

Sorgu Optimizasyonu ve Yürütme Planları Nedir?

Çalışma Kağıdı

Sorgu optimizasyonu, bir veritabanı sorgusunun en az kaynak (zaman, disk G/Ç) kullanarak tamamlanmasını sağlayan bir süreçtir. Yürütme planı ise bu sorguyu gerçekleştirmek için veritabanı tarafından izlenecek adımları belirten bir yol haritasıdır.

Sorular

1. Bir sorgunun veritabanı tarafından nasıl çalıştırılacağını belirleyen yol haritası aşağıdakilerden hangisidir?

- A) Sorgu Ayrıştırıcı
- B) Yürütme Planı
- C) İndeks Yapısı
- D) Veritabanı İstatistikleri

2. Sorgu optimizasyonunun temel amacı nedir?

- A) Sorgu yazımını kolaylaştırmak
- B) Veritabanı sunucusunun yükünü artırmak
- C) Sorguların en az kaynakla çalıştırılmasını sağlamak
- D) Yeni indeksler oluşturmak

3. Optimizatör, yürütme planlarını seçerken hangi bilgileri kullanır?

- A) Sadece kullanıcı adı
- B) Veritabanı istatistikleri ve indeks bilgileri
- C) Sorgunun metin uzunluğu
- D) Sunucu RAM miktarı

4. Basit bir SELECT sorgusu için yürütme planı nasıl oluşur?

5. JOIN işlemi içeren bir sorguda optimizasyon neden önemlidir?

6. Tanımla: Sorgu Optimizasyonu

7. Tanımla: Yürütme Planı

8. Tanımla: Veritabanı İstatistikleri

Cevap Anahtarı

1. B) Yürütme Planı — Yürütme planı, veritabanı sisteminin bir sorguyu yerine getirmek için izleyeceği adımları tanımlar.
2. C) Sorguların en az kaynakla çalıştırılmasını sağlamak — Sorgu optimizasyonu, sorguların daha hızlı ve daha az kaynak kullanarak tamamlanmasını hedefler.
3. B) Veritabanı istatistikleri ve indeks bilgileri — Optimizatör, veritabanı istatistikleri (tablo boyutları, veri dağılımı vb.) ve mevcut indeksler hakkında bilgi kullanarak en uygun planı belirler.
4. 1. Sorgu ayrıştırılır ve sözdizimsel olarak doğrulanır. 2. Sorgu, veritabanı istatistikleri ve indeks bilgileri kullanılarak analiz edilir. 3. Optimizatör, olası yürütme planlarını değerlendirir (örn. Tablo tarama, indeks tarama). 4. En verimli olduğu düşünülen plan seçilir ve yürütülür.
5. JOIN işlemi, iki veya daha fazla tabloyu birleştirdiği için maliyetli olabilir. Sorgu optimizasyonu, en uygun JOIN algoritmasını (örn. Nested Loop Join, Hash Join, Merge Join) ve tablo birleştirme sırasını seçerek performansı önemli ölçüde artırır.
6. Veritabanı sorgularının en verimli şekilde çalıştırılmasını sağlayan süreç.
7. Bir sorguyu yerine getirmek için veritabanının izleyeceği adımlar dizisi.
8. Tablolar ve indeksler hakkında optimizasyon kararlarına yardımcı olan bilgiler (örn. satır sayısı, benzersiz değerler).

Bounlu

Tüm kartlar, adım adım çözümler ve AI hoca desteği Notek uygulamasında.
Sınav tarihlerini Promy otomatik hatırlatıcıya çevirir.