

Columnstore Index– Part 1

“Achieve Blazing Fast Performance”

By

Amit R S Bansal

Director, Principal Consultant & Trainer

www.PeoplewareIndia.com

www.amitbansal.net

MVP, MCT, MCTS, etc..

Follow me on twitter now @A_Bansal

Thanks to: Eric Hanson, Susan Price & Microsoft SQL Product Team

Columnstore Index– Part 1

“Achieve Blazing Fast Performance”

Sponsored by

By



www.FaceBook.com/eDominer

PEOPLEWARE INDIA
TECHNOLOGY TRAINING SPECIALISTS

www.FaceBook.com/PeoplewareIndia

Amit R S Bansal

- **CTO, eDominator Systems & Peopleware India**
- **Corporate Trainer/Consultant & Evangelist**
- **Conducted more than 450+ workshops on SQL Server & BI for top notch IT companies world wide**
- **Microsoft MVP for SQL Server**
- **Speaker at TechED India, TechED US & TechED Europe**
- **Speaking at SQLBITs this March in London**
- **Technical Reviewer/SME – MSL courses & certifications on SQL Server**
- **Founder & President – SQLServerGeeks.com**

About SQLServerGeeks.com

- One of the fastest growing SQL Server communities
- Blogs, Articles, Podcasts, In-person events, webcasts
- Millions of page views
- People behind SQLServerGeeks: Amit Bansal, Sarabpreet Singh, Parikshit Savjani, Abhay Chaudhary, Raunak Jhawar, Ahmed Osama, Amit Karkhanis, Vasudev Menon, Ritesh Medhe, Rakesh Mishra, Piyush Bajaj, Rahul Sharma, Satnam Singh, Bhagwan Singh
- If you want to contribute, email us at admin@SQLServerGeeks.com or visit Join Us section on the website

Agenda

- “How exactly does columnstore index improve performance?”

Columnstore index

- **Accelerates data warehouse queries**
 - Improved query execution
- **More interactive**
 - Faster query response
 - Easier data exploration
 - Better decisions
- **Easy to use**
 - Fully integrated into SQL Server
 - Fewer indexes, summary aggregates, indexed views
- **Reduces TCO**
 - Manage more data with less hardware
 - Reduces the labor cost

Columnstore index

- **Column store in SQL Server**
 - New type of index
- **New query processing mode**
 - Processes data in batches
- **Accelerates data warehousing workloads**
 - Read mostly, large data volumes, star schema
 - Often sliding window
 - Star joins, aggregations, filtering
- **SQL Server SMP relational data warehouses**

Data warehouse workload

- **Read-mostly**

- Load large amounts of data
- Append new data incrementally
- Rarely update existing data
- Often retain data for given time (e.g. 1, 3, 7 yrs)

