http://mssqlfun.com/

Data Compression

Rohit Garg

Table of Contents

Data Compression..2

Types of Database Compression...2

Understanding Data Compression Types 2

Implementation of Data Compression 3

Impact of Compression...5

Disable Data Compression..5

Considerations while using row and page compression 6

Data Compression

Data Compression is feature of Microsoft SQL Server to reduce the size of table on the basis of Duplicates, Null & Zeroes. It's a process of reducing size of database & its objects by increasing CPU cycle and reducing I/O effort.

Types of Database Compression

- Row Compression
- Page Compression

Understanding Data Compression Types

1. Row compression - Row Compression does not change deal with data. It works on data types. Each data type has set flexibility range to insert data. Row Compression only changes the physical storage format of the data according to data type.

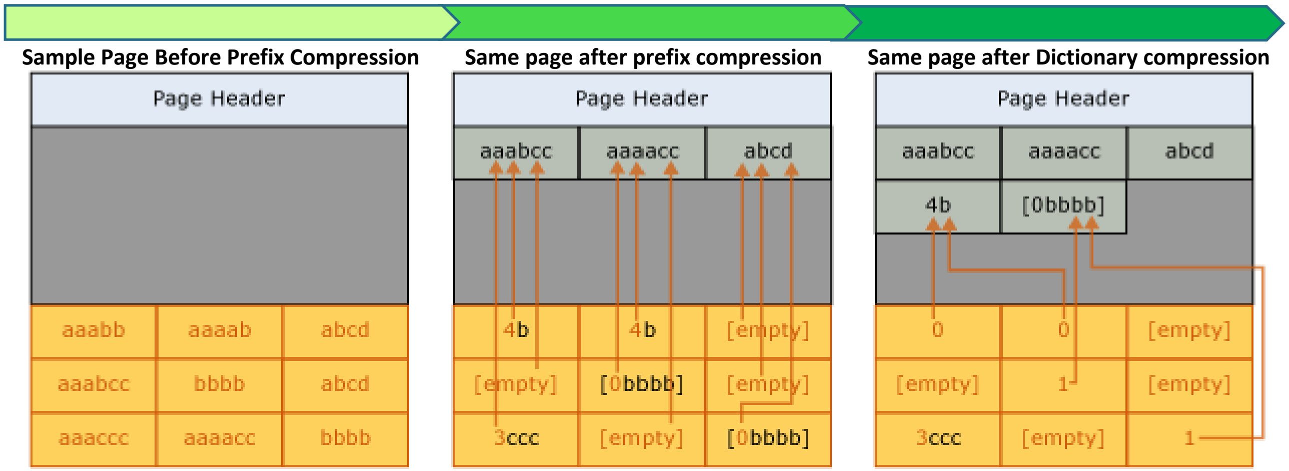
2. Page Compression - Compressing the leaf level of tables and indexes with page compression consists of three operations in the following order:

   ✓ Row compression - Row Compression will be implemented as a part of Page Compression, You need not to perform any step for this.

   ✓ Prefix compression

     - For each column, a value is identified that can be used to reduce the storage space for the values in each column.
     - A row that represents the prefix values for each column is created and stored in the compression information (CI) structure that immediately follows the page header.
     - The repeated prefix values in the column are replaced by a reference to the corresponding prefix. If the value in a row does not exactly match the selected prefix value, a partial match can still be indicated.

   ✓ Dictionary compression - After prefix compression has been completed, dictionary compression is applied. Dictionary compression searches for repeated values anywhere on the page, and stores them in the CI area. Unlike prefix compression, dictionary compression is not restricted to one column. Dictionary compression can replace repeated values that occur anywhere on a page.



