VERİ TABANI VE YÖNETİMİ DERSİ – UYGULAMA FÖYLERİ

Föy-4

Konu: İndeks Teorisi, İndeks Türleri, İndeksleri Tanımlamak ve Yönetmek

GENEL BİLGİLER (Ayrıntılı Bilgi için Kitabın 12. Ünitesi incelenebilir.)

Verileri okurken dağınık saklama ortamından nasıl daha hızlı okuma yapılacağı ile ilgili olan indeks kavramı, tablolardan veri çekmek için gerekli sorgular çalıştırılırken gereken süreyi kısaltmak için kullanılan bir kavramdır.

Yığın (Heap): Herhangi bir kurala göre sıralı olmayan veri topluluğu: Bu türdeki veriler üzerinde arama: Tablo taraması (table scan)

Clustered indeks: Bir tablo üzerinde sadece bir adet tanımlanabilen ve verilerin bir niteliğine göre fiziksel olarak sıralanması durumu

Nonclustered indeks: Bir clustered indeks ya da heap üzerinden verilere erişimi hızlandırmak için tanımlanan indekstir. Birden fazla tanımlanabilir.

Tablo 4 durumda olabilir → Heap, Sadece clustered indeksli, heap üzerinden çalışan Nonclustered indeksli, Clustered indeks üzerinden çalışan Nonclustered indeksli

<u>İndeks kullanım yerleri</u>

1- Noktasal sorguları hızlandırmak için: where sözcüğünden sonraki sorgu için

SELECT *

FROM production.product

WHERE productID=322

2- Aralık Sorgularını hızlandırmak için

3-Join işlemlerinde anahtar eşlemelerini hızlandırmak için

4-Veri tekrarından kaçınmak için (Primary key constraint ve unique constraint için → unique indeks)

5- Order by ile verileri sıralı biçimde daha hızlı döndürmek için

İndeks mimarisi hakkında detaylı bilgi için kitabın 289-294. Sayfaları arasındaki bölüme göz atabilirsiniz.

SQL Server İndeks Türleri

Clustered ve nonclustered indekslerin her birinin üzerinde de geçerli olabilecek bazı durumlar mevcuttur. Bu durumlarda tanımlanabilecek indeks çeşitlerinden bazıları şu şekildedir:

Unique İndeks: Indeks'teki verilerin tekrarlanamaması için kullanılabilir. Bunun için, bir indeksi UNIQUE deyimi ile tanımlamak yeterlidir. Hem veri çekme işlemlerinin hızlandırmada hem de verinin tekrarlanmamasını denetlemede aynı indeks kullanılabilir. UNIQUE indeks hem clustered hem de nonclustered olabilir. Primary Key tanımlama işlemini gerçekleştirir. Bu türden bir Constraint tanımı yapıldığı zaman, SQL Server bir Unique Index tanımlama işlemini gerçekleştirir.

Karma (Composite) İndeks: İlişkisel Veritabanındaki, bir indeks ister tekil ister çoğul indeks olun bir tek alandan ibaret olması sık rastlanan bir durumdur ancak bu bir kural değildir. Gerektiğinde 16 sütun veya toplam uzunluklar 900Byte'i aşmamak üzere birden fazla alanı kapsayan bir indeks tanımlanabilir

Kapsam (Covering) İndeks: Bir sorgunun WHERE kısmı dahil olmak üzere, seçilen sütunları ile birlikte bir tek indeks olarak tanımlanmasına Covering İndeks denir. Bu türden bir non-Clustered indeks tanımlandığında, SQL Server sadece bu indeksi kullanır. Covering indeks genellikle çok I/O işlemi gerektirir ama sorunun çok hızlı neticelenmesini sağlar.

OLTP ve OLAP türü sistemlerde indeks tanımlamanın yerini araştırınız.

İndeks Tanımlama Yaklaşımları

Clustered İndeks Tanımlarken: Sık sorgulanan sütunlar, boyutu küçük sütunlar, daha az değişim gösteren sütunlar tercih edilmelidir.