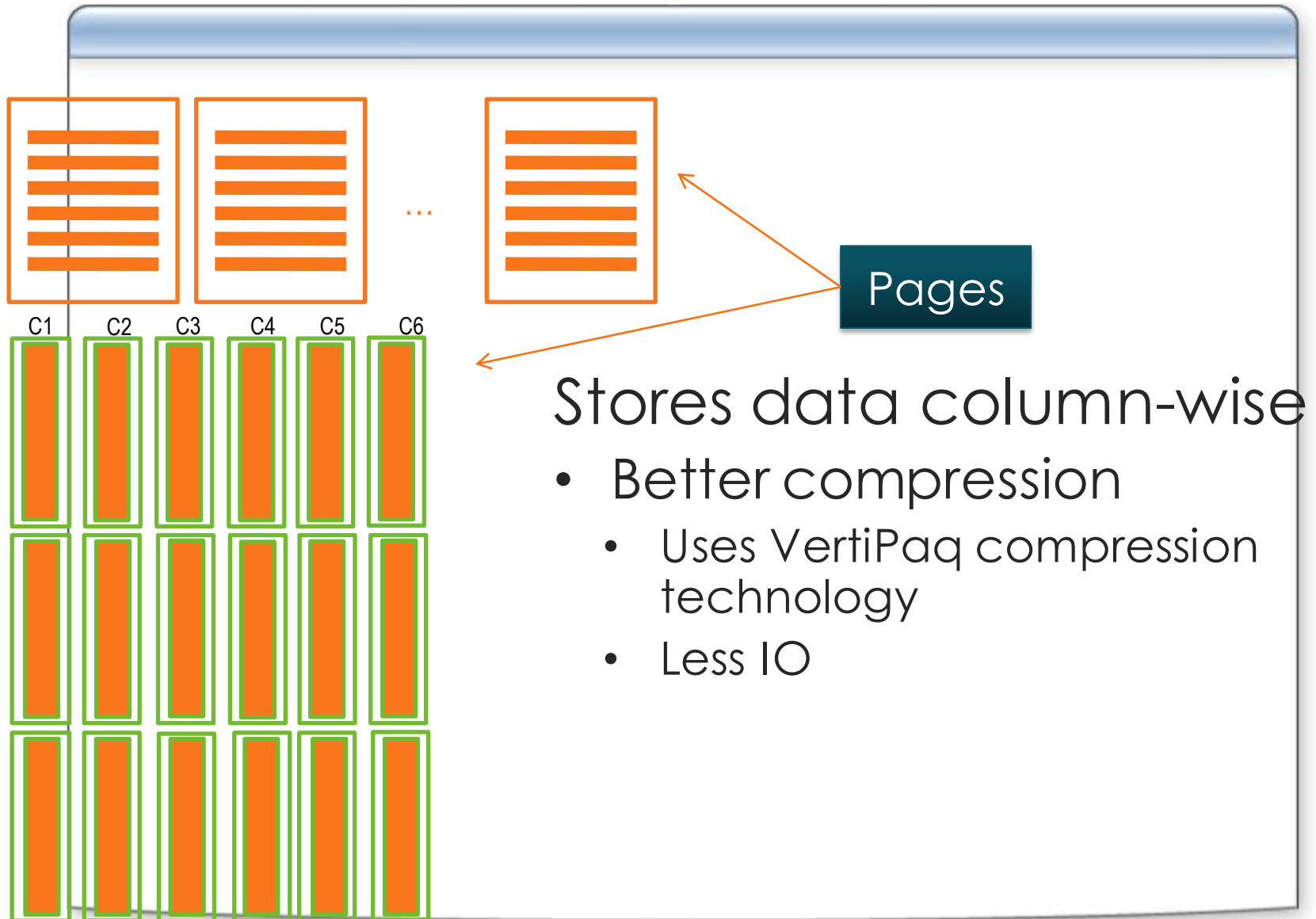
Sliding window data management

Typical Data warehouse queries

- **Process large amounts of data**
- **Reporting queries**
- **Often slow (minutes to hours)**
- **DBAs spend considerable effort**
 - Designing indexes, tuning queries
 - Building summary tables, indexed views, OLAP cubes

DEMO

Blazing fast performance (1)



Blazing fast performance (1)

| ID | Name | City | State |
|----|------|----------|-------|
| 1 | John | Seattle | WA |
| 2 | Jane | Redmond | WA |
| 3 | Jill | Redmond | OR |
| 4 | Jane | Bellevue | WA |

