Indexed View in SQL Server

Red Devilic

LOGO

Contents

LOGO

1.Introduction of Indexed View

LOGO

- What is Indexed View
- Indexed View’s Properties
- Benefits of using Indexed View

What is Indexed View

LOGO

- What is View ?
- A *view* that has one or many indexes is called Indexed View.
- In Oracle – its name is Materialized View.

Indexed View's Properties

LOGO

- Is a feature in all versions of SQL Server 2000, 2005, 2008 and 2011.
- Based on edition (Developer, Enterprise) the query processor can use an indexed view to solve queries that structurally match the view. In other edition, you must use NOEXPAND hint.
- Unlike regular View, Indexed View stores data.
- It is self-updating when base table(s) changes.

Benefits of using Indexed View

LOGO

- Indexed views can increase query performance in the following ways:
  - Aggregations can be precomputed and stored in the index to minimize expensive computations during query execution.
  - Tables can be prejoined and the resulting data set stored.
  - …

Benefits of using Indexed View

LOGO

- Joins and aggregations of large tables.
- Repeated patterns of queries.
- Repeated aggregations on the same or overlapping sets of columns.
- Repeated joins of the same tables on the same keys.
- Combinations of the above.

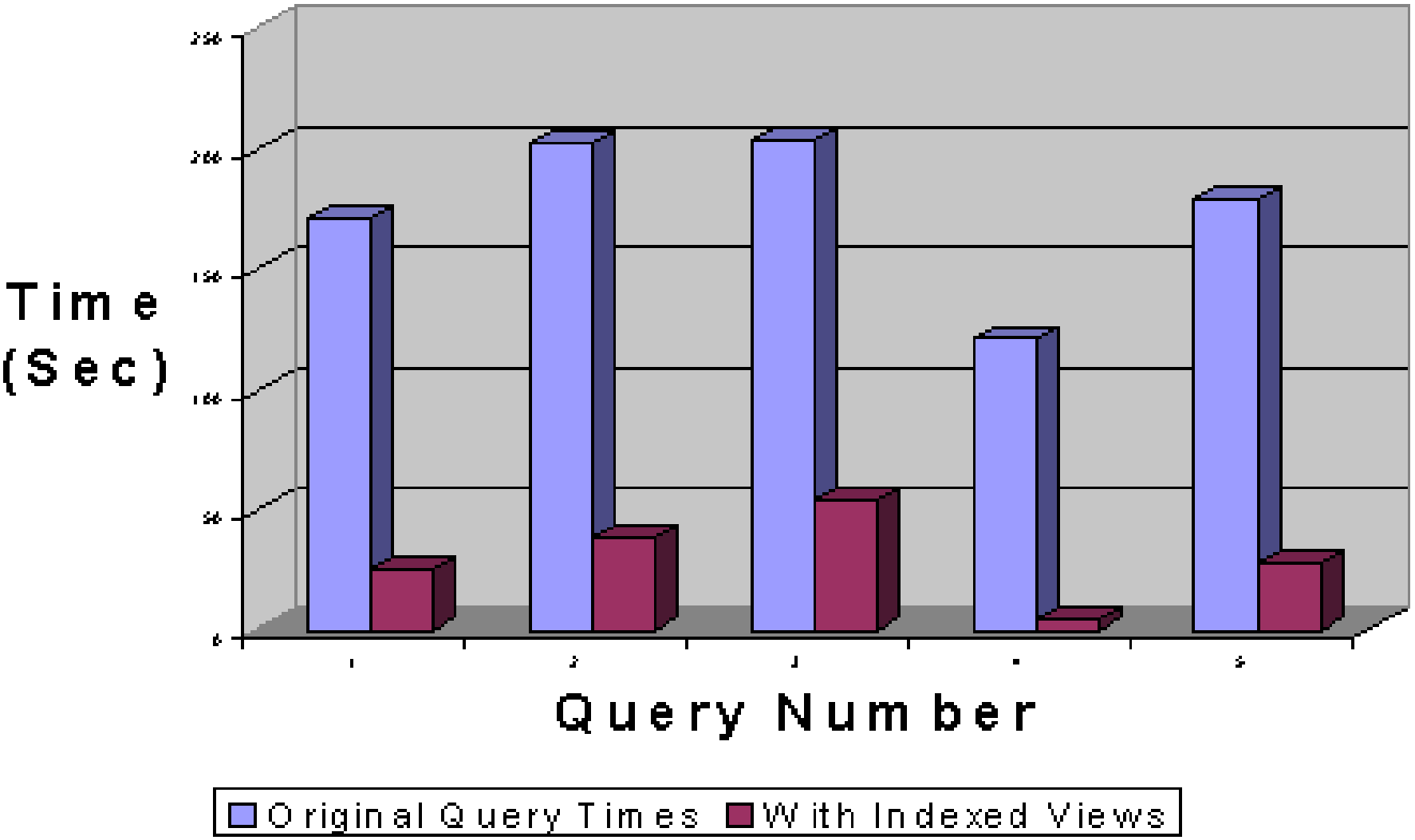
Benefits of using Indexed View

LOGO

- Applications that benefit from the implementation of indexed views include:
  - Decision support workloads.
  - Data marts.
  - Data warehouses.
  - Online analytical processing (OLAP) stores and sources.
  - Data mining workloads.

Response Time Comparison

LOGO



2.Designing and Creating Indexed View

LOGO

- Database Tuning Advisor
- Design Phase Requirements
- Requirements of creating Indexed View

Database Tuning Advisor

LOGO

- Is a SQL Server feature that helps database administrators tune their physical database design.

- Using Database Engine Tuning Advisor enhances an administrator's ability to determine the combination of indexes, indexed views, and partitioning strategies that optimize the performance of the typical mix of queries executed against a database.