Nonclustered İndeks Tanımlanırken: Performans kazanımı vs Tamirat bedeli dengesi, Arama argümanları, Yeterli Seçicilik, Yabancı Anahtarlar, Sorgu kapsama

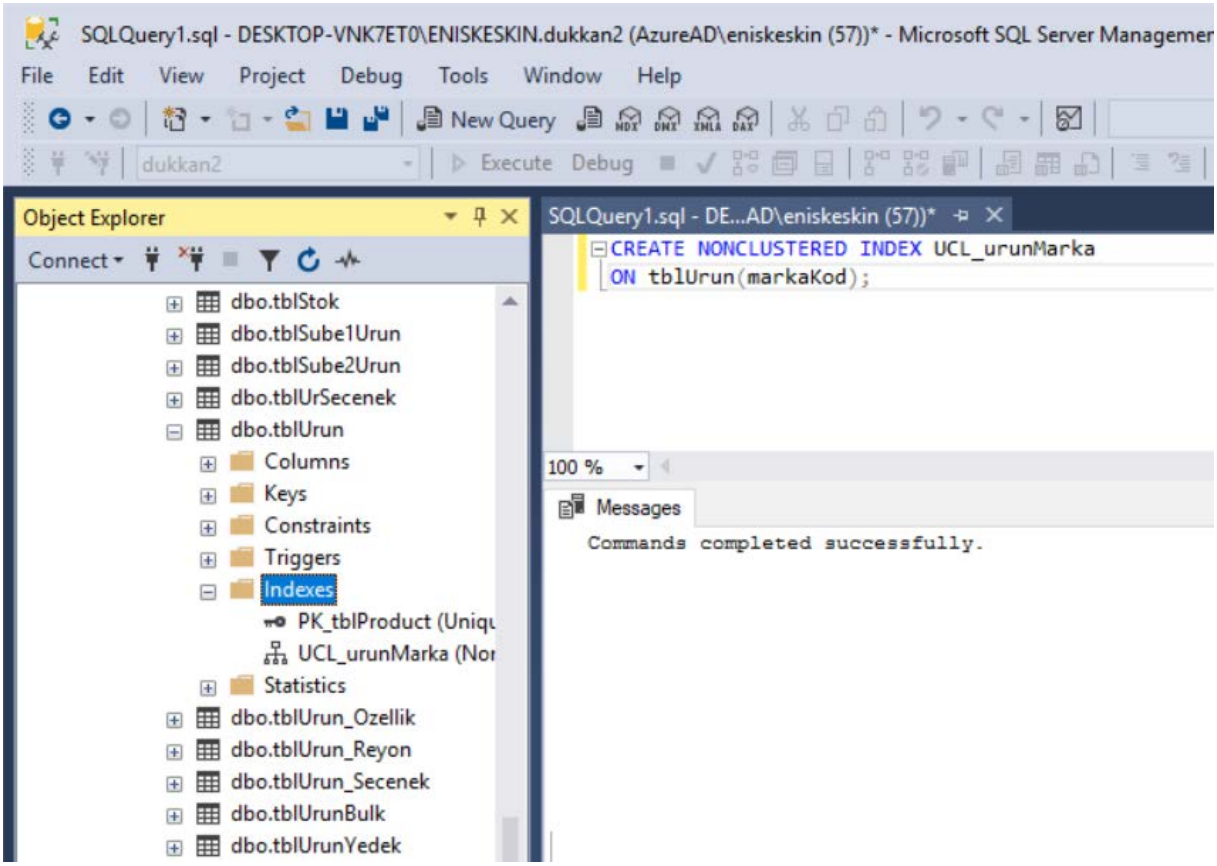
İndeks Oluşturmak

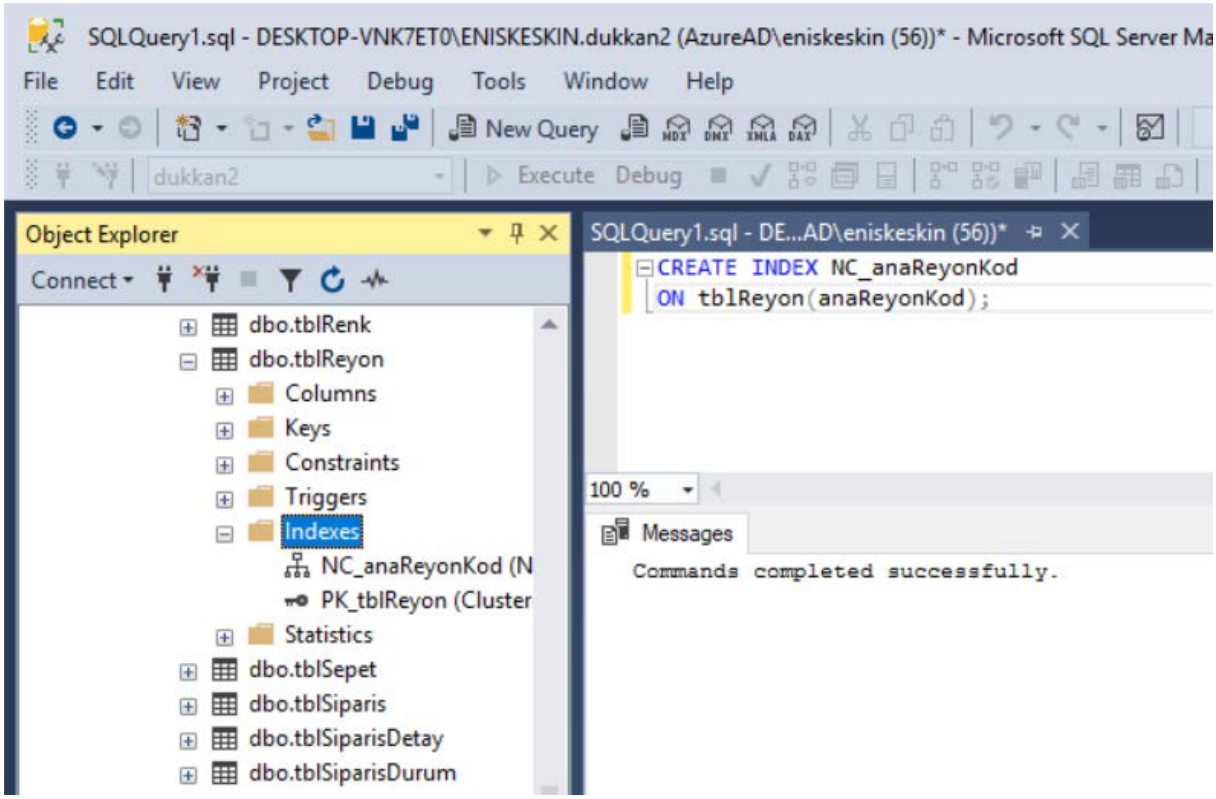
İndeks oluşturmak için en basit ifade şu şekildedir:

