VTSG EV ÖDEVİ #1

En fazla 3 kişilik grup ödevidir. Çalışmada yapılan işlemler ve kullandığınız SQL ifadeleri için kısa ve ÖZ açıklama yapınız. Gerek görülürse değerlendirme esnasında kısa görüşmeler yapılabilir. İlk 2 soru 15+15 toplam 30 puan. Diğer 7 soru her biri 10 puandır.

Şeması **T(a varchar(40), b int, c bool)** olan HEAP dosyası **2 milyon kayıt** içersin. b-niteliği değer aralığı **[0,1.5 milyon]** olmak üzere; *searchkey*= b niteliği üzerinde bir B+-tree indeks düşünelim. Böylece yapraklarda **<b,rid>** *(rid, searchkey=b olan kaydın rid'si olmak üzere)* ikilileri saklanıyor.

1- Bahsedilen tabloyu üretin (hariçte *python sonra COPY komut ile*, veya dahilde *generate_series()* yöntemleri ile bunu yapabilirsiniz). Ortaya çıkan T tablosunun **büyüklüğü (MB), içerdiği disk sayfa sayısı**, ve **b niteliğine ait sütun istatistiklerini** inceleyin (Bunlar için sistem görüntü fonksiyonlarını kullanabilirisiniz). `pg_attribute` sistem kataloğunu da sorgulayarak dosyanın **sabit / değişken uzunluklu kayıt dosyası olma durumunu** öğrenin.

2- Soruda bahsedilen B+-tree indeksi PgSQL'de oluşturun: Indeks yükleme esnasında `\timing` komutu ile indeks yükleme gecikmesi değerini ölçün. Ortaya çıkan ağacın yüksekliğini, her seviyedeki sayfa sayısını ve ağacın yüksekliğini, `pg_stat_all_indexes` sistem görünümünü kullanarak öğrenin, diğer indeks istatistiklerini inceleyin, yorumlayın. Mesela, **[0,1.5 milyon]** değer aralığında **2 milyon** kayıt ürettiğimiz için indekste tekrar değerler olmalı. Bunların sayısı ne kadar? Bunu ilk olarak T tablosunda bir SQL ile bulun. Sonra sistem katalog sorguları ile bulun. Sistem katalog ile bulduklarınız yanlış mı? O zaman sistem kataloglarını güncelleyip T üzerinde SQL ile bulduklarınızla aynı değerleri elde edin..

3- **(Yer verimliliği düşük bir indeks)** 2. soruda istenilenleri, ağaç doluluk değerinin **en fazla %60 olması** durumu için tekrar bulun. Bunun için `fillfactor` değerini kullanabilirsiniz. `fillfactor` ile bütün düğümler %60 ı geçmeyen bir ağaç oluşturabildiniz mi? 2. sorudaki değerler nasıl değişti? Niye böyle (yer verimliliği düşük olan) bir ağaç oluşturmak isteriz ki?

4- **(Sıkışık veri seti)** Aynı b değer aralığında fakat – düzenli dağılım göstermeyen-- daha sıkışık bir veri seti için indeksi oluşturun. (`fillfactor`, varsayılan değerinde kalsın.) Mesela 500.000 – 600.000 arasında daha yoğun; diğer aralıklarda daha seyrek veri olsun. (Bunu gene *generate_series()veya* hariçte *python* ile de yapabilrisniz..)
2. sorudaki değerler nasıl değişti? Nedenlerini yorumlayın. Niye böyle bir analiz yaptık..

5- **1 ve 2. soruda** sistemden elde ettiğiniz sonuçları "analitik olarak" (kendi hesaplarınızla) doğrulayın. Mesela ağacın yüksekliği, yaprak ve her seviyedeki düğüm sayıları gibi değerleri analitik olarak bulun. (`fillfactor` varsayılan değerinde kalsın ve veri dağılımı sıkışık olmasın. 3. soru ve 4. sorudaki istisna durumlar için analitik olarak göstermenize gerek yok)

6- **Soruda bahsedilen B+-tree indeksi varken ve katalog bilgileri de güncelken** aşağıdaki sorgu için sistemin **indeks kullandığı** k1 ve k2 değerleri belirleyin. Indeks kullandığı **k2-k1 maksimum** değeri nedir?
SELECT a FROM T WHERE b between **k1** and **k2**;

7- **Soruda bahsedilen B+-tree indeksi varken ve katalog bilgileri de güncelken** aşağıdaki sorgunun EXPLAIN çıktısını inceleyin ve yorumlayın.
SELECT count (distinct b) FROM T;

8- **Soruda bahsedilen B+-tree indeksi varken ve katalog bilgileri de güncelken** aşağıdaki sorgunun EXPLAIN çıktısını inceleyin ve yorumlayın.
SELECT b, count(a) FROM T WHERE a >'c' GROUP BY b;

9- **(Harici sıralama)** Sorudaki özellikleri verilen T tablosunu T-sorted isimli başka bir tabloda **a-niteliğine** göre sıralı olarak saklamak istiyoruz. Bunu SQL ile gerçekleyin. Bu SQL ifadesini EXPLAIN ile analiz edin. Harici sıralama yapıldığını görün. Eğer yapılımıyorsa *work_mem* değerini düşürün. Harici sırlama ile T_sorted tablosu elde edilmesindeki gecikme ne kadar oldu? Bu değeri analitik olarak elde edebilir misiniz? Tam olmasa da bu değerin olabiliriliğini analitik olarak gösterin. Mevcut hafıza (work_mem) ile kaç iterasyon gerekiyor? Sistemde yapılan harici sıralamada k-way merge de k değeri nedir?