

# SQL Server'da İndeks ve İstatistik Kavramları Sıkça Sorulan Sorular

## SQL Server'da İndeks kavramı nedir?

İndeks, verilerin indeksleme alanına göre sıralanmış ve tablo halini almış şeklidir. Belirli bir sıra düzenine geçmek için indeks yapısı kullanılır. Bu sayede veriler istenilen sıraya göre dizilmiş olur ve verilerin olduğu sıradan daha farklı bir şekilde görünümü sağlanır.

## İndeks oluşturmaın faydaları ve performansa etkileri nedir?

Veritabanında saklanan verilerin sayısının artması performansta olumsuz sonuçlara neden olabilir. Dağınık bir yapıda olan verilerde istenilen veriyi aramak için tablo taraması (Table Scan) işlemi yapılır. Bu işlemin küçük boyutlu bir tabloda yapılması kolaydır ancak artan veri miktarına göre bu işlem zaman kaybına neden olabilir. Verilere erişimde hızı arttırmak için Index oluşturma yöntemi kullanılır.

## İndeks bozulmalarının tespiti için nasıl bir yöntem izlenebilir?

SQL Server'da kullanılan clustered veya non-clustered indeksler, bulundukları tabloda insert, update, delete gibi işlemlerin yapılması sonucunda, ilk günkü düzenliliklerini yitirirler. Bu nedenle, indekslerin dağınıklık durumlarını (fragmentation) gerektikçe gözden geçirmek ve bozulmalar olduğunda yeniden düzenlemek gerekir. İndeks dağınıklığını görüntülemek ve tespiti için sys.dm\_db\_index\_physical\_stats objesi kullanılabilir. Tüm veritabanı genelinde dağılan indekslerin oranlarının gözlemlenmesi için aşağıdaki SQL sorgusu kullanılabilir.

```
SELECT OBJECT_NAME(ind.OBJECT_ID) AS TableName,  
ind.name AS IndexName, indexstats.index_type_desc AS IndexType, indexstats.avg_fragmentation_in_percent  
FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL, NULL) indexstats  
INNER JOIN sys.indexes ind ON ind.object_id = indexstats.object_id AND ind.index_id = indexstats.index_id  
ORDER BY indexstats.avg_fragmentation_in_percent DESC
```

## Fragmente olan indexlerin tespiti sonrası hangi senaryolar uygulanmalıdır?

Zamanla oluşacak düzensizliklerin önüne geçmek için, indeksleri yeniden düzenleme (defragmentasyon) yapmak gerekebilir. Bunun için takip edilebilecek iki yöntem vardır. İndeksler çeşitli yöntemlerle yeniden oluşturulabilir (Rebuild) veya yeniden düzenlenebilir (ReOrganize). Defragmentasyon yapılırken dağınık indekslerin hangi yöntem kullanılarak rebuild veya reorganize edileceğine aşağıdaki oranlar dikkate alınarak karar verilebilir.

**Fragmente Oranı** > 5% and < = 30% - ReOrganize

**Fragmente Oranı** > 30% - ReBuild

## SQL Server'da İstatistik kavramı nedir?

SQL Server tarafından, her bir indexe dair istatistikler tutulur. Bu istatistikler, indeksin ne kadar kullanışlı olabileceğine dair bilgiler oluşturulurken kullanılır ve SQL Server Query Optimizer denilen sunucu uygulama tarafından sorgu planı oluşturma aşamasına kullanılır. İstatistikler tablonun bulunan verilerin dağılımını gösterir ve sorgu planı oluştururken kullanılırlar.

## İstatistiklerin veri erişimine ve SQL server performansına etkileri nelerdir?

İstatistikler sayesinde, sorgu planı (Query Plan) oluşturulurken sorgu sonucunda ulaşılacak tahmini kayıt sayısı bulunur. Bulunan değerler indekslere erişim şeklini belirler. Bu sayede en az maliyetli sorgu planının çalıştırılması sağlanır. İstatistiklerin temel amacı, datayı en hızlı ve en az maliyetli şekilde kullanıcının karşısına getirmektir.

## İstatistikler hakkında nasıl bilgi edinilebilir?

SQL Server'da bulunan istatistiklere dair bilgi almak için aşağıdaki yöntemler kullanılabilir.

