

Problem 1: EMP_SEX sütununun olası veri sıklığı nedir?

Cevap: EMP_SEX sütunu, çalışanların cinsiyetini temsil eder. Genellikle iki değer alır: 'E' (erkek) ve 'K' (kadın). Veri sıklığı, her bir değerın dağılımına bağlıdır. Eğer çalışanların yaklaşık yarısı erkek ve yarısı kadınsa, veri sıklığı düşüktür. Ancak, eğer çalışanların büyük çoğunluğu tek bir cinsiyetten ise (örneğin, %90'ı erkek), o zaman veri sıklığı yüksektir.

Problem 2: Hangi indeksleri oluşturmalsınız? Gerekli SQL komutlarını yazın.

Cevap: Sorgu, EMP_SEX ve EMP_AREACODE sütunlarına filtre uyguluyor. Bu nedenle, bu sütunlarda indeks oluşturmak sorgu performansını artırabilir.

SQL

```
CREATE INDEX idx_emp_sex ON ÇALIŞAN (EMP_SEX);  
CREATE INDEX idx_emp_areacode ON ÇALIŞAN (EMP_AREACODE);
```

Problem 3: İki alternatif erişim planı oluşturun.

Cevap:

Plan 1:

1. ÇALIŞAN tablosunu tara.
2. EMP_SEX = 'K' ve EMP_AREACODE = '615' filtrelerini uygula.
3. EMP_LNAME ve EMP_FNAME'e göre sırala.

Plan 2:

1. idx_emp_sex indeksini kullanarak kadın çalışanları bul.
2. idx_emp_areacode indeksini kullanarak 615 alan kodlu çalışanları bul.
3. İki sonucun kesişimini al.
4. EMP_LNAME ve EMP_FNAME'e göre sırala.

Problem 4: EMP_DOB sütununun olası veri sıklığı nedir?

Cevap: EMP_DOB sütunu, çalışanların doğum tarihlerini temsil eder. Her çalışanın farklı bir doğum tarihi olması muhtemeldir. Bu nedenle, veri sıklığı yüksektir.

Problem 5: EMP_DOB üzerinde bir dizin oluşturmalsınız mı? Neden ya da neden olmasın?

Cevap: EMP_DOB üzerinde bir dizin oluşturmak genellikle faydalı değildir. Çünkü veri seyrekliği yüksek olduğu için, dizin kullanmak yerine tablonun tamamını taramak daha hızlı olabilir. Ancak, eğer sorgular genellikle belirli bir doğum tarihi aralığını filtreliyorsa, o zaman bir dizin faydalı olabilir.

Problem 6: Sorgu tarafından muhtemelen ne tür veritabanı G/Ç işlemleri kullanılacaktır?

Cevap: Sorgu, ÇALIŞAN tablosunu tarayacak ve her satır için YEAR(EMP_DOB) fonksiyonunu hesaplayacaktır. Bu, sıralı tarama (sequential scan) ve fonksiyon değerlendirme (function evaluation) işlemlerini içerir.

Problem 7: VTYS ne tür bir optimizasyon kullanacaktır?

Cevap: Tablo istatistikleri yoksa, VTYS kural tabanlı optimizasyon kullanacaktır. Bu, sorgu optimizasyonunun önceden tanımlanmış kurallara göre yapıldığı anlamına gelir.

Problem 8: Sorgu tarafından muhtemelen ne tür veritabanı G/Ç işlemleri kullanılacaktır?

Cevap: Sorgu, ÜRÜN tablosunu tarayacak, her satır için P_PRICE değerini ortalama P_PRICE ile karşılaştıracak ve sonucu filtreleyecektir. Bu, sıralı tarama, toplama (aggregation) ve filtreleme işlemlerini içerir.

Problem 9: P_PRICE sütununun olası veri seyrekliği nedir?

Cevap: P_PRICE sütunu, ürünlerin fiyatlarını temsil eder. Fiyatlar genellikle belirli bir aralıkta değişir, ancak her ürünün farklı bir fiyatı olabilir. Bu nedenle, veri seyrekliği orta veya yüksek olabilir.

Problem 10: Bir dizin oluşturmalı mısınız? Neden ya da neden olmasın?