Row Store

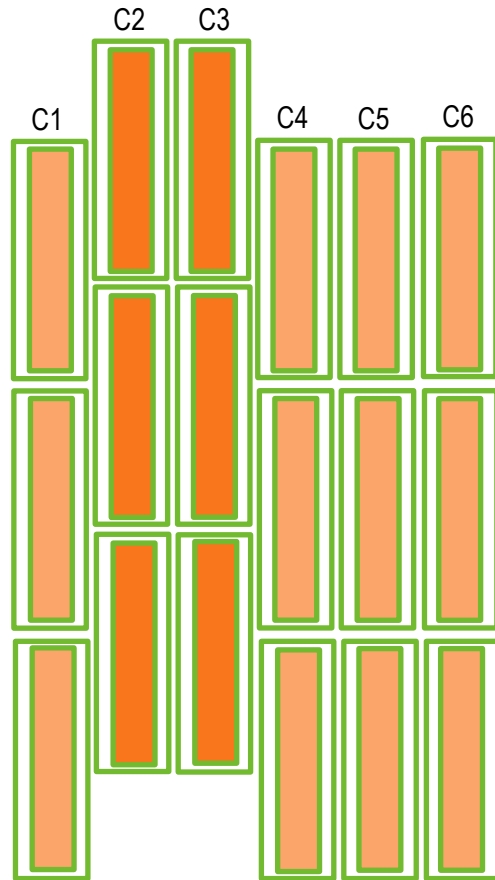
1 John Seattle WA
2 Jane Redmond WA
3 Jill Redmond OR
4 Jane Bellevue, WA

Column Store (compressed)

| | | | | |
|--|---------|---------|---------|----------|
| | 1 | 2 | 3 | 4 |
| | John | Jane | Jill | Jane |
| | Seattle | Redmond | Redmond | Bellevue |
| | WA | WA | OR | WA |

Blazing fast performance (2)

SELECT region, sum (sales) ...



Fetches only needed columns from disk

- Less IO
- Better buffer hit rates

Blazing fast performance (3)

- **Advanced query processing technology**
- **Batch mode execution of some operations**
 - Processes column data in batches
- **Compact data representation**
- **Highly efficient algorithms**
- **Better parallelism**

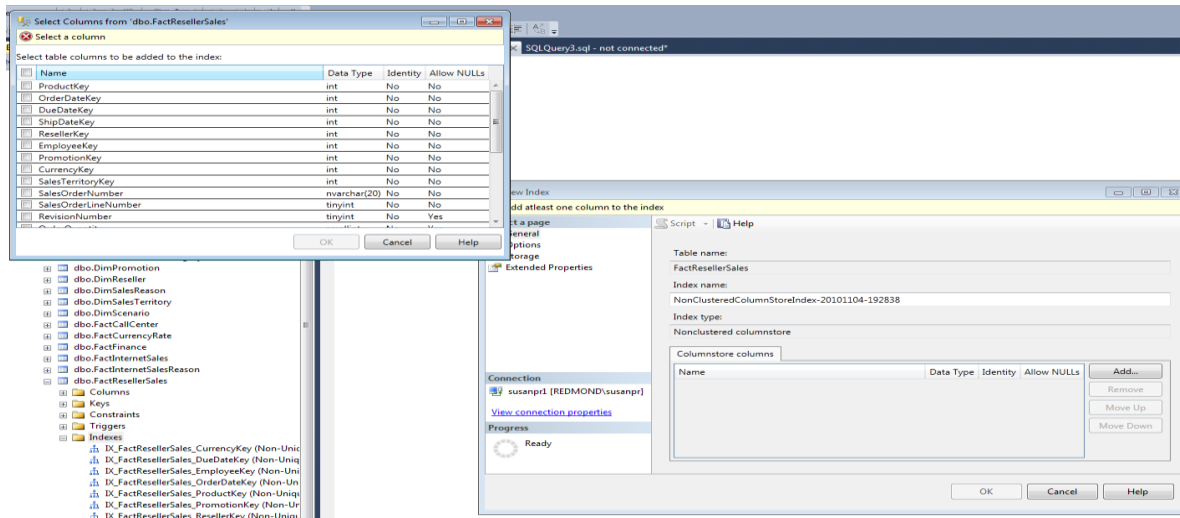
Constraints on use with other indexes & partitioning

- **Base table must be clustered B-tree or heap**
- **Columnstore index:**
 - nonclustered
 - one per table
 - must be partition-aligned
 - not allowed on indexed view
 - can't be a filtered index

Creating columnstore index

Create a columnstore index

- Create the table
- Load data into the table
- Create a non-clustered columnstore index on all, or some, columns



Constraints on use with other indexes & partitioning

- **Let the optimizer do the work**
- **Optimizer makes a cost-based decision**
 - Data access method
 - Columnstore index
 - Clustered (row-based) index
 - Nonclustered (row-based) index
 - Heap
 - Processing mode
 - Batch mode
 - Row mode

Using Hints...