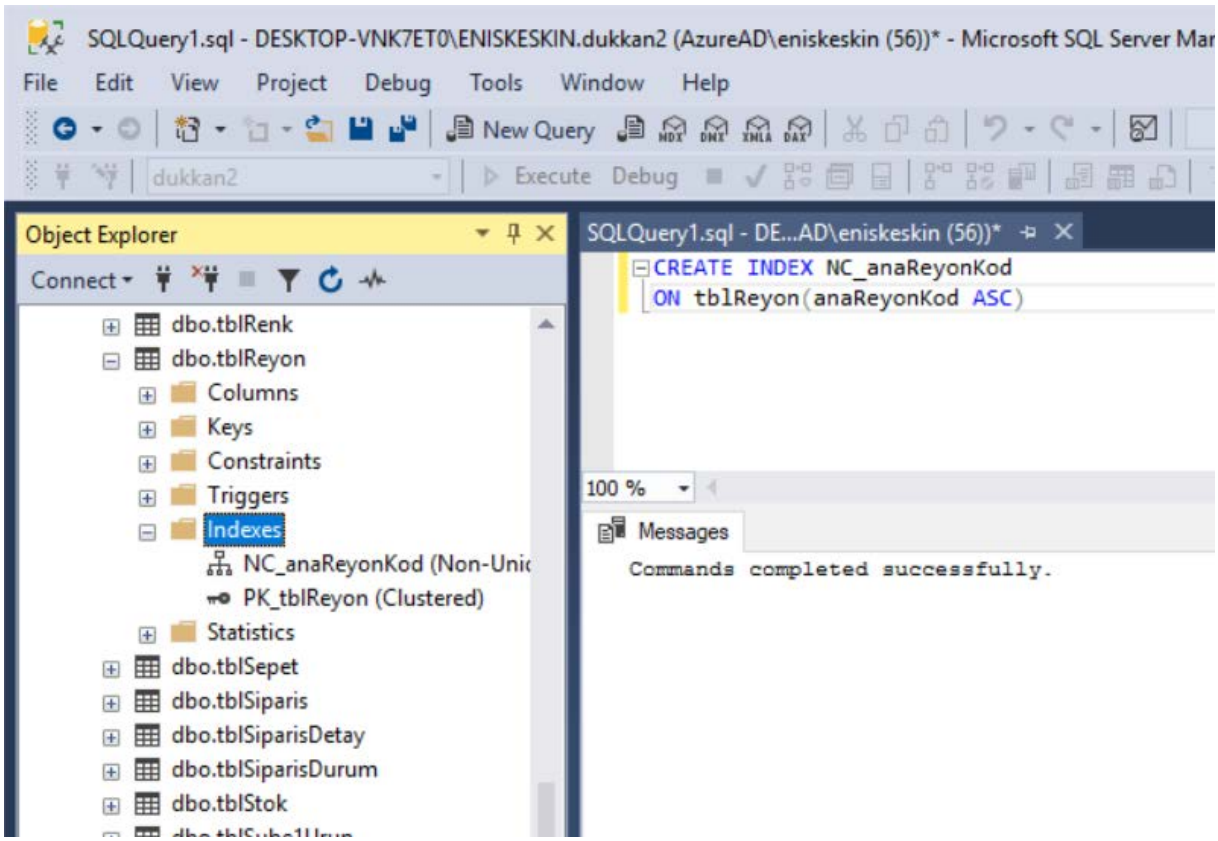
CREATE indeks-tipi INDEX indeks-ismi

ON tablo-ismi(sutun-ismi)

İfade	Açıklama
İndeks_tipi	UNIQUE CLUSTERED veya sadece CLUSTERED, NONCLUSTERED şeklinde indeksin tipini belirten bir kelime. Tip belirtilmezse, SQL Server NONCLUSTERED olduğunu varsayar
İndeks_ismi	İndekse verilen isim. Genellikle Clustered ise CL, Unique ise UCL, Nonclustered ise NC… şeklinde örneklerle bir isim vermek, anlaşılırlığı arttıracaktır.
Tablo_ismi	Index'in üstünde tanımlandığı tablo ya da view'in ismi
Sutun_ismi	Tablo veya view'de indekslenmesi istenen sütun veya sütunların ismi.(Composite Index'te birden fazla sütun bulunur.)




SQLQuery1.sql - DESKTOP-VNK7ET0\ENISKESKIN.dukkan2 (AzureAD\eniskeskin (56))* - Microsoft SQL Server Ma
File Edit View Project Debug Tools Window Help
New Query
dukkan2
Execute Debug
Object Explorer
Connect
dbo.tblRenk
dbo.tblReyon
Columns
Keys
Constraints
Triggers
Indexes
NC_anaReyonKod (N
PK_tblReyon (Cluster
Statistics
dbo.tblSepet
dbo.tblSiparis
dbo.tblSiparisDetay
dbo.tblSiparisDurum
SQLQuery1.sql - DE...AD\eniskeskin (56))*
CREATE INDEX NC_anaReyonKod
ON tblReyon(anaReyonKod);
100 %
Messages
Commands completed successfully.