Cevap: Eğer sorgular genellikle P_PRICE'a göre filtreleme yapıyorsa, bir dizin oluşturmak faydalı olabilir. Ancak, eğer sorgular nadiren P_PRICE'ı kullanıyorsa, dizin oluşturmak gereksiz olabilir.

Problem 11: LINE_UNITS sütununun olası veri seyrekliği nedir?

Cevap: LINE_UNITS sütunu, satılan ürünlerin birim sayılarını temsil eder. Birim sayıları genellikle belirli bir aralıkta değişir, ancak her satışın farklı bir birim sayısı olabilir. Bu nedenle, veri seyrekliği orta veya yüksek olabilir.

Problem 12: Bir dizin oluşturmalı mısınız? Eğer öyleyse, dizin sütun(lar)ı ne olurdu ve dizini neden oluştururdunuz?

Cevap: Eğer sorgular genellikle LINE_UNITS'e göre filtreleme veya sıralama yapıyorsa, bir dizin oluşturmak faydalı olabilir. Dizin sütunu LINE_UNITS olurdu.

Problem 13: P_CODE üzerinde bir dizin oluşturmalı mısınız? Eğer öyleyse, dizini oluşturmak için SQL komutunu yazın.

Cevap: Eğer sorgular genellikle P_CODE'a göre gruplama veya filtreleme yapıyorsa, bir dizin oluşturmak faydalı olabilir.

SQL
CREATE INDEX idx_p_code ON HAT (P_CODE);

Problem 14: Bu tablo için istatistik oluşturacak komutu yazınız.

Cevap:

SQL
ANALYZE TABLE HAT;

Problem 15: P_QOH ve P_PRICE sütunlarının olası veri sıklığı nedir?

Cevap: P_QOH (eldeki miktar) ve P_PRICE (fiyat) sütunları, ürünlerin özelliklerini temsil eder. Her ürünün farklı bir miktarı ve fiyatı olabilir. Bu nedenle, her iki sütunun da veri sıklığı yüksek olabilir.

Problem 16: Bir dizin oluşturmalı mısınız? Eğer öyleyse, dizin sütun(lar)ı ne olacaktır ve dizini neden oluşturmalısınız?

Cevap: Eğer sorgular genellikle P_QOH ve P_PRICE'a göre filtreleme yapıyorsa, bu sütunlarda birleşik bir dizin oluşturmak faydalı olabilir.

Problem 17: Hangi indeksleri oluşturmalısınız ve neden? İndeksleri oluşturmak için SQL komutunu yazın.

Cevap: Sorgu, V_STATE sütununa filtre uyguluyor ve V_NAME'e göre sıralama yapıyor. Bu nedenle, bu sütunlarda indeks oluşturmak sorgu performansını artırabilir.

SQL
CREATE INDEX idx_v_state ON SATICI (V_STATE);
CREATE INDEX idx_v_name ON SATICI (V_NAME);

Problem 18: Sorgu tarafından satırların yüzde kaç döndürülecektir?

Cevap: Tablo P11.18'e göre, 10.000 satıcının 113'ü 'TN' eyaletinde. Bu nedenle, sorgu satırların %1.13'ünü döndürecektir.

Problem 19: Sorguyu yürütmek için büyük olasılıkla ne tür I/O veritabanı işlemleri kullanılır?

Cevap: Sorgu, SATICI tablosunu tarayacak, V_STATE'i filtreleyecek ve V_NAME'e göre sıralayacaktır. Bu, sıralı tarama, filtreleme ve sıralama işlemlerini içerir.

Problem 20: İki alternatif erişim planı oluşturun.

Cevap:

Plan 1:

1. SATICI tablosunu tara.
2. V_STATE = 'TN' filtresini uygula.
3. V_NAME'e göre sırala.

Plan 2:

1. idx_v_state indeksini kullanarak 'TN' eyaletindeki satıcıları bul.
2. V_NAME'e göre sırala.

Problem 21: Hangi endeksleri önerirsiniz?

Cevap: Sorgu, V_STATE ve V_AREACODE sütunlarına filtre uyguluyor ve P_PRICE'a göre sıralama yapıyor. Bu nedenle, bu sütunlarda indeks oluşturmak sorgu performansını artırabilir.

