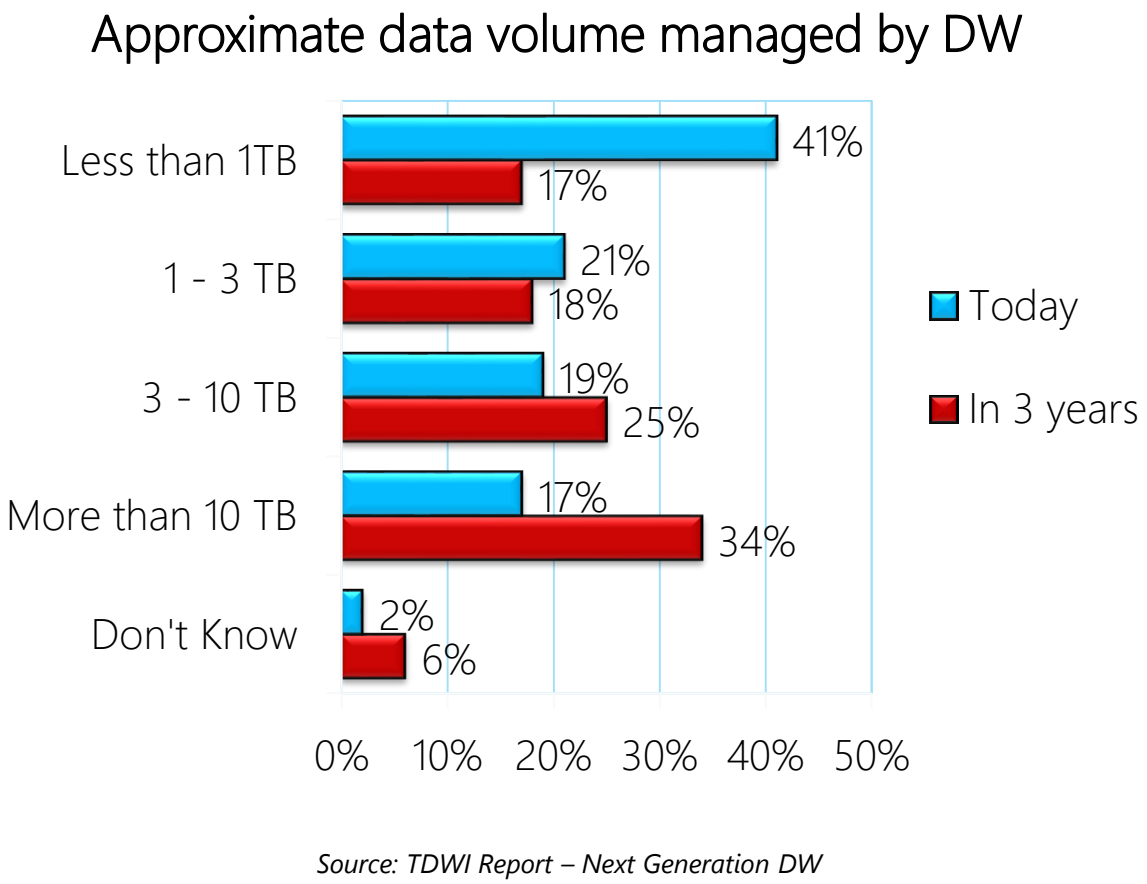
Trends in the Data Warehousing Space



Source: TDWI Report – Next Generation DW

- **Scale more:** DW systems continue to grow at a fast pace, **scalability** is a key concern, growing a system **from 10s of TBs, to 100s of TB, to PBs**
- **Performance at scale:** ability to analyze massive amounts of data while offering **interactive query response**
- **Data warehousing for masses:** drive down price per TB

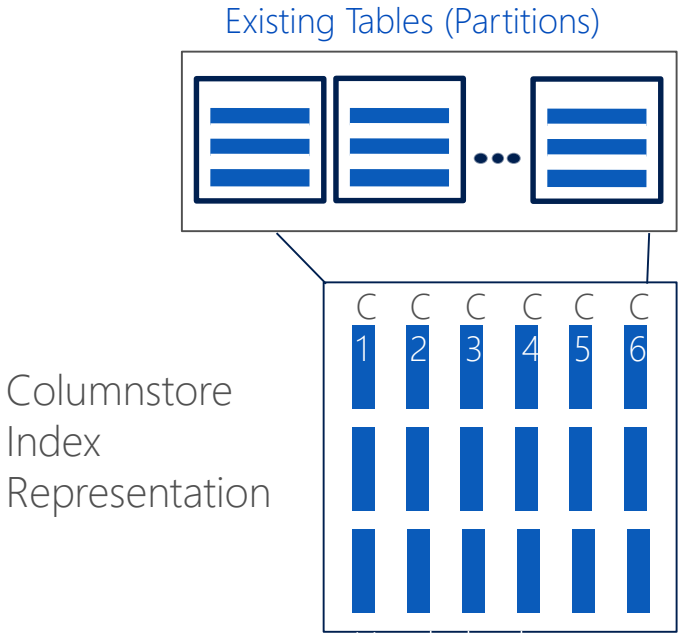
Columnstore designed to address above need