SQLQuery1.sql - DESKTOP-VNK7ET0\ENISKESKIN.dukkan2 (AzureAD\eniskeskin (56))* - Microsoft SQL Server Mar
File Edit View Project Debug Tools Window Help
New Query
dukkan2
Execute Debug
Object Explorer
Connect
dbo.tblRenk
dbo.tblReyon
Columns
Keys
Constraints
Triggers
Indexes
NC_anaReyonKod (Non-Unic
PK_tblReyon (Clustered)
Statistics
dbo.tblSepet
dbo.tblSiparis
dbo.tblSiparisDetay
dbo.tblSiparisDurum
dbo.tblStok
SQLQuery1.sql - DE...AD\eniskeskin (56))*
CREATE INDEX NC_anaReyonKod
ON tblReyon(anaReyonKod ASC)
100 %
Messages
Commands completed successfully.

Yukarıdaki indeksleri tanımlayın ve sol menüden ilgili tablo altında indeksleri gözlemleyin. Daha önceden tanımlı iseler ya silip yeniden oluşturun ya da isimlerini farklı vererek yeniden oluşturun.

Unique İndeks

Indeksteki verilerin tekrarlayamaması maksadıyla kullanabiliriz. Bunun için unique deyimi ile tanımlamak yeterlidir. Hem veri çekme işlemlerini hızlandırmada hem de verilerin tekrarlamamasını denetlemede aynı indeks kullanılabilir. UNIQUE indeks clustered türden olabileceği gibi, nonclustered türden de olabilir. Primary Key Constraint veya Unique Constraint tanımı yapıldığı zaman, SQL Server bir Unique Index tanımlama işlemini gerçekleştirir. Bu türden bir Constraint tanımlanırken, SQL Server'e clustered olup olmayacağı hakkında bilgi verilebilir.

Primary key constraint, özel olarak bir seçenek belirtilmedi ise clustered unique indeks, unique constraint ise unique indeks olarak tanımlanır.

Örnek:

tblDoviz, tablosu ürün fiyat listeleri gösterilirken sürekli olarak kullanılmakta ve içerisinde sadece 2-3 satır yer almakta. Join işlemini hızlandırmak için bu tablo üstünde Primary Key Constraint'i Clustered Unique Indeks olarak tanımlayabiliriz.

```
ALTER TABLE tblDoviz
ADD CONSTRAINT PK_Doviz PRIMARY KEY (dovizKod)
CLUSTERED;
```

Örnek:

tblUrun_Reyon tablosunun iki sütunu birlikte birincil anahtar yapıldığında otomatik olarak karma indeks oluşturacaktır.

```
ALTER TABLE tblUrun_Reyon
ADD CONSTRAINT PK_tblUrun_Reyon PRIMARY KEY (urunKod, ReyonKod)
CLUSTERED;
```

Kapsam İndeks

Örnek:

E-Ticaret sitemizin herhangi bir yerinde şu türde bir sorgumuz olsun:

SELECT markaKod, urunAd, urunOzet

FROM tblUrun

WHERE barkod='FLOPPY_ALPS';

tblUrun tablosunda markaKod sütunu üstünde hali hazırda bir indeks zaten tanımlı. Şayet, barkod, urunAd, ve urunOzet sütunlarını içeren bir Covering Indeks tanımlarsak bu sorgu daha hızlı gelecektir.

Bunun için şu şekilde bir Covering Index oluşturabiliriz.

CREATE INDEX CV_tblUrun

ON tblUrun (barkod,urunAd,urunOzet)

Indeks'teki sütunların sırasına dikkat edin. Barkod, urunAd, urunOzet.. WHERE şartında birden fazla sütun AND ile filtreleniyorsa aynı sırada indekslenmesi gerekir

İndeks Seçeneklerini Ayarlamak

Indekslerle ilgili birçok ayarlama yapılabilir. Bu seçeneklerin genel kullanımı şu şekildedir:

CREATE indeks-tipi INDEX indeks-ismi

ON tablo-ismi (aranan-sutun-isimleri)

WITH(seçenek=değer, …n)

Seçeneklerin genel bir listesini tabloda bulabilirsiniz.

