**Data Compression**

SQL Server 2017 and Azure SQL Database support **row and page** compression for rowstore tables and indexes, and supports **columnstore and columnstore archival** compression for columnstore tables and indexes.

For rowstore tables and indexes, use the data compression feature to help reduce the size of the database. In addition to saving space, data compression can help improve performance of I/O intensive workloads because the data is stored in fewer pages and queries need to read fewer pages from disk. However, extra CPU resources are required on the database server to compress and decompress the data, while data is exchanged with the application. You can configure row and page compression on the following database objects:

* A whole table that is stored as a heap.
* A whole table that is stored as a clustered index.
* A whole nonclustered index.
* A whole indexed view.
* For partitioned tables and indexes, you can configure the compression option for each partition, and the various partitions of an object do not have to have the same compression setting.

**Enable Compression**

ALTER TABLE Production.TransactionHistory REBUILD PARTITION = ALL

WITH (DATA\_COMPRESSION = ROW);

ALTER INDEX IX\_TransactionHistory\_ProductID ON Production.TransactionHistory REBUILD PARTITION = ALL WITH (DATA\_COMPRESSION = PAGE);

**DISABLE Compression**

ALTER TABLE Person.Person REBUILD PARTITION = ALL

WITH (DATA\_COMPRESSION = NONE);

ALTER INDEX AK\_Person\_rowguid ON Person.Person REBUILD PARTITION = ALL

WITH (DATA\_COMPRESSION = NONE);

For columnstore tables and indexes, all columnstore tables and indexes always use columnstore compression and this is not user configurable. Use columnstore archival compression to further reduce the data size for situations when you can afford extra time and CPU resources to store and retrieve the data. You can configure columnstore archival compression on the following database objects:

* A whole columnstore table or a whole clustered columnstore index. Since a columnstore table is stored as a clustered columnstore index, both approaches have the same results.
* A whole nonclustered columnstore index.
* For partitioned columnstore tables and columnstore indexes, you can configure the archival compression option for each partition, and the various partitions do not have to have the same archival compression setting.

To perform archival compression, SQL Server runs the Microsoft XPRESS compression algorithm on the data. Add or remove archival compression by using the following data compression types:

* Use **COLUMNSTORE\_ARCHIVE** data compression to compress columnstore data with archival compression.
* Use **COLUMNSTORE** data compression to decompress archival compression. The resulting data continue to be compressed with columnstore compression.

ALTER TABLE ColumnstoreTable1 REBUILD PARTITION = 1

WITH (DATA\_COMPRESSION = COLUMNSTORE\_ARCHIVE) ;

ALTER TABLE ColumnstoreTable1 REBUILD PARTITION = ALL

WITH (DATA\_COMPRESSION = COLUMNSTORE\_ARCHIVE) ;

ALTER TABLE ColumnstoreTable1 REBUILD PARTITION = ALL

WITH (DATA\_COMPRESSION = COLUMNSTORE\_ARCHIVE ON PARTITIONS (2,4)) ;

ALTER TABLE ColumnstoreTable1 REBUILD PARTITION = 1

WITH (DATA\_COMPRESSION = COLUMNSTORE) ;

ALTER TABLE ColumnstoreTable1 REBUILD PARTITION = ALL

WITH (DATA\_COMPRESSION = COLUMNSTORE) ;

ALTER TABLE ColumnstoreTable1 REBUILD PARTITION = ALL

WITH (DATA\_COMPRESSION = COLUMNSTORE ON PARTITIONS (2,4) ) ;

ALTER TABLE ColumnstoreTable1 REBUILD PARTITION = ALL WITH (

DATA\_COMPRESSION = COLUMNSTORE ON PARTITIONS (4,5),

DATA COMPRESSION = COLUMNSTORE\_ARCHIVE ON PARTITIONS (1,2,3) ) ;

**Performance**

Compressing columnstore indexes with archival compression, causes the index to perform slower than columnstore indexes that do not have the archival compression. Use archival compression only when you can afford to use extra time and CPU resources to compress and retrieve the data.

The benefit of archival compression, is reduced storage, which is useful for data that is not accessed frequently.