

İnceleme Soruları:

1)SQL performans ayarlaması nedir?

SQL performans ayarlaması (tuning), veritabanı sorgularının daha hızlı ve verimli çalışmasını sağlamak için yapılan optimizasyon işlemleridir. Bu süreç; indeksleme, sorgu optimizasyonu, gereksiz veri getirmenin önlenmesi, veritabanı tasarımının iyileştirilmesi ve donanım kaynaklarının etkin kullanımı gibi teknikleri içerir. Amaç, sistemin yanıt süresini kısaltmak ve kaynak tüketimini azaltmaktır.

2) Veritabanı performans ayarlaması nedir?

Veritabanı performans ayarlaması (tuning), veritabanının daha hızlı ve verimli çalışmasını sağlamak için yapılan optimizasyon işlemleridir. Bu süreçte indeksleme, sorgu optimizasyonu, bellek yönetimi, veri tablosu düzenlemeleri ve önbellekleme teknikleri gibi yöntemler kullanılır. Amaç, sorguların daha kısa sürede çalışmasını ve sistem kaynaklarının en verimli şekilde kullanılmasını sağlamaktır.

3) Performans ayarlama faaliyetlerinin çoğunun odak noktası nedir ve bu odak noktası neden vardır?

Performans ayarlama faaliyetlerinin odak noktası, çalışanların ve organizasyonların daha verimli ve etkili bir şekilde çalışabilmesini sağlamak için performanslarını artırmaktır. Bu odak noktası, verimliliği artırma, hedeflere ulaşma, çalışan motivasyonunu yükseltme ve beceri gelişimini destekleme amacını taşır. Yani, organizasyonun başarısını iyileştirmek için sürekli performans izleme ve geliştirme süreci uygulanır.

4) Veritabanı istatistikleri nedir ve neden önemlidir?

Veritabanı istatistikleri, veritabanındaki veri dağılımı, indeksler, sorgu süreleri gibi performansla ilgili bilgileri içeren verilerdir. Bu istatistikler, veritabanı yönetim sisteminin (DBMS) sorguları daha hızlı ve verimli bir şekilde işlemesine yardımcı olur.

Önemi şunlardır:

- **Performans İyileştirme:** Veritabanı sorguları daha hızlı çalışır, çünkü DBMS, verilerin nasıl düzenlendiğini ve hangi yollarla erişildiğini bilir.
- **Verimli Sorgu Planları:** İstatistikler, sorgu planlarının daha verimli seçilmesine yardımcı olur.
- **Kaynak Yönetimi:** Veritabanının daha dengeli ve etkin bir şekilde kaynak kullanmasını sağlar.

Kısaca, veritabanı istatistikleri, veritabanı performansını optimize etmek ve sistemin verimli çalışmasını sağlamak için çok önemlidir.

5) Veritabanı istatistikleri nasıl elde edilir?

Veritabanı istatistikleri, genellikle otomatik olarak veya manuel olarak güncellenir. Otomatik olarak, DBMS veri değişikliklerini izler ve istatistikleri günceller. Manuel olarak ise veritabanı yöneticileri, belirli komutlar (örneğin, `UPDATE STATISTICS` veya `ANALYZE`) kullanarak istatistikleri elde edebilir. Bu istatistikler, sorgu performansını iyileştirmek için kullanılır.

6) Tablolar, dizinler ve kaynaklar için tipik veritabanı istatistik ölçümleri nelerdir?

Tablolar İçin İstatistikler:

- Satır Sayısı: Tabloyu oluşturan toplam satır sayısı.
- Veri Dağılımı: Veri değerlerinin dağılımı, boş değerler ve sıklıklar.
- Sayfa Sayısı: Tabloyu depolayan fiziksel veri sayfalarının sayısı.

Dizinler İçin İstatistikler:

- Dizin Boyutu: Dizin yapısının fiziksel boyutu.
- Dizin Etkinliği: Dizinle yapılan arama, ekleme, silme işlemlerinin sıklığı.
- Kök Düzümü Yüksekliği: Dizin yapısındaki en üst seviyenin yüksekliği, erişim hızını etkiler.