Seçenek	Alabilceği Değerler	İşlev
PAD_INDEX	{ON|OFF}	Ara seviye indeks sayfalarındaki boşluk oranının dal seviyelere aynı olup olmayacağını tayin eder. Aynı değil ise 1 indekslik boşluk bırakır.
FILLFACTOR	Fillfactor yüzde oranı	Uç seviye indeks sayfalarında başlangıçta ne kadar boşluk bırakılacağını tayin eder.
IGNORE_DUP_KEY	{ON|OFF}	Unique indeks ile tanımlı bir sütuna çoklu satır eklemesi yapılırken, tekrarlayan satırlar gelirse nasıl davranılacağını belirler.
DROP_EXISTING	{ON|OFF}	Olan bir indeksin silinip yeniden oluşturulamayacağını tayin eder.
MAXDOP	{0-64} Default: 0	Indeks değiştirme ve oluşturma işlemleri sırasında sistemin tamamının sömürülmesi önlemek üzere, kaç CPU kullanılacağını kısıtlamak için 0'dan farklı değer verebilirsiniz.

tblUrun tablosu üstündeki markaKod sütununu indekslerken online olarak DML ifadelerine izin verilmesini ve en fazla 3 CPU kullanılmasını istiyorsak:

```
CREATE CLUSTERED INDEX UCL_urunMarka
ON tblUrun(markaKod)
WITH(MAXDOP = 3)
```

FILLFACTOR ve PAD_INDEX Parametrelerini Ayarlamak

Bir indeksi, sayfalarında boş yerler bırakmak üzere tanımlamak için şu genel ifade kullanılabilir:

```
CREATE indeks_tipi INDEX indeks_ismi
ON tablo_ismi(sutun_ismi)
WITH (FILLFACTOR= yüzde_oran)
```

tblSİparisDetay tablosu üstünde %80 doluluk oranında bir indeks tanımlayalım ve bu indeksin ara seviyelerde de bu şekilde boşluk bırakmasını sağlayalım. Çünkü, sistemimizin sürekli olarak sipariş olarak almasını bekliyoruz.

Indeksler Üzerinde Değişiklik Yapmak

Indeksler üstünde çeşitli amaçlar için değişiklik yapmak gerekebilir. Bu türden değişiklikleri şu şekilde gruplayabiliriz.

Yeniden derlemek (Rebuild): Bir indeki silip yenisini oluşturmak suretiyle, kapladığı alanı azaltmak ve tablodaki kayıtları karşılamak üzere içyapısını yeniden inşa etmek amacıyla kullanılır.

Yeniden düzenleme (Reorganize): İndeks tanımında yer alan FILL FACTOR değerine eşit olarak sadece uç seviye indeks sayfalarını yeniden yapılandırır.

Indeksi Kullanma Kapatmak (Disable): Geçici olarak indeksin sorgu iyileştirmelerinde kullanılmasını sağlamak gerektiğinde tercih edilebilir. Kapatılan indeks bir clustered indeks ise tablo offline geçmiş olur. Yani veri okumaya ve yazmaya karşı engellenmiş olur.

Indeks seçeneklerini değiştirmek (Set): Yukarıdaki tabloda verdiğimiz seçeneklere yeni değer vermek için kullanılır. Ancak bazı seçenekler ALTER INDEX ifadesi ile değiştirilmez.

ALTER INDEX için genel ifade şu şekilde özetlenebilir.

```
ALTER INDEX {indeks-ismi | ALL}
ON {sema-ismi.tablo-ismi.indeks-ismi}
{
 REBUILD [WITH(insa_secenekleri)] |
 REORGANIZE[ WITH(LOB_COMPACTION=ON|OFF) ]
 DISABLE |
 SET (indeks_secenekleri)
}
```

ALTER INDEX ifadesi ile şu seçenekler ayarlanamaz:

- İndeks üstünde parçalama işlemleri yapılamaz
- İndekse sütun eklenemez, sütun çıkartılamaz

BU türden seçenekler için, CREATE INDEX ifadesini DROP EXISTING ile çalıştırmak gerekir.