Columnstore Index

- In-memory columnstore
- Lives in both memory and disk
- Built-in to core RDBMS engine
- Customer benefits:
  - 10-100x faster
  - Reduced design effort
  - Hyper-efficient storage subsystem
  - Works on customers' existing hardware
  - Easy upgrade, easy management

"By using SQL Server 2012 In-Memory Columnstore, we were able to extract about 100 million records in **2 or 3 seconds** versus the **30 minutes required** previously. "

- Atsuo Nakajima Asst Director, Bank of Nagoya



Target Intended Workloads

Data warehouse: data written once, read many times

Queries are against large amounts of data

- Aggregating, filtering are common operations

Star & snowflake schemas are common

- Fact tables and dimensions

New data is typically appended to fact tables

- Dimension tables more likely to have “random” updates

Updates of fact tables are rare or do not exist

Traditional Storage Models
Data Stored Row-wise: Heaps, B-trees

Columnstore Index Storage Model
Data Stored Column-wise

Each page stores data from a single column

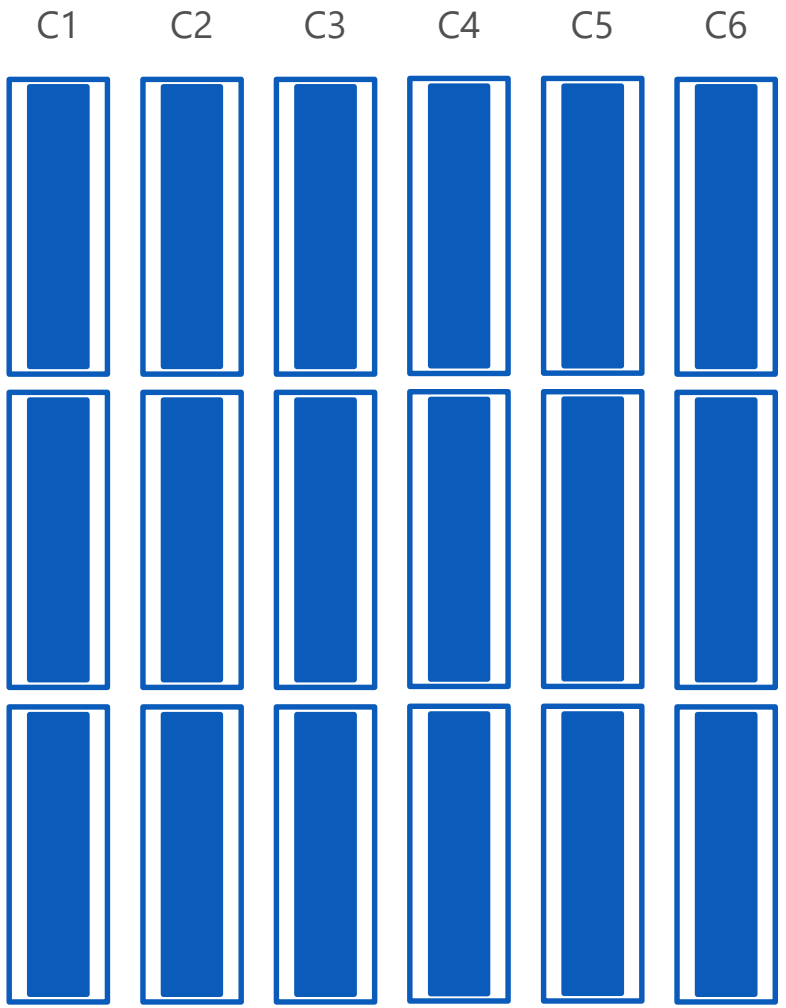
Highly compressed

More data fits in memory

Each column can be accessed independently

Fetch only columns that are needed

Can dramatically decrease I/O



Batch Mode: Improving CPU Utilization

Biggest advancement in query processing in years!

Data moves in batches through query plan operators

- Minimizes instructions per row
- Takes advantage of cache structures

Highly efficient algorithms

Better parallelism

Columnstore in SQL 2012

SQL Server 2012 Columnstore functionality

- Non-clustered Columnstore indexes
- Improved compression, compared to ROW/PAGE compression
- Improved query performance for large read scenarios

Limitations

- No DML support, no updates (data refresh)
- Only secondary, non-clustered, Columnstore indexes supported
- Poor memory management (Resource Governor was not honored, index build/rebuild, run-time)
- Limited data types support
- Limited batch operations supported

Clustered Columnstore Index

Why is clustered index important?

