

KURALLAR:

1. Ödevde 10 soru olup, her soru 1 puandır. Bu ödevin dönem notuna etkisi %10 dur. Yani alınan her puan dönem notuna 1 puan olarak yansır.
2. Ödevde kopya çekmek disiplin suçudur. Ödev de bir sorunun dahi kopya olması durumunda ödevin tamamı (kopya çeken ve kopya veren ayırımı yapılmaksızın) kopya sayılacak olup, ayrıca not olarak -10 puan verilecektir. Yani dönem sonu toplam notunuzdan 10 puan silinecektir. Disiplin soruşturmasında ceza almanız durumunda dersten doğrudan kalacaksınız.
3. Eposta ile ödev sadece kabul edilmeyecektir.
4. Geç ödev teslimi yapılmayacaktır. Ödevden haberi olmamak mazeret sayılmaz.
5. Bu ödevi PostgreSQL veritabanında yapacaksınız. Başka veritabanında yapılan ödevler kabul edilmez.
6. Veritabanını oluşturmak için aşağıdaki CREATE TABLE komutlarını kullanınız. Tabloların içine kendi INSERT komutlarınızla her sorguda en az bir kayıt döndürecek şekilde rastgele kayıtlar ekleyiniz.
7. Sorguları gereksiz yere karmaşık yazmayınız. İhtiyaç yoksa yada soruda talep edilmiyorsa cevapta alt sorgu kullanmayınız. İhtiyaç duyulmayan tabloları sorgularda gereksiz yere kullanmayınız. Yani: o tablo olmadan sorgu doğru çalışıyorsa, sorguda o tabloya ihtiyaç yok demektir. Bazı sorgularda özellikle kullanılması istenen komut ve operatörler varsa onları kullanarak soruları cevaplayınız. Aykırı durumlarda not kırılacaktır.
8. Ara/geçici tablolara, viewlere (sanal tablolara) ve sütun/alan adlarına anlamlı isimler veriniz.
9. **Ödevler PostgreSQL'e içine kopya edilip PDF'e çevrilebilir. PDF olarak teslim edilecektir.**
10. **SQL için her sorguyu ve sorgunun sonucunu (döndürülen kayıtları) kopyala-yapıştır ile veya başka bir şekilde bir metin veya MS Word dosyasına aktarınız. Bu dosyada**
 - a. **Her soru için soru numarası, soru metni, sorgu ve VTYSnin döndürdüğü kayıtlar yani sorgu sonucu bulunmalıdır. Sorgunuzu ve alt sorgularınızı yazarken kolay okunabilmesi için indentation / hizalama kullanınız. SELECT/FROM/WHERE/GROUP BY/HAVING/ORDER BY/... cümleciklerini farklı satırlarda ve alt alta gelecek şekilde düzgün olarak yazınız. Sorgularda anahtar kelimeler büyük harfle, tablo ve alan adları küçük harfle yazılı olmalıdır.**
 - b. Dosya: VT-NNNNNNNN-HWX. şeklinde isimlendirilmelidir. NNNNNNNN: öğrenci-no, X: ödev-no, ext: dosya formatı (doc, docx, txt veya pdf) Dosyanın en başında dersin kodu, adı, öğrenci numaranız, adınız ve ödev numarası (1. ödev), hangi veritabanı sistemi ile yapıldığı yazılı olmalıdır. Eğer grup olarak yaptıysanız gruptakilerin numara ve adları olmalıdır. Ayrıca 'gündüz öğretimi', 'akşam öğretimi' diye belirtiniz.

hospital (hid, name, city) // hatane (hid, adi, sehir)
doctor (did, name, specialty, hid) // doktor (did, adi, uzmanlikAlani, hid)
patient (pid, name, status:{'Pos','Neg'}) // hasta(pid, adi, sonTestSonucu:{'Pos','Neg'})
admitted (pid, hid, admissionDate, entubationDate, deathDate, did) // hastaKabul (pid,hid,kabulTarih, entubeTarih, olumTarih, did)
testType (tid, name, price, accuracy, origin:{'ch','gr',..., 'tr'}) // testTürü (tid, adi, fiyat, güvenilirlik, orijin:{'ch','gr',..., 'tr'})
test (pid, tid, date, result:{'Pos','Neg'}) // test(pid, tid, tarih, sonuc:{'Pos','Neg'})

Yukarıdaki veritabanı şemasını kullanarak aşağıdaki soruları SQL ile yazınız. SQL sorgu ve cevaplarını PostgreSQLde çalıştırarak yukarıdaki KURALLAR'da açıklandığı şekilde veriniz.

