**ColumnStore Indexes evolution from SQL Server 2012, 2014 to 2016**

**ColumnStore Indexes** were first introduced in SQL Server 2012, and this created a new way to store and retrieve the Index or Table data in an efficient manner.

ColumnStore uses **xVelocity** technology that was based on **Vertipaq engine**, this uses a new **Columnar storage** technique to store data that is highly **Compressed** and is capable of In-memory Caching and highly parallel data scanning with Aggregation algorithms.

ColumnStore or **Columnar data format** does not store data in traditional RowStore fashion, instead the data is grouped and stored as one column at a time in **Column Segments**.

The **Traditional RowStore** stores data for each row and then joins all the rows and store them in **Data Pages**, and is still the same storage mechanism for Heap and Clustered Indexes.

Thus, a **ColumnStore** increases the performance by reading **less data** that is highly **compressed**, and in **batches** from disk to memory by further reducing the **I/O**.

**–>** To know more about ColumnStore Indexes check **[**[**MSDN BoL**](https://msdn.microsoft.com/en-us/library/gg492088(v=sql.110).aspx)**]**.

**–> Let’s see the journey of ColumnStore index from SQL Server 2012 to 2016:**

We will start with **SQL Server 2012** offering for ColumnStore Indexes:

1. A Table (Heap or BTree) can have **only one NonClustered ColumnStore Index**.

2. A Table with NonClustered ColumnStore Index becomes **readonly** and cannot be updated.

3. The NonClustered ColumnStore Index uses **Segment Compression** for high compression of Columnar data.

With **SQL Sever 2014** some new features were added, like:

1. You can create **one Clustered ColumnStore Index** on a Table, and no further Indexes can be created.

2. A Table with Clustered ColumnStore Index can be **updated** with INSERT/UPDATE/DELETE operations.

3. Both Clustered & NonClustered ColumnStore Index has new Archival Compression options i.e. **COLUMNSTORE\_ARCHIVE** to further compress the data.

Now with **SQL Server 2016** new features have been added and some limitations are removed:

1. A Table will still have **one NonClustered ColumnStore Index**, but this will be **updatable** and thus the Table also.

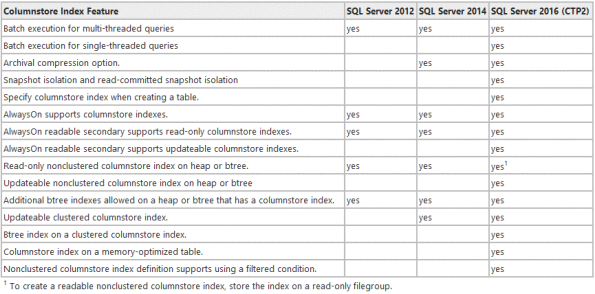
2. Now you can create a **Filtered NonClustered ColumnStore Index** by specifying a WHERE clause to it.

3. A Clustered ColumnStore Index table can have **one or more NonClustered RowStore Indexes**.

4. Clustered ColumnStore Index can now be Unique indirectly as you can create a **Primary Key** Constraint on a Heap table (and also **Foreign Key** constraints).

5. Columnar Support for **In-Memory** (Hekaton) tables, as you can create one one ColumnStore Index on top of them.

**–> Here is the full feature summary of ColumnStore Indexe evolution from SQL Server 2012 to 2016:**

[](https://sqlwithmanoj.files.wordpress.com/2015/06/sql-server-2016-columnstoreindexes.png)