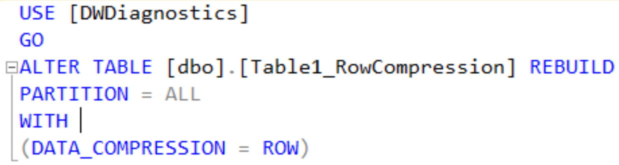
Source - https://msdn.microsoft.com/en-us/library/cc280464.aspx

Implementation of Data Compression

Row Compression | Page Compression

T-SQL Command

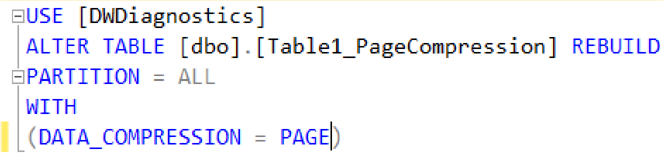
Row Compression



```
USE [DWDiagnostics]
GO
ALTER TABLE [dbo].[Table1_RowCompression] REBUILD
PARTITION = ALL
WITH
(DATA_COMPRESSION = ROW)
```

ALTER TABLE <TABLE_NAME> WITH (DATA_COMPRESSION=ROW)

Page Compression

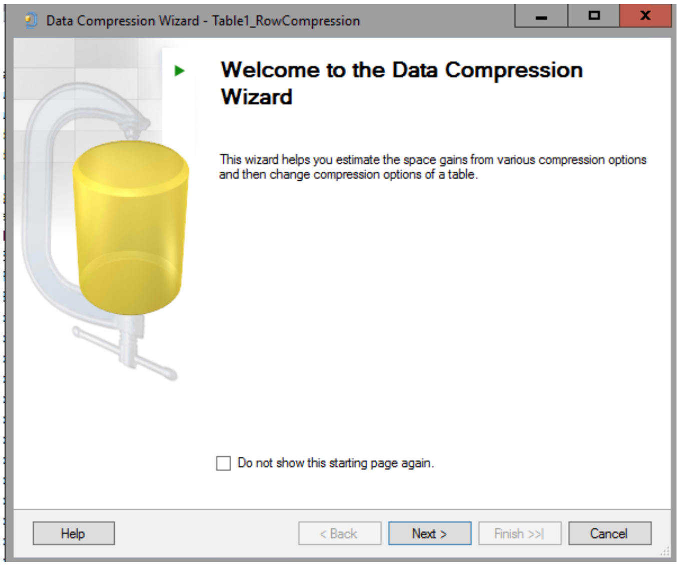


```
USE [DWDiagnostics]
ALTER TABLE [dbo].[Table1_PageCompression] REBUILD
PARTITION = ALL
WITH
(DATA_COMPRESSION = PAGE)
```

ALTER TABLE <TABLE_NAME> WITH (DATA_COMPRESSION=PAGE)

SSMS

Step 1 - Open Data Compression wizard. Right Click on Table > Storage > Manage Compression

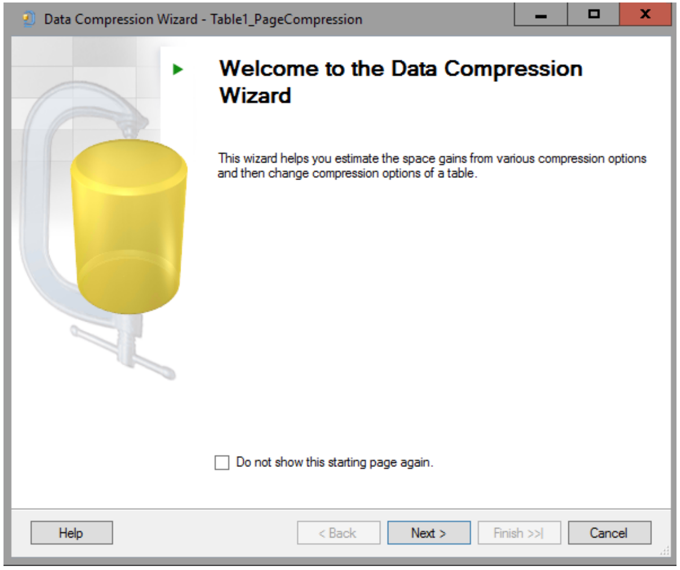


Data Compression Wizard - Table1_RowCompression

Welcome to the Data Compression Wizard

This wizard helps you estimate the space gains from various compression options and then change compression options of a table.

