

علاوه بر فشرده سازی خودکار بک آپ‌ها که پیشتر در مورد آن‌ها [صحبت شد](#) ، اس کیوال سرور 2008 دو نوع فشرده سازی دیگر را نیز پشتیبانی می‌کند:

#### : Row Compression

حالت row compression نحوه‌ی ذخیره سازی فیزیکی داده‌ها را تغییر می‌دهد. فعال سازی آن اثرات زیر را خواهد داشت:  
(الف) متادیتای هر رکورد را حداقل می‌کند (منظور از متادیتا اطلاعاتی مانند اطلاعات ستون‌ها، طول و آفست و غیره است)  
(ب) داده‌های عددی و رشته‌هایی با طول ثابت، به صورت اطلاعاتی با طول متغیر ذخیره خواهند شد، درست مانند varchar ها.

برای ایجاد جدولی که row compression در آن به صورت پیش‌فرض فعال است، می‌توان مانند مثال زیر عمل کرد:

```
CREATE TABLE MyTable
(
    ID int identity Primary key,
    Name char(100),
    Email char(100)
)
WITH (DATA_COMPRESSION = Row);
GO
```

و اگر جدول موجودی را می‌خواهید تغییر داده و این خاصیت را بر روی آن فعال نمایید، روش زیر را اعمال کنید:

```
Alter TABLE MyTable REBUILD WITH (DATA_COMPRESSION=Row, MAXDOP=2);
```

در اینجا MAXDOP مشخص می‌کند که از چند CPU باید برای فشرده سازی استفاده شود. (اگر جدولی حجیم دارید، به این صورت می‌توان عملیات فشرده سازی را سریعتر به پایان رساند)

#### : Page Compression

در روش دوم فشرده سازی اطلاعات در اس‌کیوال سرور 2008 ، که مهم‌ترین حالت موجود نیز می‌باشد، اطلاعات مشترک، بین سطرها یک صفحه به اشتراک گذاشته می‌شوند. این روش از فناوری‌های زیر استفاده می‌کند:

(الف) روش row compression که در مورد آن صحبت شد جزئی از این روش است.

(ب) Prefix Compression : به ازای هر ستون در یک صفحه، Prefix های تکراری یافت شده و در هدر مخصوص فشرده سازی ذخیره می‌شوند (محل این هدر پس از هدر صفحه است). سپس هرجایی که به این Prefix ها اشاره شده‌باشد، عدد منحصر بفرد شناسایی کننده آن‌ها نسبت داده می‌شود.

(ج) Dictionary Compression : در این حالت مقادیر تکراری یک صفحه جستجو شده و در هدر فشرده سازی صفحه ذخیره می‌شوند. حالت Prefix Compression فقط به یک ستون منحصر می‌شود اما Dictionary Compression به کل صفحه اعمال می‌گردد.

برای فعال سازی آن در یک جدول جدید به روش زیر می‌توان عمل نمود:

```
CREATE TABLE MyTable
(
    ID int identity Primary key,
```

```
Name char(100),
Email char(100)
)
WITH (DATA_COMPRESSION = Page);
```

و برای اعمال آن به جدولی موجود از روش زیر می‌توان استفاده کرد:

```
Alter TABLE MyTable REBUILD WITH (DATA_COMPRESSION=Page, MAXDOP=2);
```

یک سری رویه‌های ذخیره شده سیستمی جدید نیز برای محاسبه حجم جداول، پیش و پس از فشرده سازی (بدون فشرده سازی واقعی) نیز در این نگارش گنجانده شده‌اند که به شرح زیر هستند:

```
-- بررسی اینکه چه میزان فضا با اعمال فشرده سازی صفحات قابل صرفه جویی خواهد بود
EXEC sp_estimate_data_compression_savings 'schemaname', 'TableName', NULL, NULL, 'PAGE';

-- بررسی اینکه چه میزان فضا با اعمال فشرده سازی ردیف‌ها قابل صرفه جویی خواهد بود
EXEC sp_estimate_data_compression_savings 'schemaname', 'TableName', NULL, NULL, 'ROW';
```

بنابراین قبل از اینکه فشرده سازی را فعال نمائید، ابتدا بررسی کنید آیا واقعا میزان قابل توجهی اطلاعات فشرده خواهند شد و نتیجه حاصل رضایت بخش است یا خیر. همچنین باید در نظر داشت که جداول و یا ایندکس‌هایی که read و write بالایی دارند برای این منظور مناسب نیستند. برای یافتن آن‌ها کوئری زیر را اجرا کنید:

```
USE dbName;
SELECT objectname = OBJECT_NAME(s.object_id),
       indexname = i.name,
       i.index_id,
       reads = range_scan_count + singleton_lookup_count,
       'leaf_writes' = leaf_insert_count + leaf_update_count + leaf_delete_count,
       'leaf_page_splits' = leaf_allocation_count,
       'nonleaf_writes' = nonleaf_insert_count + nonleaf_update_count +
       nonleaf_delete_count,
       'nonleaf_page_splits' = nonleaf_allocation_count
FROM   sys.dm_db_index_operational_stats (DB_ID(), NULL, NULL, NULL) AS s
INNER JOIN sys.indexes AS i
      ON i.object_id = s.object_id
WHERE  OBJECTPROPERTY(s.object_id, 'IsUserTable') = 1
      AND i.index_id = s.index_id
ORDER BY
       leaf_writes DESC,
       nonleaf_writes DESC
```

و جهت تکمیل مبحث می‌توان به مقاله بسیار جامع زیر که اخیرا توسط مایکروسافت منتشر شده است رجوع نمود:

[Data Compression: Strategy, Capacity Planning and Best Practices](#)