

ÖDEV 2: SQL, Teslim tarih ve şekli: 02.12.2021 24:00, AKSİS

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, not olarak -10 puan verilecektir. Yani dönem sonu toplam notunuzdan 10 puan silinecektir. Disipline verilmeniz durumunda desten doğrudan kalacaksınız.
3. Ödevler AKSİS üzerinden teslim edilecektir.
4. Geç ödev teslimi yapılmayacaktır. Ödevden haberi olmamak mazeret sayılmayacaktır.
5. Ödev PostgreSQL veritabanında yapılacaktır.
6. Cevapları gereksiz yere karmaşık yazmayınız. İhtiyaç duyulmayan tabloları sorguda gereksiz yere kullanmayınız. Bazı soruları sorguda kullanılması istenen operatörlerle (örnek: EXISTS, IN, =SOME vb.) ifade ediniz. Aykırı durumlarda not kırılacaktır.
7. Geçici ve sanal tablolara ve alan adlarına anlamlı isimler veriniz.
8. Her sorguyu ve sorgunun sonucunu (döndürülen kayıtları) kopyala-yapıştır ile veya başka bir şekilde bir metin veya MS Word dosyasına aktarınız. Bu dosyayı PDF olarak saklayabilirsiniz.
 - a. Ödev çözümünün başında dersin kodu, adı, öğrenci numaranız, adınız ve ödev numarası (1. ödev), hangi veritabanı sistemi ile yapıldığı yazılı olmalıdır.
 - b. Her sorgu için soru numarası, sorgunun kendisi ve sorgu sonucu ödevde konulmalıdır. Sorgunuzu ve alt sorgularınızı her bir SELECT/FROM/WHERE/GROUP BY/HAVING/ORDER BY/... cümlecığı farklı satırlarda alt alta gelecek şekilde düzgün olarak yazınız veya biçimlendiriniz (indentation). Anahtar kelimeler büyük harfle, tablo ve alan adları küçük harfle yazılı olmalıdır.

VERİTABANI ŞEMASI

Student (sid, name, did, noOfCourses, GPA)// öğrenci(ogrenci-no, adi, bolum-no, dersSayisi)
Take (sid, cid, grade) // ders-al(ogrenci-no, ders-kodu, notu)
Course (cid, title, credits, did, noOfStudents)// ders(ders-kodu, adi, kredisi, bolum-no, ogrSayisi)
Department (did, name, noOfStudents) // bolum(bolum-no, adi, ogrSayisi)
Teacher (tid, name, placeOfBirth, did) // hoca(hoca-no, adi, dogum-yeri, bolum-no)
Teach (tid, cid) // ders-ver(hoca-no, ders-kodu)

SORGULAR

(Önce yukarıdaki tabloları CREATE TABLE ile oluşturunuz. Sonra tablolara rastgele kayıtlar INSERT ediniz. Sonra sorgularınızı yazıp test ediniz.

1. Tüm öğrencilerin ağırlıklı not ortalamalarını (GPA) bir alt sorguyla hesaplayıp güncelleyiniz. Açıklama: Şu komuttaki alt sorguyu yazmanız gerekecektir. UPDATE student SET GPA = (.....)

GPA = SUM(grade x credits)/SUM(credits) formülüyle hesaplanır.

2. Bölümlerin hepsinde ders veren hocaların kayıtlarını listeyeyiniz. Açıklama: Bu bir ilişkisel cebir bölme işlemidir. Bu soru şu sorgu kalıbıyla cevaplanabilir:

SELECT * FROM teacher t WHERE ((tüm dersler) – (t hocasının verdiği dersler) = ϕ)

3. Hocaların verdiği ders sayılarının ortalamasının altında ders veren hocaların "tid, name, verdiği ders sayısı ve derslerini alan öğrencilerin sayılarını" listeleyiniz. Açıklama: Bu soru 3 adımda çözülebilir.

- (i) Her hoca'nın kaç ders verdiđi bulunur: dersSayi(tid, dersSayisi)
 - (ii) dersSayi sorgusundaki dersSayisi' deđerlerinin AVG ile ortalaması bulunur: ortalamaDers(ort)
 - (iii) dersSayi, ortalamaDers ve teacher tablolarından dersSayisi > ortalamaDers.ort olan kayıtlar için soruda istenenler hesaplanır.
4. Kredisi > 3 olan ve en fazla 20 öđrencinin aldıđı derslerin "tid, öğrenci sayısı, not ortalaması"larını dersi alan öğrenci sayısına göre artan, dersteki notların ortalamasına göre azalan sırada listeleyniz. Açıklama: Bu sorguda SELECT FROM WHERE GROUP BY HAVING ORDER BY cümleciklerinin hepsini kullanmanız gerekmektedir.
 5. Aldıđı tüm derslerdeki notu o dersteki (kendi notu hariç olmak üzere) notların ortalamasından yüksek olan öğrencilerin sid'lerini listeleyniz.
 6. Bölümlerin "did, öğrenci sayılarını ve bölümdeki öğrencilerin notlarının ortalamalarını" listeleyniz. Bu soruda öğrenci sayısı için SELECT cümlecisinde alt-sorgu kullanılacaktır. Açıklama: SELECT did, (alt sorgu) "ogrenci sayısı", (alt sorgu) ogrenciNotOrtalama FROM department d. gibi yazılır.
 7. En az iki farklı hocadan yada iki farklı bölümden ders alan öğrencilerin kayıtlarını listeleyniz. Açıklama: "yada" dendiđi için UNION kullanıp 2 alt sorgu yazılabilir. Alt sorgularda COUNT(DISTINCT teach.tid) ve COUNT(DISTINCT course.did) cümlecikleriyle sayma işlemi yapacaktır. İki farklı hoca için teach ve take tablolarına erişim, iki farklı ders için take ve course tablolarına erişim gereklidir.
 8. "Ali KURT" ve "Ayse KURT" adlı öğrencilerin aldıđı derslerin kredilerinin toplamında daha fazla kredi alan öğrencilerin kayıtlarını listeleyniz. Açıklama: Birinci adımda öğrenciye göre gruplama yapıp kredi toplamı diđer 2 öğrencinin kredi toplamından büyük olan grupların sid'leri listelenir. İkinci adımda student tablosundan bu öğrencilerin kayıtları listelenir.
 9. Aynı notu 2 farklı dersten almamış yani tüm notları birbirinden farklı olan öğrencilerin kayıtlarını WHERE içinde UNIQUE fonksiyonu kullanarak veriniz.
 10. Girilmemiş yada verilmemiş notu olmayan (IS NOT NULL öğrencilerin listeleyniz