☐ Do not show this starting page again.



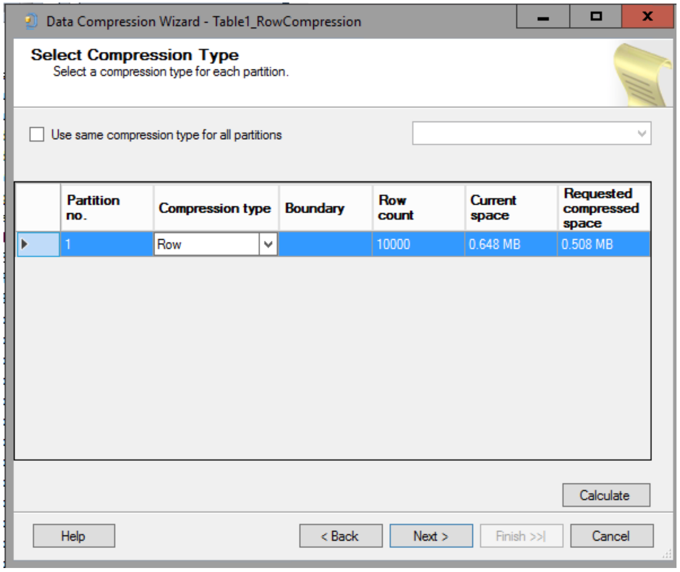
Data Compression Wizard - Table1_PageCompression

Welcome to the Data Compression Wizard

This wizard helps you estimate the space gains from various compression options and then change compression options of a table.

☐ Do not show this starting page again.

Step 2 - Select Compression Type & you can click on calculate to check the impact



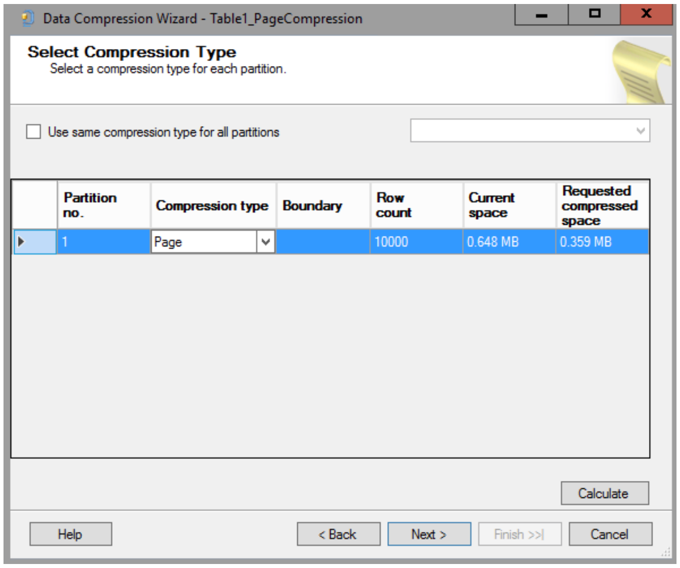
Data Compression Wizard - Table1_RowCompression

Select Compression Type

Select a compression type for each partition.

☐ Use same compression type for all partitions

Partition no.	Compression type	Boundary	Row count	Current space	Requested compressed space
1	Row		10000	0.648 MB	0.508 MB



Data Compression Wizard - Table1_PageCompression

Select Compression Type

Select a compression type for each partition.