Kaynaklar İçin İstatistikler:

- I/O İstatistikleri: Disk okuma ve yazma işlemleri.
- CPU Kullanımı: Veritabanı işlemleri için harcanan işlemci zamanı.
- Bellek Kullanımı: Veritabanının ihtiyaç duyduğu bellek miktarı.

7) SQL DDL deyimlerinin (`CREATE TABLE` gibi) işlenmesi DML deyimlerinin gerektirdiği işleminden nasıl farklıdır?

DDL (`CREATE`, `ALTER`, `DROP`) komutları veritabanı yapısını değiştirir ve **otomatik commit edilir**, geri alınamaz.

DML (`INSERT`, `UPDATE`, `DELETE`, `SELECT`) komutları ise verileri değiştirir ve **transaction içinde çalışarak geri alınabilir**.

DDL yapıyı, DML ise veriyi yönetir.

8) Basit bir ifadeyle, DBMS bir sorguyu üç aşamada işler. Aşamalar nelerdir ve her aşamada ne gerçekleştirilir?

DBMS bir sorguyu üç aşamada işler:

1. **Analiz ve Optimizasyon:** Sorgunun doğruluğu kontrol edilir ve en iyi yürütme planı belirlenir.
2. **Derleme ve Planlama:** Sorgu, belirlenen plana göre yürütülecek şekilde hazırlanır.
3. **Yürütme ve Sonuç Üretme:** Sorgu çalıştırılır ve sonuçlar kullanıcıya döndürülür.

9) İndeksler bu kadar önemliyse, neden her tablodaki her sütunu indekslemiyoruz? (Veri seyrekliğinin oynadığı rol hakkında kısa bir tartışma ekleyin).

Her sütunu indekslemiyoruz çünkü:

1. **Performans Kaybı:** Güncelleme, ekleme ve silme işlemlerini yavaşlatır.
2. **Depolama Maliyeti:** Fazla indeks gereksiz yer kaplar.
3. **Sorgu Optimizasyonu:** Gereksiz indeksler sorgu planlamasını zorlaştırır.

Veri Seyrekliğinin Rolü:

- **Seyrek (farklı değerler çoksa):** İndeks faydalıdır.
- **Yoğun (tekrarlayan değerler çoksa):** İndeks pek etkili olmaz.

Bu yüzden sadece **sık sorgulanan ve seçici sütunlar** indekslenmelidir.

10) Kural tabanlı bir optimize edici ile maliyet tabanlı bir optimize edici arasındaki fark nedir?

- **Kural Tabanlı Optimize Edici (Rule-Based Optimizer - RBO):**
 - Sabit kurallar kullanarak sorgu planını belirler. Örneğin, indeks varsa kullan, birleşim sırasını değiştirme gibi. **Veri miktarı veya maliyet hesaplaması yapmaz.**
- **Maliyet Tabanlı Optimize Edici (Cost-Based Optimizer - CBO):**

Veritabanı istatistiklerini kullanarak sorgu planlarının **maliyetini hesaplar** ve en düşük maliyetli olanı seçer. **Daha esnek ve genellikle daha verimlidir.**

11) Optimize edici ipuçları nedir ve nasıl kullanılır?

Optimize edici ipuçları (Query Hints), SQL sorgularının nasıl çalışacağını yönlendiren komutlardır.

`/*+ HINT */` şeklinde sorguya eklenir.

```
SELECT /*+ INDEX(emp emp_idx) */ * FROM emp WHERE dept_id = 10;
```

“Belirtilen indeksin (emp_idx) kullanılmasını zorunlu kılar.”

Ne İşe Yarar?

- İndeks kullanımını yönlendirme
- Birleşim (JOIN) yöntemini belirleme
- Sorgu paralelleştirmeyi sağlama

12) Dizinlerin oluşturulması ve kullanılmasına ilişkin bazı genel kurallar nelerdir?