1. (10puan-IN/EXISTS/=SOME/ALL) 'Ali KURT' adlı doktorun çalıştığı hastanenin kaydını (hospital tablosundaki tüm sütunları) döndüren sorguyu IN, EXISTS ve SOME/ALL ile 3 farklı şekilde veriniz.
2. (10puan-EXISTS) 'Ali KURT' adlı doktorun çalışmadığı hastanelerin kayıtlarını (hospital tablosundaki tüm sütunları) döndüren sorguyu NOT IN, NOT EXISTS ve SOME/ALL ile 3 farklı şekilde veriniz.
3. (10puan-UNION/EXCEPT/INTERSECT)
 - a. Hem 'Biontech' (testType.name alanını kullanınız) hem de 'Moderna' (testType.name alanını kullanınız) testinden Negatif sonuç alan hastaların kayıtlarını INTERSECT ile listeleyiniz.
 - b. Bir Türk (testType.origin='tr') veya Alman (testType.origin='gr') açısından Pozitif sonuç alan hastaların kayıtlarını UNION ile listeleyiniz.
 - c. Hiç test yaptırmamış olan hastaların kayıtlarını EXCEPT/MINUS ile listeleyiniz.
4. (10puan-GROUP BY-HAVING) Testleri orijin (testType.orgin) ülkelerine göre gruplayarak; sadece doğruluk oranı (testType.accuracy > 80) yüzde 80'in üzerinde olan testler verileri/kayıtları kullanılarak ve 'tr' haricindeki ülkeler için olmak üzere; her ülke için kaç test yapıldığı (test tablosundan bulunur), testler için toplam kaç para ödendiği (testType.price), yapılan testlerin yüzde kaçının pozitif, yüzde kaçının negatif geldiğini listeleyiniz.
5. (10puan-UNIQUE) Aynı isimli testten (testType.name) iki yada daha fazla yaptırmamış (yani isim bakımından farklı testleri yaptırmış) olan hastaların kayıtlarını UNIQUE ile listeleyiniz.
6. (10puan WITH/FROM altsorgu) Hasta başı ortalama test sayısından daha fazla test yaptıran hastaların hid'lerini listeleyiniz. Bunu yaparken ilk iki adımı yani "hasta başı test sayısını" bir WITH alt sorgusuyla, "bunların ortalamasını" başka bir WITH sorgusuyla yazınız.
7. (10puan SELECT alt sorgusu) Hastane kayıtlarını listeyiniz. Fakat kayıtları listelerken her hastanedeki doktor ve yatan hasta sayılarını SELECT'in içerisine birer adet alt sorgu ekleyerek hesaplattırınız. (SELECT *, (alt-sorgu1) doktor-sayısı, (alt-sorgu2) hasta-sayısı FROM hastane)
8. (10puan-HAVING altsorgusu) Uzmanlık alanı 'dahiliye' (specialty='dahiliye') olan en az bir (yani herhangi bir) doktora sahip hastaneler için hid ve o hastanede çalışan tüm doktorların sayısını listeleyiniz (HAVING içerisinde söz konusu hastanenin dahiliye uzmanı çalıştırıp7çalıştırmadığı kontrolünün yapılması gerekmektedir.)
9. (10puan-VIEWS) statistics (date, noOfTests, noOfPositives, noOfDeaths, noOfEntubed) [istatistik (tarih, gunlukTestSayisi, gunlukPozitifSayisi, gunlukOlumSayisi, gunlukEntubeSayisi)] şeklinde bir view oluşturunuz. Bu view'i oluşturmak için önce test tablosundaki test kayıtlarını date(tarih) alanına göre GROUP BY yaparak günlük pozitif ve negatif sayıları bulunabilir. Sonra admitted tablosundaki kayıtlar admissionDate alanına göre GROUP BY yapılarak gunlukOlumSayisi, gunlukEntubeSayisi bulunabilir. Son adımda önceki iki adımda elde edilen sonuçlar gruptamanın yapıldı tarih alanı üzerinden join edilir. Bazı günlerde test, bazı günlerde de olum/entube olamayacağı düşünülerek bu join işleminin FULL OUTER JOIN olması gerekmektedir.
10. 'Ali KURT' adlı hastanın son testinden sonra test yaptıran hastaların kayıtlarını listeleyiniz. (Bir alt sorguda Ali KURT'un test tarihlerini döndürünüz. Hastaların test tarihini bu alt sorgudaki tarihlerle >SOME/ALL operatörünü kullanarak karşılaştırınız)