SUM(ActualCost) AS SubTotal,  
           AVG(ActualCost) AS AvgCost,  
           COUNT(*) AS ProdSaleCount  
    FROM bigTransactionHistory  
    GROUP BY ProductID)  
SELECT TransactionID, TH.ProductID  
FROM bigTransactionHistory TH  
JOIN Totals T ON TH.ProductID = T.ProductID;
```

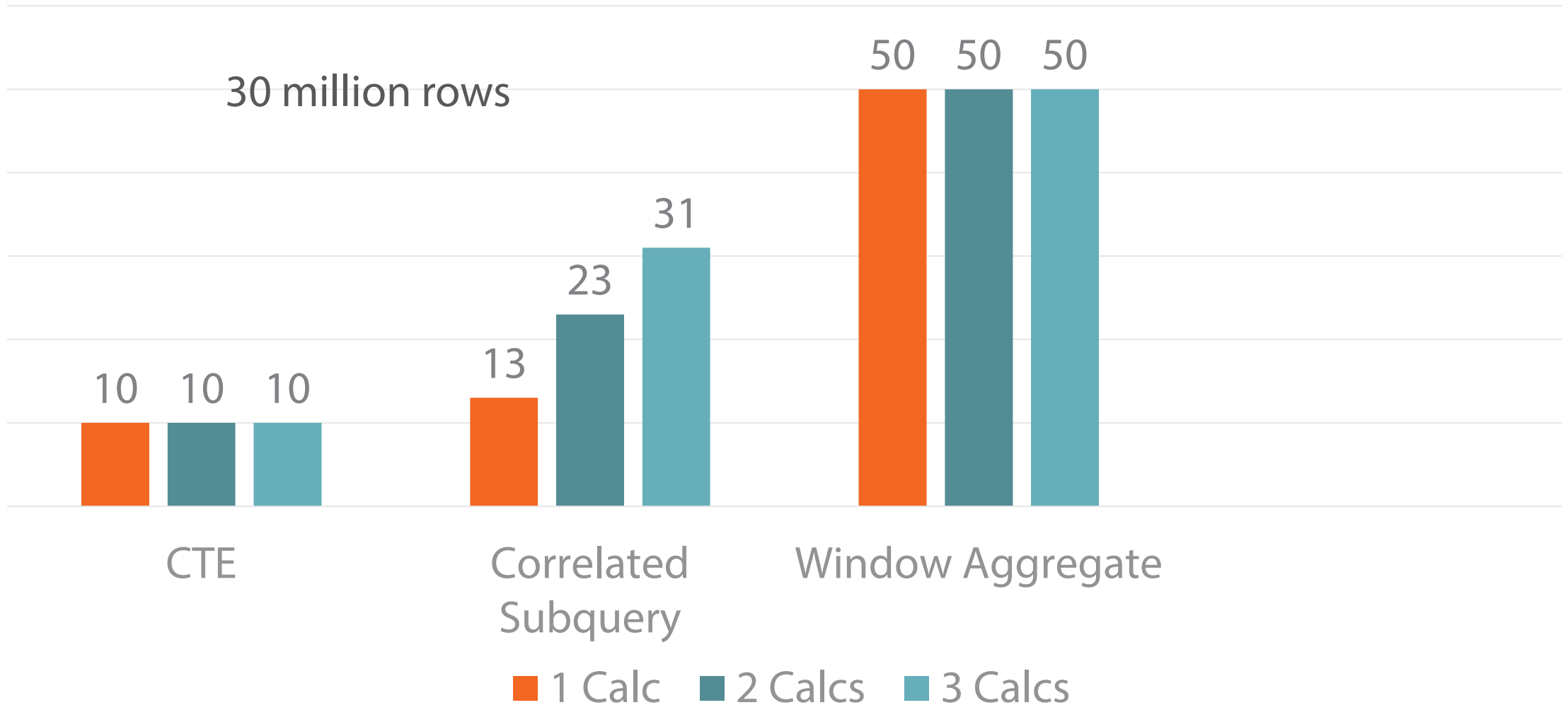
Performance Comparison: Subquery

```
SELECT TransactionID, ProductID,  
    (SELECT SUM(ActualCost) FROM BigTransactionHistory A  
    WHERE A.ProductID = B.ProductID) AS SubTotal,  
    (SELECT AVG(ActualCost) FROM BigTransactionHistory A  
    WHERE A.ProductID = B.ProductID) AS AvgCost,  
    (SELECT COUNT(*) FROM BigTransactionHistory A  
    WHERE A.ProductID = B.ProductID) AS ProdSaleCount  
FROM BigTransactionHistory B;
```

Performance Comparison: Window Aggregate

```
SELECT TransactionID, ProductID,  
       SUM(ActualCost) OVER(PARTITION BY ProductID)  
       AS SubTotal,  
       AVG(ActualCost) OVER(PARTITION BY ProductID)  
       AS AvgCost,  
       COUNT(*) OVER(PARTITION BY ProductID) AS ProdSaleCount  
FROM bigTransactionHistory;
```

Run Time in Seconds



Course Overview



Functions added in 2005 and 2012

Real world examples

Performance tuning

Coming up next: Framing