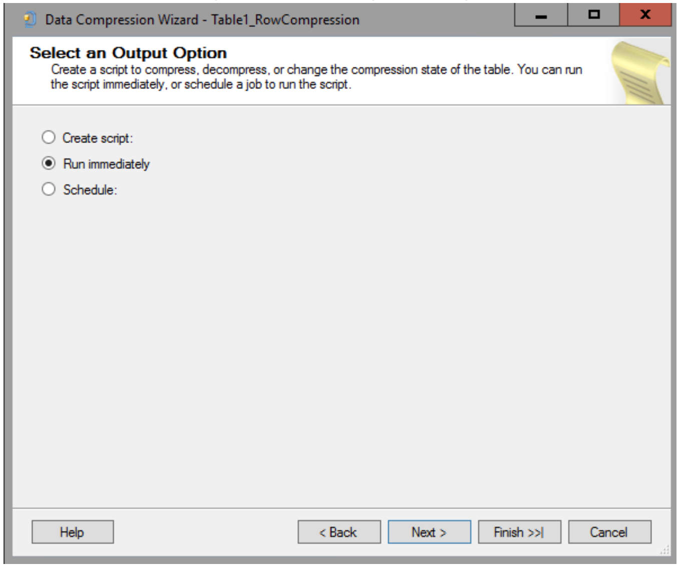
☐ Use same compression type for all partitions

Partition no.	Compression type	Boundary	Row count	Current space	Requested compressed space
1	Page		10000	0.648 MB	0.359 MB

You can use below system store procedure to estimate the compression results

EXEC sp_estimate_data_compression_savings 'Schema','Table_Name',Null,Null,'Type_of_Compression'

Step 3 - Select Option if you want to run it immediately or later by generating script



Data Compression Wizard - Table1_RowCompression

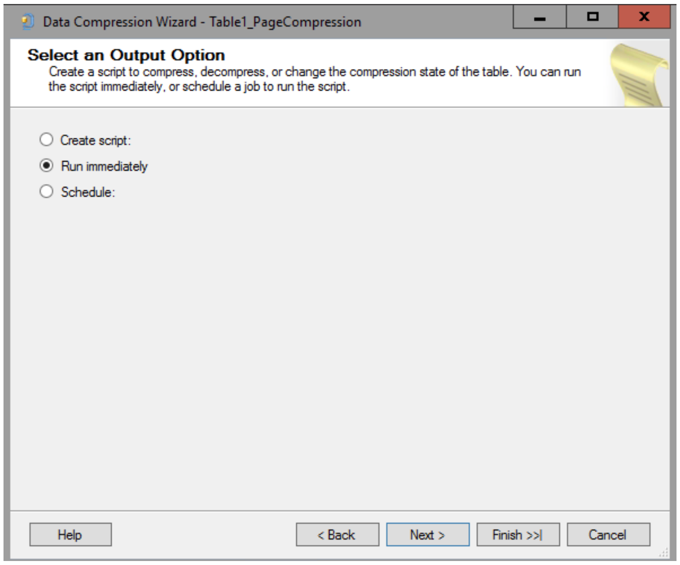
Select an Output Option

Create a script to compress, decompress, or change the compression state of the table. You can run the script immediately, or schedule a job to run the script.

○ Create script:

● Run immediately

○ Schedule:



Data Compression Wizard - Table1_PageCompression

Select an Output Option

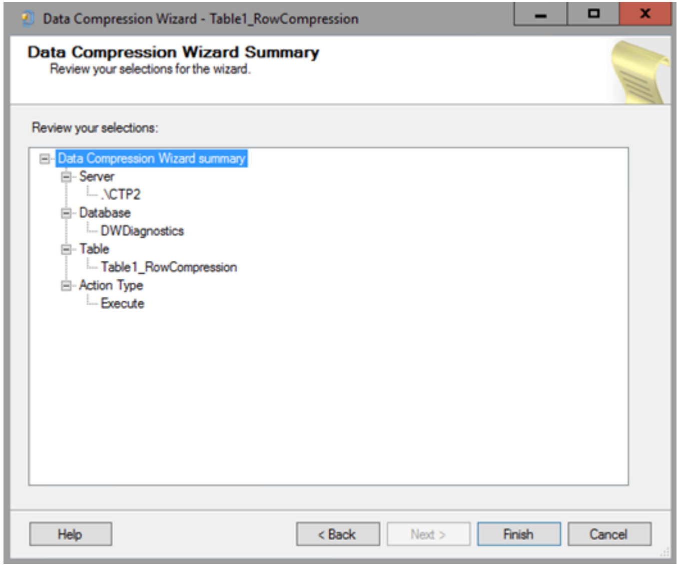
Create a script to compress, decompress, or change the compression state of the table. You can run the script immediately, or schedule a job to run the script.