Yeniden derleme

Bir indeksi yeniden derleyerek fillfactor, pad_indeks, ignore_dupkey gibi seçenekleri değiştirilebilir.

tblUrun tablosu üstündeki indeksler 80% fillfactor ile, TEMPDB'de yeniden sıralanmak üzere, istatistiklerini otomatik hesaplayacak şekilde yeniden derlensin:

```
USE Dukkan
GO
ALTER INDEX ALL ON dbo.tblUrun
REBUILD WITH (FILLFACTOR = 80, SORT_IN_TEMPDB = ON,
 STATISTICS_NORECOMPUTE = ON);
GO
```

Yeniden organize etme

REORGANIZE ile aralarda boşalan indeks sayfalarının içeriği tam olarak boşaldı ise bu tür sayfaların atılmasını sağlar. Bunun dışında, indekse başlangıçta ayrılan sayfaları kullanmaya devam edecek işlemler yaptığından fazla kaynak tüketmez. Bu nedenle de tempdb'de ek yer gereksinimi yoktur. Verileri kendi yerlerinde işler.

tblUrun.urunTanitim alanı VARCHAR(max) türden bir alan olduğundan, bu tablo üstündeki

```
ALTER INDEX CV_tblUrun
ON tblUrun
REORGANIZE WITH (LOB_COMPACTION = ON)
```

Kapatma

Bazen tablolar üstündeki indeksleri bir süre için kullanıma kapatmak gerekebilir.

Örnek:

tblUrun tablosu üstündeki UCL_urunMarka adlı clustered indeksi bir süre için kapatalım:

```
ALTER INDEX CV_tblUrun
ON tblUrun
DISABLE
```

Seçenekleri değişme

İndekslerin bazı seçenekleri SET ifadesi ile değiştirmek mümkün. Bu seçenekleri bir listesi şu şekilde verilebilir:

ALTER INDEX UCL_urunMarka

ON tblUrun

SET(ALLOW_PAGE_LOCKS=ON);

İndeksleri Silmek

DROP INDEX indeks-ismi

ON sema-ismi.tablo-ismi

[WITH seçenekler]

İndekslere Bakım Yapmak

İndeksler, verilerin eklenip silinmesi ile zaman içerisinde düzenlerini kaybedebilirler. Bu dağınıklığı ortadan kaldırmak için iki yöntem takip edilebilir.

→ Düzenleme (Reorganize) veya Yeniden Derleme (Rebuild)

→ Silerek oluşturma (Create indeks…drop existing)

İndeks Dağınıklıklarını Gözlemlemek

Bir indeksin dağınıklık durumunu bilmeniz, onu yeniden derleme veya düzenlemek gerekip gerekmediği hakkında fikir verir. Özellikle büyük veriler çalışırken, indeks derleme işlemleri çok fazla külfeti olan işlemler olarak ele alınmalıdır.

İndekslerin dağınıklık durumlarını gözlemlemek için SQL Server *sys.dm_db_index_physical_stats function* fonksiyonunu sağlar. Fonksiyonun genel kullanımı şu şekildedir.