Design Phase Requirements

LOGO

- The view and all tables referenced in the view, must be in the same database and have the same owner.
- A unique clustered index must be created before any other indexes can be created on the view.
- Certain SET options
- The view must be created using schema binding, and any user-defined functions referenced in the view must also be created with the SCHEMABINDING option.

Requirements of creating Indexed View

LOGO

- The view must meet the following requirements:
  - The view must be created using the WITH SCHEMABINDING option.
  - Tables must be referenced by the view using two-part names (schemaname.tablename).
  - User-defined functions must be referenced by the view using two-part names (schemaname.functionname).
  - More: MSDN & BOL.

3. Examples, Solutions & Performance

LOGO

- Setup Environments
  - SET ANSI_NULLS ON
  - SET ANSI_PADDING ON
  - SET ANSI_WARNINGS ON
  - SET CONCAT_NULL_YIELDS_NULL ON
    SET NUMERIC_ROUNDABORT OFF
  - SET QUOTED_IDENTIFIER ON
  - SET ARITHABORT ON
- Sample DB AdventureWorks in SQL Server 2005 SP3

3. Examples, Solutions & Performance

LOGO

DEVILIC\RED.A...SQLQuery1.sql* | Summary

```
SELECT TOP 5 ProductID, Sum(UnitPrice*OrderQty) -
     Sum(UnitPrice*OrderQty*(1.00-UnitPriceDiscount)) AS Rebate
FROM Sales.SalesOrderDetail
GROUP BY ProductID
ORDER BY Rebate DESC
```

Messages | Execution plan

Query 1: Query cost (relative to the batch): 100%

SELECT TOP 5 ProductID, Sum(UnitPrice*OrderQty) - Sum(UnitPrice*OrderQty*(1.00-UnitPriceDiscount)) AS Rebate FROM Sales.SalesOrderDetail GRO…

Clustered Index Scan

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated I/O Cost	0.915718
Estimated CPU Cost	0.133606
Estimated Operator Cost	1.04932 (63%)
Estimated Subtree Cost	1.04932
Estimated Number of Rows	121317
Estimated Row Size	29 B
Ordered	False
Node ID	4

Object

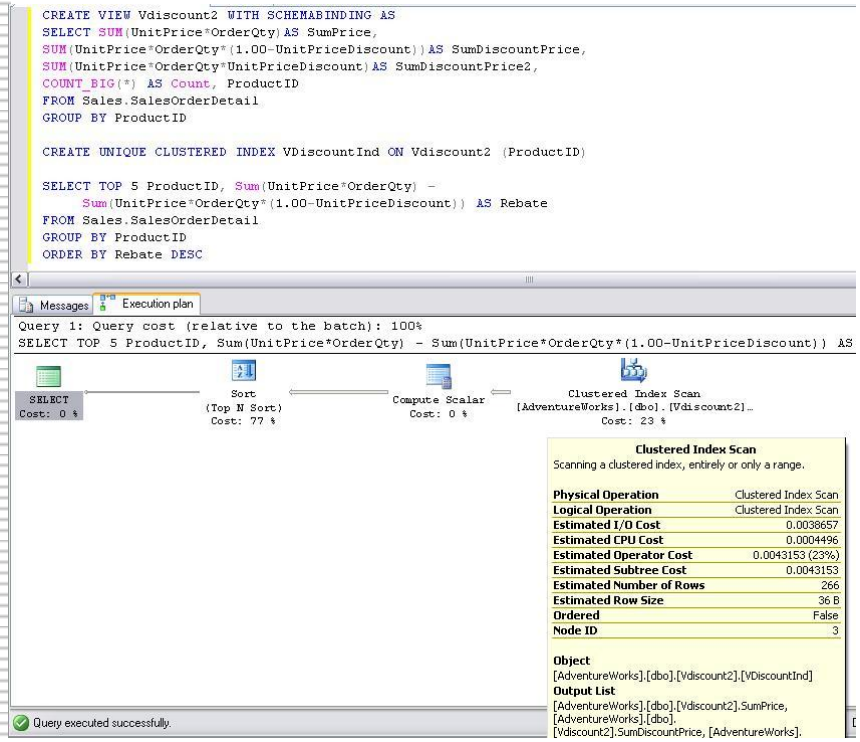
[AdventureWorks].[Sales].[SalesOrderDetail].
[PK_SalesOrderDetail_SalesOrderID_SalesOrderDetailID]