- Saves space used
- Simplifies management – no secondary indexes to maintain

Columnstore (and clustered Columnstore index) will be PREFERRED storage engine for DW scenarios

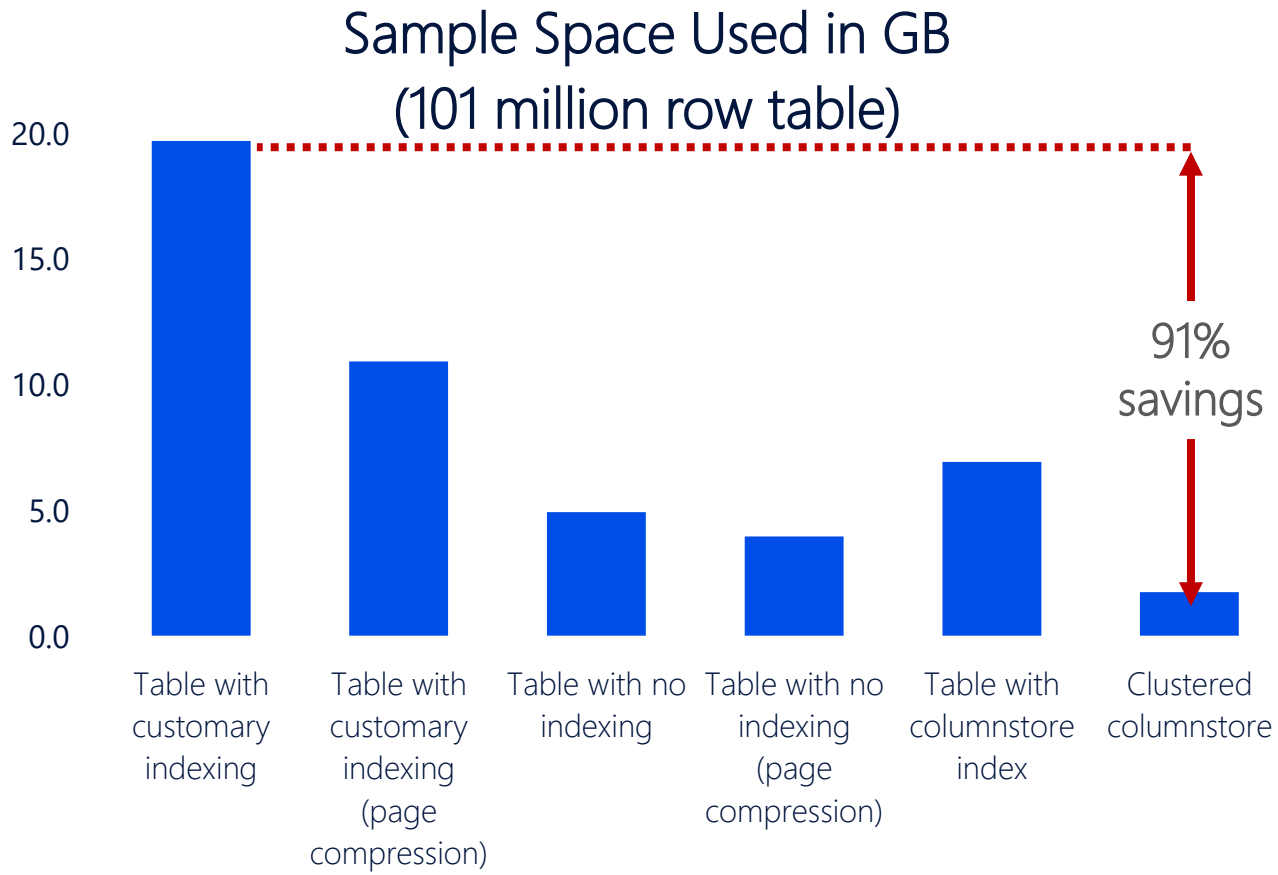
- We encourage users to either move existing tables to CCI, or start using CCI for new tables

Additional data types are supported

- High precision decimal, datetimeoffset, binary, varbinary, uniqueidentifier, etc.)
- Unsupported types: spatial, XML, max types

DDL supported

- Evolve your schema design as needed



** Space Used = Table space + Index space

Clustered Columnstore index
Key Characteristics

Available in Enterprise, Developer, and Evaluation editions

Updateable

Includes all columns in the table

Only index on the table, cannot be combined with any other indexes

Uses Columnstore compression

Columns not physically sorted. Stores data to improve compression and performance

Nonclustered Columnstore Index
Key Characteristics

No need to include all of the columns in the table

Stores a copy of the columns in the index

Is not updateable. Changes = rebuild index

Can be combined with other indexes on the table

Uses Columnstore compression

Columns not physically sorted. Stores data to improve compression and performance

Archival Compression - What's New?

Adds an additional layer of compression on top of the inherent compression used by Columnstore

Shrink on-disk database sizes by up to 27%

Compression applies per partition and can be set either during index creation or during rebuild

Use archival compression only when extra time and CPU resources to compress and retrieve the data are affordable

Columnstore Enhancements Summary

New functionalities delivered

- Clustered and updateable Columnstore index
- Columnstore archive option for data compression
- Global batch aggregation

Main benefits

- Real-time super fast data warehouse engine
- Ability to continue queries while updating without the need to drop and recreate index or partition switching
- Huge disk space saving due to compression
- Ability to compress data 5–15x using archival per-partition compression
- Better performance and more efficient (less memory) batch query processing using batch mode rather than row mode