

Implementing Partitioned Views



Gerald Britton

IT SOLUTIONS DESIGNER, SQL SERVER SPECIALIST

@GeraldBritton www.linkedin.com/in/geraldbritton



```
CREATE VIEW ViewName
```

```
AS
```

```
SELECT col1, col2, col3, ...
```

```
FROM Table1
```

```
UNION ALL
```

```
SELECT col1, col2, col3, ...
```

```
FROM Table2
```

```
UNION ALL
```

```
SELECT col1, col2, col3, ...
```

```
FROM Table3
```

◀ Normal CREATE header

◀ SELECT data from first table

◀ UNION ALL

◀ SELECT data from second table

◀ UNION ALL

◀ SELECT data from third table



SELECT Statements

All columns should be included

Columns in same position should be the same type

CHECK constraint on at least one column
- Partitioning column

A column can only appear once in each select list



Partitioning Column Requirements



Part of the PRIMARY KEY



Cannot be computed, identity, default or timestamp (rowversion) column



Only one partitioning constraint



No restrictions on updateability



Requirements on Member Tables

Local or remote

A table can
appear only once

No indexes on
computed
columns

Primary keys on
same columns

Same ANSI
padding settings



Demo



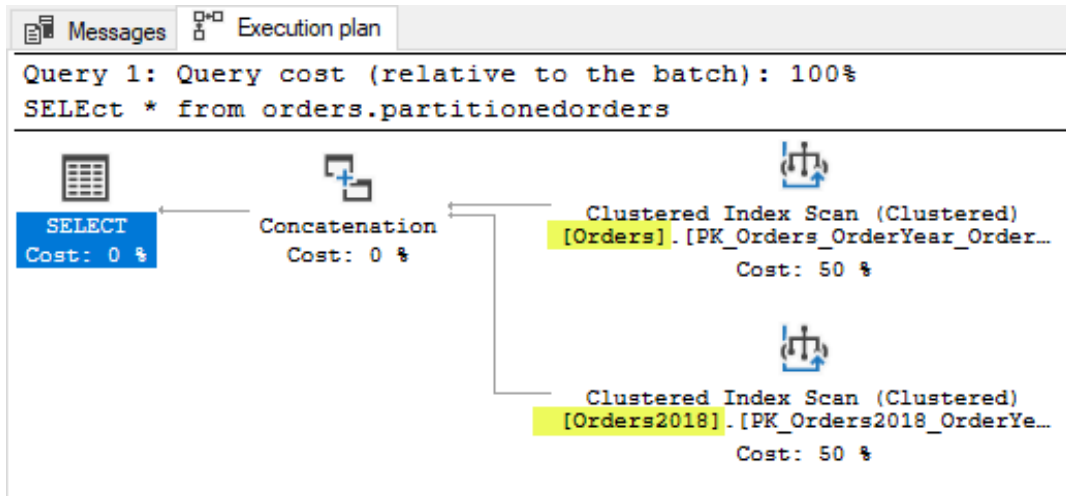
Keep current and older orders separated

Add a partitioning column

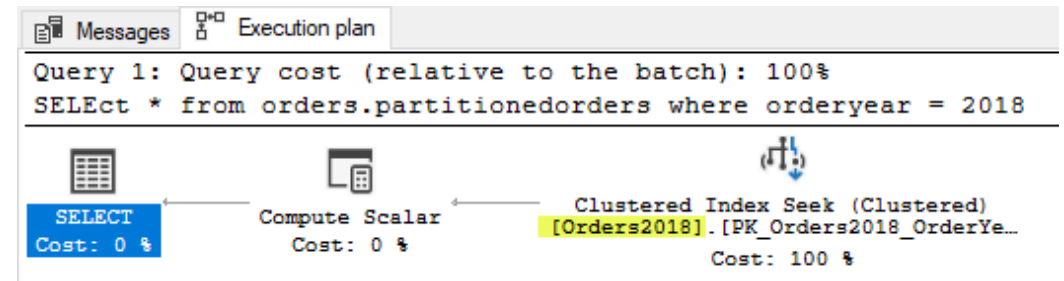
Create a partitioned view



Different Execution Plans



SELECT * FROM
Orders.PartitionedOrders



SELECT * FROM
Orders.PartitionedOrders
WHERE OrderYear = 2018;



Practical Notes

Complicated?

Design for partitioning

Maintenance

A simpler way?

- Partitioned tables



Updating Partitioned Views



INSERT statement must supply values for all the columns in the view



Cannot use the DEFAULT keyword



Values for the partitioning column must satisfy constraints



Cannot update identity or timestamp columns



Summary



Motivations

Building partitioned views

Requirements and restrictions

Partitioning column constraint

Pruned execution plans

Bobs Shoes orders by year

Complications

Modifying partitioned views

Distributed, partitioned views