○ Create script:

● Run immediately

○ Schedule:

Step 4 - Final Configuration window before you give final go ahead



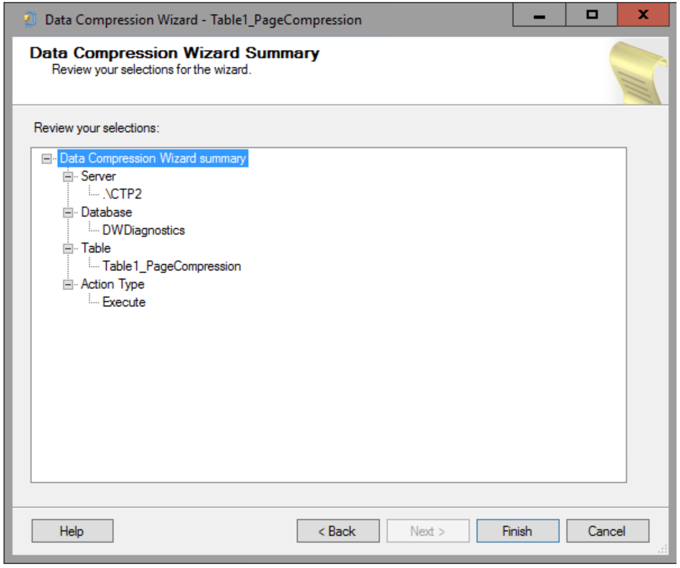
Data Compression Wizard - Table1_RowCompression

Data Compression Wizard Summary

Review your selections for the wizard.

Review your selections:

- Data Compression Wizard summary
  - Server
    - .\CTP2
  - Database
    - DWDiagnostics
  - Table
    - Table1_RowCompression
  - Action Type
    - Execute



Data Compression Wizard - Table1_PageCompression

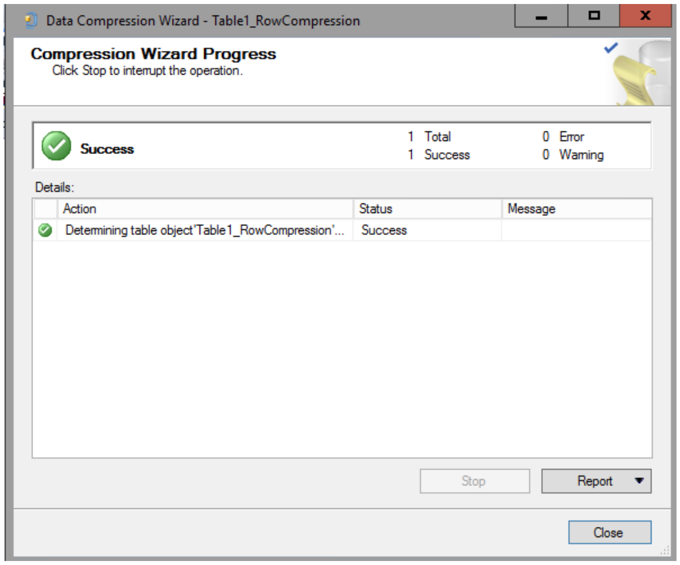
Data Compression Wizard Summary

Review your selections for the wizard.