Problem 22: Problem 21'de önerdiğiniz dizinleri oluşturmak için gereken komutları yazın.

Cevap:

SQL

```
CREATE INDEX idx_v_state ON SATICI (V_STATE);  
CREATE INDEX idx_v_areacode ON SATICI (V_AREACODE);  
CREATE INDEX idx_p_price ON ÜRÜN (P_PRICE);
```

Problem 23: ÜRÜN ve SATICI tablolarının istatistiklerini oluşturmak için kullanılan komut(lar)ı yazın.

Cevap:

SQL

```
ANALYZE TABLE ÜRÜN;  
ANALYZE TABLE SATICI;
```

Problem 24: Aşağıdaki sorguya dayalı olarak hangi dizini önerirsiniz ve bunu oluşturmak için hangi komutu kullanırsınız?

SQL

```
SEÇİNİZ P_CODE, P_DESCRIPT, P_QOH, P_PRICE, V_CODE  
ÜRÜN  
NEREDE V_CODE = '21344'  
ORDER BY P_CODE;
```

Cevap: Sorgu, V_CODE'a göre filtreleme yapıyor ve P_CODE'a göre sıralama yapıyor. Bu nedenle, bu sütunlarda indeks oluşturmak sorgu performansını artırabilir.

SQL

```
CREATE INDEX idx_v_code ON ÜRÜN (V_CODE);  
CREATE INDEX idx_p_code ON ÜRÜN (P_CODE);
```

Problem 25: Sorguyu yeniden yazmak ve gerekli sonuçları daha verimli bir şekilde üretmek için Bölüm 11-5b'de verilen önerileri kullanın.

SQL

```
SEÇİNİZ P_CODE, P_DESCRIPT, P_QOH, P_PRICE, V_CODE  
ÜRÜN  
NEREDE P_QOH < P_MIN AND P_MIN = P_REORDER AND P_REORDER = 50  
ORDER BY P_QOH;
```

Cevap:

SQL

```
SEÇİNİZ P_CODE, P_DESCRIPT, P_QOH, P_PRICE, V_CODE  
ÜRÜN  
NEREDE P_QOH < 50  
ORDER BY P_QOH;
```

Problem 26: Hangi indeksleri önerirsiniz? Bu indeksleri oluşturmak için gerekli komutları yazınız.

Cevap: Sorgu, P_QOH'a göre filtreleme ve sıralama yapıyor. Bu nedenle, bu sütunda bir indeks oluşturmak sorgu performansını artırabilir.

SQL

```
CREATE INDEX idx_p_qoh ON ÜRÜN (P_QOH);
```

Problem 27: Ayda 15.000 fatura ürettiğinizi varsayarsak, tasarımcıya türetilmiş niteliklerin kullanımı hakkında ne önerirsiniz?

Cevap: Türetilmiş nitelikler, hesaplama gerektirdiği için performans üzerinde olumsuz etki yaratabilir. Bu nedenle, sık kullanılan türetilmiş nitelikleri önceden hesaplayıp saklamak veya sorguları optimize etmek önerilir.

Problem 28: Problem 27'de verdiğiniz tavsiyelere uyduğunuzu varsayarsak, sorguyu nasıl yeniden yazardınız?

Cevap:

SQL

```
SEÇİNİZ CUS_CODE, MAX(LINE_UNITS * LINE_PRICE)
MÜŞTERİ
DOĞAL BİRLEŞTİRME FATURA
DOĞAL BİRLEŞTİRME HAT
NEREDE CUS_AREACODE = '615'
GROUP BY CUS_CODE;
```

Problem 29: Problem 28'de yazdığınız sorgu için hangi indeksleri önerirsiniz ve hangi SQL komutlarını kullanırsınız?

Cevap: Sorgu, CUS_AREACODE'a göre filtreleme ve CUS_CODE'a göre gruplama yapıyor. Bu nedenle, bu sütunlarda indeks oluşturmak sorgu performansını artırabilir.

SQL

```
CREATE INDEX idx_cus_areacode ON MÜŞTERİ (CUS_AREACODE);
CREATE INDEX idx_cus_code ON MÜŞTERİ (CUS_CODE);
```