Output List

[AdventureWorks].[Sales].[SalesOrderDetail].OrderQty,
[AdventureWorks].[Sales].[SalesOrderDetail].ProductID,
[AdventureWorks].[Sales].[SalesOrderDetail].UnitPrice,
[AdventureWorks].[Sales].
[SalesOrderDetail].UnitPriceDiscount

3. Examples, Solutions & Performance

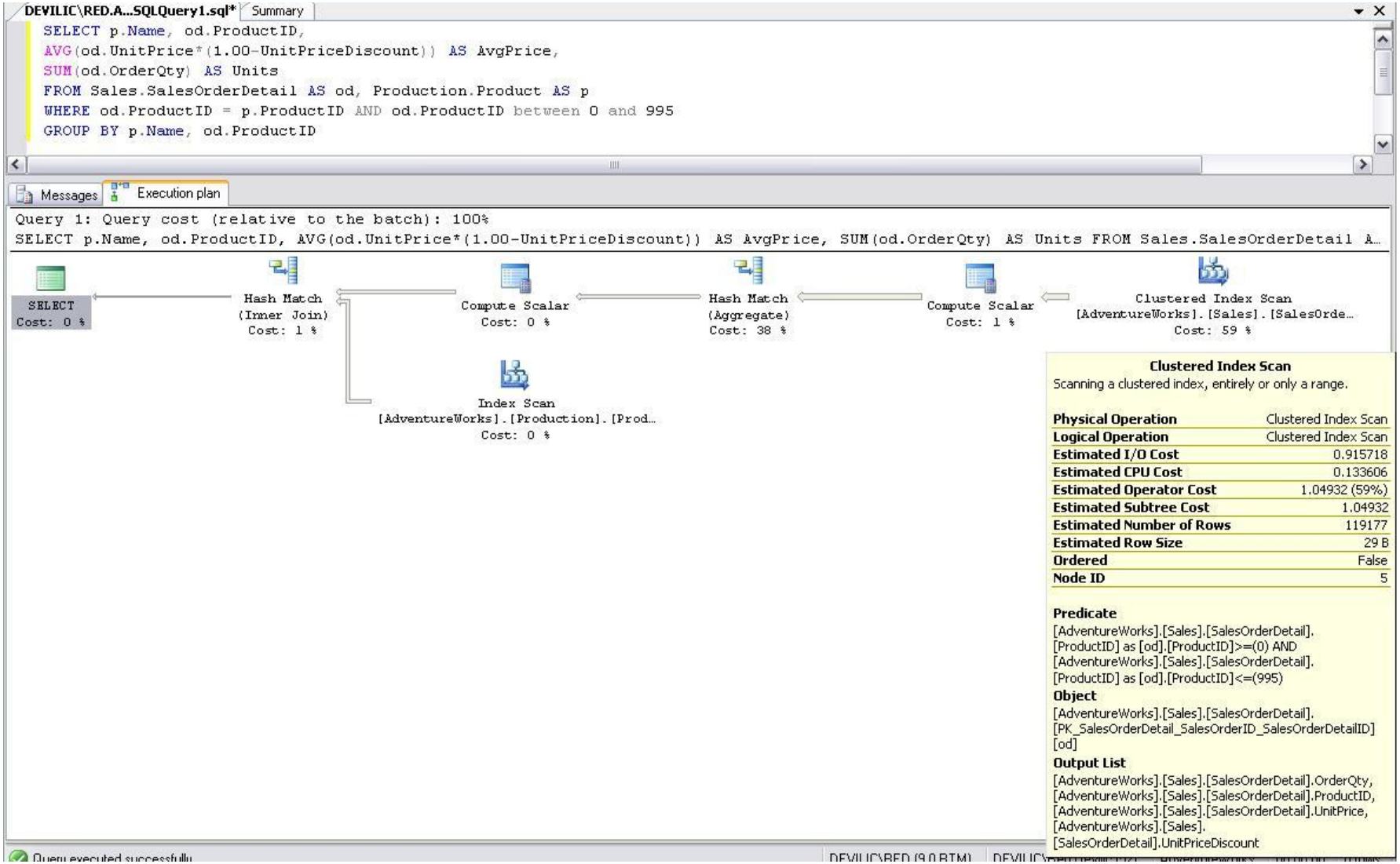
LOGO



- A Clustered Index Scan on Sales.SalesOrderDetail table.
- A Hash Match/Aggregate operator.
- A TOP 5 sort operator based on the ORDER BY clause.

3. Examples, Solutions & Performance

LOGO



3. Examples, Solutions & Performance

LOGO

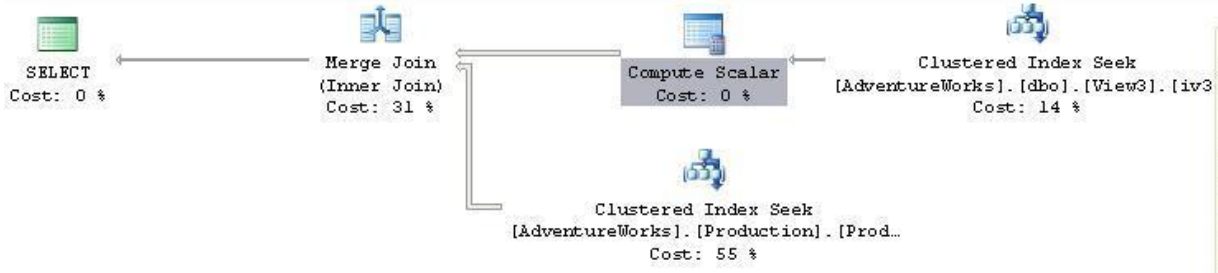
DEVILIC\RED.A...SQLQuery1.sql* | Summary

```
CREATE VIEW View3 WITH SCHEMABINDING AS
SELECT ProductID, SUM(UnitPrice*(1.00-UnitPriceDiscount)) AS Price,
COUNT_BIG(*) AS Count, SUM(OrderQty) AS Units
FROM Sales.SalesOrderDetail
GROUP BY ProductID

CREATE UNIQUE CLUSTERED INDEX iv3 ON View3 (ProductID)

SELECT p.Name, od.ProductID,
AVG(od.UnitPrice*(1.00-UnitPriceDiscount)) AS AvgPrice,
SUM(od.OrderQty) AS Units
FROM Sales.SalesOrderDetail AS od, Production.Product AS p
WHERE od.ProductID = p.ProductID AND od.ProductID between 0 and 995
GROUP BY p.Name, od.ProductID
```

Messages | Execution plan