Review your selections:

- Data Compression Wizard summary
  - Server
    - .\CTP2
  - Database
    - DWDiagnostics
  - Table
    - Table1_PageCompression
  - Action Type
    - Execute

Step 5 - Compression implemented successfully



Data Compression Wizard - Table1_RowCompression

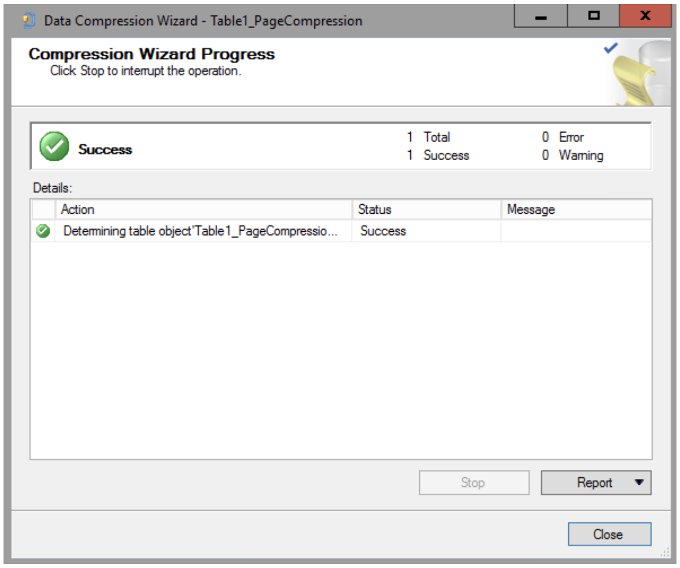
Compression Wizard Progress

Click Stop to interrupt the operation.

Success — 1 Total, 1 Success, 0 Error, 0 Warning

Details:

Action	Status	Message
✓ Determining table object 'Table1_RowCompression'...	Success	



Data Compression Wizard - Table1_PageCompression

Compression Wizard Progress

Click Stop to interrupt the operation.

Success — 1 Total, 1 Success, 0 Error, 0 Warning

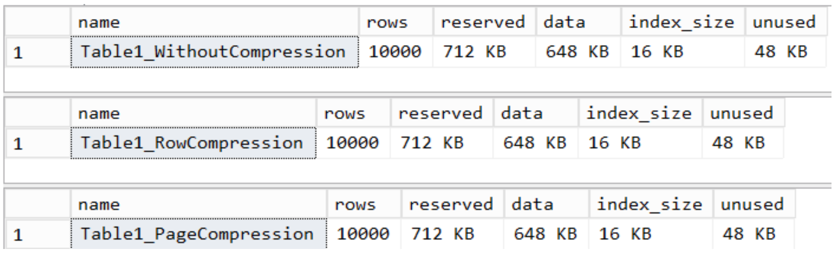
Details:

Action	Status	Message
✓ Determining table object 'Table1_PageCompressio...	Success	

Impact of Compression

We have created 3 identical tables with same data & compare their space without compression, with Row & Page compression. We can see that data size of table having Page compression is using least space.

Before Compression:-

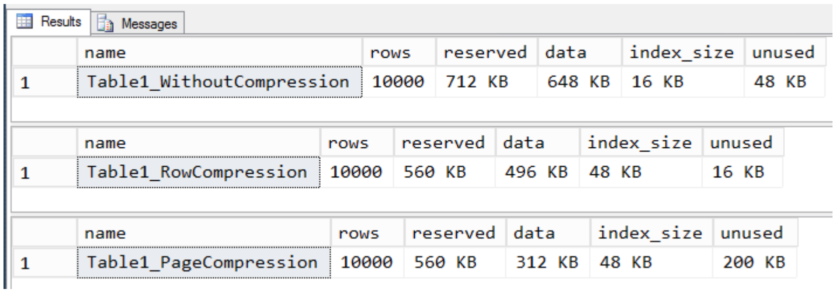


	name	rows	reserved	data	index_size	unused
1	Table1_WithoutCompression	10000	712 KB	648 KB	16 KB	48 KB

	name	rows	reserved	data	index_size	unused
1	Table1_RowCompression	10000	712 KB	648 KB	16 KB	48 KB

	name	rows	reserved	data	index_size	unused
1	Table1_PageCompression	10000	712 KB	648 KB	16 KB	48 KB