Sys.dm_db_index_physical_stats(

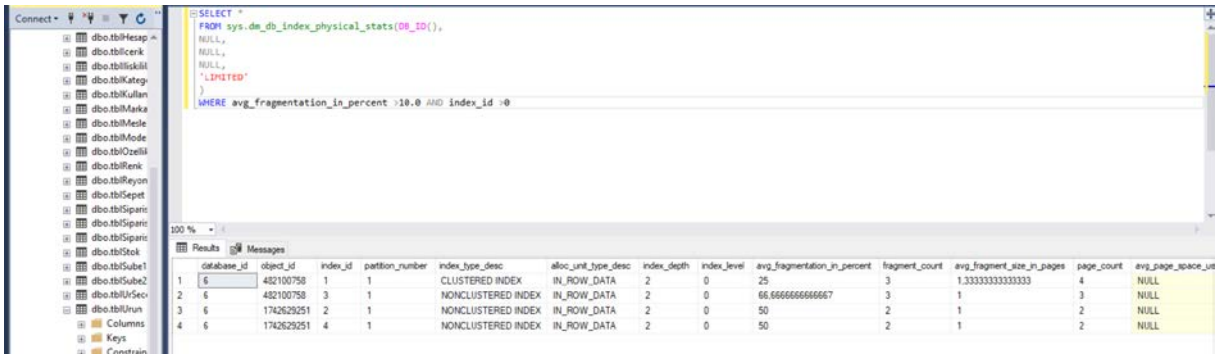
{ database_id | NULL},

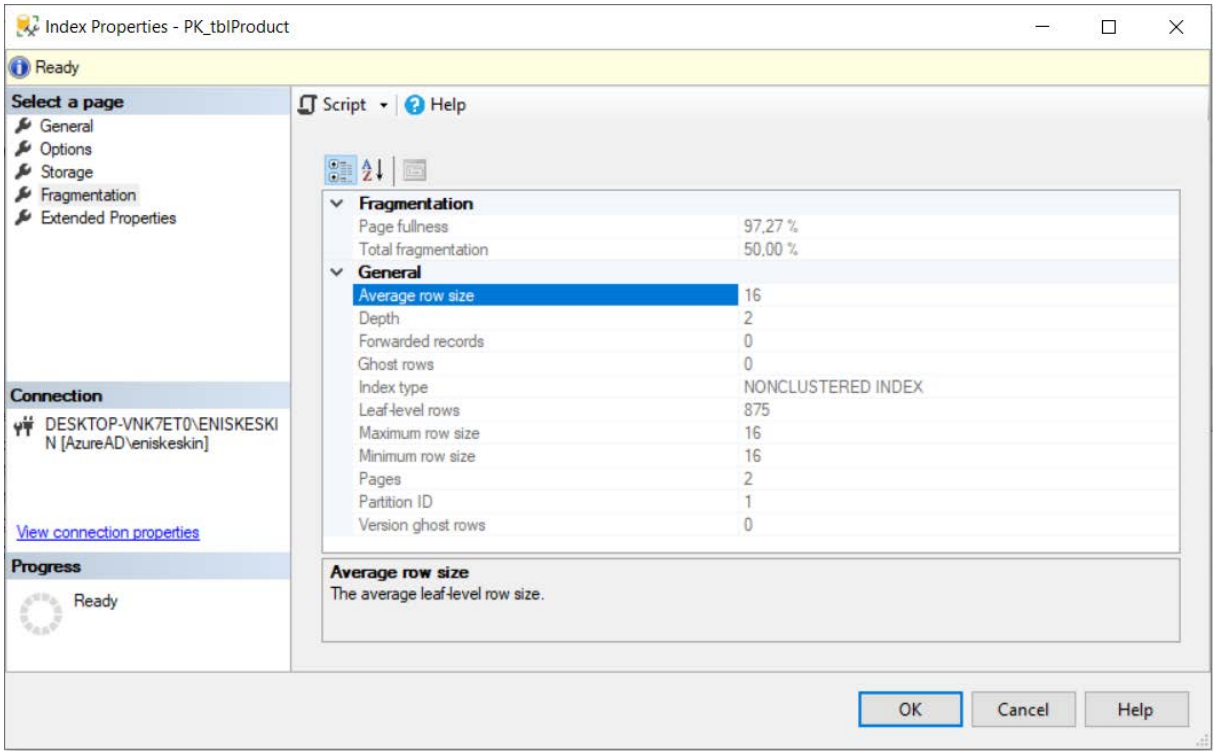
{object_id | NULL},

{index_id |NULL |0},

{partition_number | NULL},

{mode | NULL |DEFAULT })





BÖLÜM SONU ETKİNLİĞİ

1. Select urunkod, urunad, urunozet
   From tblurun
   Where urunad LIKE '%floppy%'

   Bu sorgunun hızlanması için nasıl bir indeks tanımlanmalıdır? Tanımlayınız.

2. İndeksler zaman içinde deforme olurlar. Bu durumda neler yapılabilir? Gözden geçiriniz.