- **use the columnstore index**
 - `select distinct (SalesTerritoryKey) from
dbo.FactResellerSales with (index (ncci))`
- **use a different index**
 - `select distinct (SalesTerritoryKey) from
dbo.FactResellerSales with (index (ci))`
- **ignore columnstore**
 - `select distinct (SalesTerritoryKey) from
dbo.FactResellerSales
option(ignore_nonclustered_columnstore_index)`

Data Type restrictions

- **Create index:**

- Only on common business data types

| | |
|------------|--|
| Yes | int, real, string, money, datetime, decimal <= 18 digits |
| No | decimal > 18 digits, binary, BLOB, CLR, (n)varchar(max), uniqueidentifier, datetimeoffset with precision > 2 |

- **Maintain table: limited operations**

- Can read but not update the data
- Can switch partitions in and out

- **Process queries: all read-only T-SQL queries run**

- Some queries are accelerated more than others

Query Performance restrictions

- **Outer joins**
- **Unions**
- **Consider modifying queries to hit “sweet spot”**
 - Inner joins
 - Star joins
 - Aggregation

But the table is ready only, how do I update?

DEMO

Simple guidelines

- **Workload**

- Read mostly
- Most updates append new data
- Star join queries
- Queries to scan and aggregate large data volumes

- **Workflow**

- Permits partitioning (or drop & rebuild index) to handle new data
- Typically a nightly load window

- **Build a columnstore index on large fact tables**

- **Consider a columnstore index for large dimension tables (May be)**

Columnstore index webcast – part 2

- **Segments**
- **Memory Management**
- **Compression**
- **Understanding the Execution Plan**
- **Row Mode vs Batch Mode**
- **Best Practices**

Call to Action

- Download the presentation and scripts – www.SQLServerGeeks.com/Files
- Try it out
- If you have questions, post on our forums – www.SQLServerGeeks.com/Forums
- Online resources: MSDN, TechNET, SQL Server Tech Center
- Follow me on twitter: @A_Bansal

Continue your learning...

- Be a member - www.SQLServerGeeks.com
- www.FaceBook.com/SQLServerGeeks
- Twitter @SQLServerGeeks
- Presentation & Scripts uploaded on www.SQLServerGeeks.com/files

Next steps...

- Provide your feedback
 - you shall receive a mail shortly
 - On facebook –
www.FaceBook.com/SQLServerGeeks

Connect with Amit Bansal

- www.AmitBansal.net
 - LinkedIn, FB, Twitter, GuestBook
 - Twitter: @A_Bansal
- Join my network
 - Personal Site – <http://www.amitbansal.net>
 - LinkedIn – <http://www.linkedin.com/in/AmitBansal2010>
 - Forum – www.SQLServerGeeks.com
 - FaceBook – <http://www.facebook.com/people/Amit-Bansal/525339346>
 - Twitter – http://www.twitter.com/A_Bansal
 - Blog – www.SQLServerGeeks.com

Q&A

We await your feedback on

www.FaceBook.com/SQLServerGeeks

A word from our sponsors:

“If you need advanced trainings on SQL Server and/or other Microsoft products and technologies for your team, write to enquiry@peoplewareindia.com or call +91 9007009017”

Thank you for attending this webcast ☺

Thank you ☺

for suggestions, please email at admin@SQLServerGeeks.com