After Compression:-

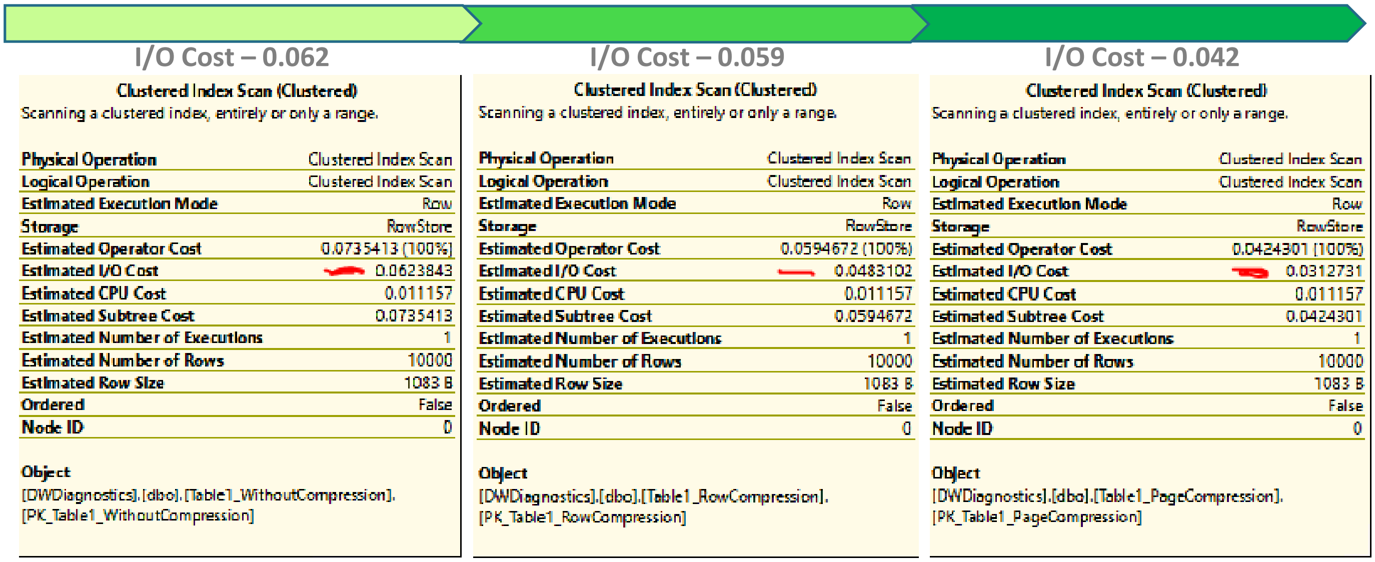


	name	rows	reserved	data	index_size	unused
1	Table1_WithoutCompression	10000	712 KB	648 KB	16 KB	48 KB

	name	rows	reserved	data	index_size	unused
1	Table1_RowCompression	10000	560 KB	496 KB	48 KB	16 KB

	name	rows	reserved	data	index_size	unused
1	Table1_PageCompression	10000	560 KB	312 KB	48 KB	200 KB

We also compare execution of all above 3 tables & as expected, we found I/O cost is lowest for Page Compression.



I/O Cost – 0.062

Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated Execution Mode	Row
Storage	RowStore
Estimated Operator Cost	0.0735413 (100%)
Estimated I/O Cost	0.0623843
Estimated CPU Cost	0.011157
Estimated Subtree Cost	0.0735413
Estimated Number of Executions	1
Estimated Number of Rows	10000
Estimated Row Size	1083 B
Ordered	False
Node ID	0

Object
[DWDiagnostics].[dbo].[Table1_WithoutCompression].
[PK_Table1_WithoutCompression]

I/O Cost – 0.059

Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated Execution Mode	Row
Storage	RowStore
Estimated Operator Cost	0.0594672 (100%)
Estimated I/O Cost	0.0483102
Estimated CPU Cost	0.011157
Estimated Subtree Cost	0.0594672
Estimated Number of Executions	1
Estimated Number of Rows	10000
Estimated Row Size	1083 B
Ordered	False
Node ID	0

Object
[DWDiagnostics].[dbo].[Table1_RowCompression].
[PK_Table1_RowCompression]

I/O Cost – 0.042

Clustered Index Scan (Clustered)

Scanning a clustered index, entirely or only a range.