- **Sık kullanılan sütunlarda indeks oluştur:** WHERE, JOIN gibi sıkça sorgulanan sütunlar için indeks ekleyin.
- **Veri seyrekliğine dikkat et:** Seyrek veri (farklı değerler) için indeks daha faydalıdır.
- **Birleştirilmiş indeksler kullan:** Birden fazla sütun bir arada kullanılıyorsa, birleşik indeksler yararlı olabilir.
- **Okuma işlemleri için hızlandırır, yazma işlemlerini yavaşlatır:** İndeksler sadece okuma (SELECT) işlemlerini hızlandırır.
- **İndeks sayısını sınırlı tut:** Çok fazla indeks yazma işlemlerini zorlaştırabilir.

13) Çoğu sorgu optimizasyon tekniği, optimize edicinin işini kolaylaştırmak için tasarlanmıştır. SQL kodunda koşullu ifadeler yazmayı düşünüyorsanız hangi faktörleri aklınızda tutmalısınız?

SQL kodunda koşullu ifadeler yazarken aklınızda tutmanız gereken bazı faktörler şunlardır:

1. **İndeks Kullanımı:** Koşullu ifadelerde kullanılan sütunların indekslenmiş olması, sorgu performansını artırır.
2. **Veri Seyrekliğine Dikkat Et:** Koşulda sık kullanılan, daha az tekrar eden verilere öncelik verin.
3. **NULL Değerleri:** NULL değerler ile yapılan karşılaştırmalar bazen sorgu performansını olumsuz etkileyebilir, dikkatli olun.
4. **Koşul Sıralaması:** Daha seçici koşulları (daha az veri döndüren) önce yazmak, sorgu hızını artırabilir.
5. **Ağır İşlemlerden Kaçın:** Koşullarda karmaşık hesaplamalardan veya alt sorgulardan kaçınarak sorguyu basit tutmaya çalışın.

14) Çok sayıda tablo ve izin içeren bir VTYS'de veri dosyalarını yönetmek için ne öneride bulunursunuz?

Çok sayıda tablo ve izin içeren bir VTYS'de veri dosyalarını yönetmek için şu önerilere dikkat edebilirsiniz:

1. **Veritabanı Tasarımını İyi Yap:** Tabloları ve dizinleri doğru şekilde normalleştir, gereksiz tekrarları engelle.
2. **İndeksleri Optimize Et:** Sadece sık kullanılan sütunlarda indeks oluştur, gereksiz indekslerden kaçın.
3. **Dizin Bakımını Yap:** Dizinleri düzenli olarak yeniden oluştur ve güncelle, böylece sorgu performansı korunur.
4. **Veri Dağıtımını İyi Yap:** Veri dosyalarını mantıklı bir şekilde dağıtarak disk alanını verimli kullan.
5. **Yedekleme ve Kurtarma Stratejisi:** Veritabanı yedeklemelerini düzenli yap, veri kaybını önlemek için kurtarma planları oluştur.

15) RAID ne anlama gelir ve yaygın olarak kullanılan bazı RAID seviyeleri nelerdir?

RAID (Redundant Array of Independent Disks), veri güvenliği ve performansı artırmak için birden fazla diski birleştiren bir teknolojidir. Yaygın RAID seviyeleri şunlardır:

1. **RAID 0:** Verileri hız için birden fazla diske bölerek yazar, ama veri güvenliği yoktur.
2. **RAID 1:** Verileri iki diske aynı anda yazar, veri güvenliği sağlar ama kapasiteyi yarıya indirir.
3. **RAID 5:** Verileri ve pariteyi dağıtarak hem hız hem de veri güvenliği sağlar, bir disk arızasında veri kurtarılabilir.
4. **RAID 10:** RAID 1 ve RAID 0'ın birleşimi, yüksek hız ve veri güvenliği sağlar ama maliyeti yüksektir.