STATS\_DATE(OBJECT\_ID, STATS\_ID) komutu ile indeks hakkında tarih bilgisine ulaşılabilir.

Belirli bir indeks hakkında daha fazla bilgi almak için aşağıdaki komutlar kullanılabilir:

- DBCC SHOW\_STATISTICS (TABLE\_NAME,INDEX\_NAME) komutu kullanılabilir.
- Belirli bir tabloda yer alan istatistiklerin son güncelleme zamanının bulunması için aşağıdaki SQL komutu kullanılabilir.  
SELECT name AS stats\_name,  
STATS\_DATE(object\_id, stats\_id) AS statistics\_update\_date FROM sys.stats  
WHERE object\_id = OBJECT\_ID('TBLCASABIT');

### İstatistiklerin güncel olması nasıl sağlanır?

İstatistikleri güncellemek için 3 farklı yöntem kullanılabilir.

**FULLSCAN:** Bu yöntemde istatistiğin bağlı olduğu tablo ya da indexed view'daki tüm satırlar taranarak istatistik güncellenir.

UPDATE STATISTICS TBLCASABIT TBLCASABIT\_IND\_KOD WITH FULLSCAN

**SAMPLE:** Bu yöntemde sample ifadesinden sonra belirtilen satır sayısı ya da yüzde oranı kadar satır örnek alınarak istatistik güncellenir.

UPDATE STATISTICS TBLCASABIT TBLCASABIT\_IND\_KOD WITH SAMPLE 1000 ROWS

**RESAMPLE:** Bu yöntemde istatistiğin en son örnek alınan oranı baz alınarak güncellenir. UPDATE STATISTICS TBLCASABIT TBLCASABIT\_IND\_KOD WITH RESAMPLE

Veritabanında bulunan tüm tablolardaki indekslerin güncellenmesi için SP\_UPDATESTATS prosedürü kullanılabilir.

### Query Optimizer sorguların çalıştırma yöntemlerini belirlerken hangi bilgileri kullanır?

SQL Server'ın performans hesaplama temelleri maliyet bazlı optimizasyona dayanır. SQL Server'da Query Optimizer, SQL sorgularının çalıştırılması ile ilgili alternatifler yöntemleri inceleyerek karar veren bir veritabanı optimizasyon bileşenidir. Query Optimizer tarafından karar verilirken veri dağılım şekli ve sorguların maliyetleri oldukça önemlidir. Verilerin kullanımında en çok ihtiyaç duyulan indekslerdir. İndekslerin de etkili bir şekilde kullanılmasında istatistiklerin doğru yapılandırılması ve istatistik bilgilerinin güncel olması çok önemlidir. Doğru ve güncel istatistikler olmadığında, SQL Server sorgularındaki satır sayısı tahminlemesi de (estimated row number) yanlış yapılacaktır. Bunun sonucunda gereksiz IO kullanımı oraya çıkarak sorgu sonucu uzun süren bir işleme girecektir. Dolayısıyla Query Optimizer tarafından sorgu çalıştırma planları oluşturulurken indeks ve istatistik bakımlarının yapılıyor olması oldukça önemlidir.

### SQL Server'da indekslerin bakımı ve istatistiklerin güncellenmesi periyodik olarak planlanabilir mi?

SQL Server Maintenance Plan ile SQL'de bulunan indeks ve istatistiklerin bakımı için bir plan oluşturulabilir ve oluşturulan plan için bir görev zamanlayıcısı belirlenerek istenilen periyotlarda çalıştırılması sağlanabilir. Aynı zamanda Netsis 9.0.27 sürümü ile birlikte gelen Zamanlanmış Görevler eklentisi üzerinden de indeks ve istatistik bakım planlarının oluşturulması mümkün hale gelmiştir.

Bunun için; Zamanlanmış Görevler eklentisi ön tanımlı işlemler altında bulunan Veritabanı Bakım İşlemleri seçeneği tanımlanarak istenilen tablo bazında veya tablo parametresi belirlenmediğinde tüm veritabanı tablolarında index bakımı yapılabilir. Bu işlem ile birlikte dağılım oranı %30 üzerindeki indeksler için Rebuild, %5 ile %30 arasındaki indeksler için ReOrganize işlemi gerçekleştirilmekte olup aynı zamanda istatistik güncelle parametresi ile mevcut istatistiklerin güncellenmesi sağlanabilir.