Physical Operation	Clustered Index Scan
Logical Operation	Clustered Index Scan
Estimated Execution Mode	Row
Storage	RowStore
Estimated Operator Cost	0.0424301 (100%)
Estimated I/O Cost	0.0312731
Estimated CPU Cost	0.011157
Estimated Subtree Cost	0.0424301
Estimated Number of Executions	1
Estimated Number of Rows	10000
Estimated Row Size	1083 B
Ordered	False
Node ID	0

Object
[DWDiagnostics].[dbo].[Table1_PageCompression].
[PK_Table1_PageCompression]

Disable Data Compression

```
ALTER TABLE <TABLE_NAME> WITH (DATA_COMPRESSION=NONE)
```

Considerations while using row and page compression

- Data Compression is SQL Server feature is does not required any application level changes
- In Data Compression, You can buying less I/O operations & low disk space requirement in exchange of CPU overhead, You should plan & decide wisely before selecting this option
- Compression is available in limited editions of SQL Server
- System tables cannot be compressed
- Compression reduce the space requirement of row that can help to store more rows in page but MAX size of row will not change
- MAX Size of Row after compression (8060) > Current size of Row + Compression Overhead
  To enable compression above condition should be true. In case of Current size of row & compression overhead is more than MAX row size, compression cannot be enabled on table. When data type like VARCHAR is the row-size check is performed when the format is enabled
- Page compression needs around double space while enabling this option although you can shrink database after compression
- When table has multiple partitions, we need to specify the partition on which compression needs to enabled
- Non-clustered indexes do not inherit the compression property of the table
- When a clustered index is created on a table or you can specify if you want different compression for clustered index
- When a heap is configured for page-level compression, pages receive page-level compression only in the following ways:
  - Data is bulk imported with bulk optimizations enabled.
  - Data is inserted using INSERT INTO ... WITH (TABLOCK) syntax and the table does not have a non-clustered index.
  - A table is rebuilt byes executing the ALTER TABLE ... REBUILD statement with the PAGE compression option.
- New pages allocated in a heap as part of DML operations will not use PAGE compression until the heap is rebuilt. Rebuild the heap by removing and reapplying compression, or by creating and removing a clustered index.
- Any Change in compression setting of table will need all non-clustered indexes on the table(with no clustered index) to be rebuild so that new row allocation can be mentioned
- The disk space requirements for enabling or disabling row or page compression are the same as for creating or rebuilding an index
- To determine the compression state of partitions in a partitioned table, query the data_compression column of the sys.partitions catalog view
- When you are compressing indexes, leaf-level pages can be compressed with both row and page compression. Non–leaf-level pages do not receive page compression.
- Tables which implemented the vardecimal storage format in SQL Server 2005 will retain that setting when upgraded. You can apply Row & page compression but you will not get results as per requirements. We suggest to not to use vardecimal storage format as this feature is decrypted in SQL Server 2016 (https://msdn.microsoft.com/en-us/library/ms143729.aspx).
- When table is created with page compression, DB engine does not do any compression from start. Meta data is having compression information for future use. As data start coming to table, compression start working. Till the time page isn't full it was only Row Compressed, once page is full page & new row is ready to add DB engine applies Page compression to page. If Page can be compressed to add new row then page will be compressed otherwise it will be left as it is & new Page is allocated for new row.

Reference - https://msdn.microsoft.com/en-us/library/cc280449.aspx