Query 1: Query cost (relative to the batch): 100%

SELECT p.Name, od.ProductID, AVG(od.UnitPrice*(1.00-UnitPriceDiscount)) AS AvgPrice, SUM(od.OrderQty) AS Units FROM Sales.Sale



Clustered Index Seek

Scanning a particular range of rows from a clustered index.

Physical Operation	Clustered Index Seek
Logical Operation	Clustered Index Seek
Estimated I/O Cost	0.003125
Estimated CPU Cost	0.0001833
Estimated Operator Cost	0.0033083 (14%)
Estimated Subtree Cost	0.0033083
Estimated Number of Rows	23.94
Estimated Row Size	40 B
Ordered	True
Node ID	4

Object
[AdventureWorks].[dbo].[View3].[iv3]

Output List
[AdventureWorks].[dbo].[View3].ProductID,
[AdventureWorks].[dbo].[View3].Price,
[AdventureWorks].[dbo].[View3].Count,
[AdventureWorks].[dbo].[View3].Units

Seek Predicates
Start Range: [AdventureWorks].[dbo].[View3].ProductID

Query executed successfully.

DEVILIC

4. Indexed View's FAQ

LOGO

- Why use Indexed View? Can I create a summary table instead of Indexed View?
- Why isn't my indexed view being picked up by the query optimizer for use in the query plan?
  - Version of SQL Server
  - Requirements not match.
- Disadvantage of Indexed View

Thank You !

Red Devilic – RED.DEVILIC@gmail.com

LOGO