

VERİTABANI ŞEMASI

sarki(sarkino, ad, tur, sure, bestecino, yazarno)
sarkici(sarkicino, ad, tur, dogumTarih, dogumYer, albumSayisi)
album(albumno, ad, yil, fiyat, sarkicino, stokAdet)
albumdekiSarki(albumno, sarkino, sıra)
besteci(bestecino, ad, tur, sarkiSayisi)
sozyazari(yazarno, ad, dogumYer)

SORULAR (Her soru aynı puandır)

- Aşağıdaki problem için bir E-R diagramı çizin ve bu diagramı ilişkisel veritabanı şemasına çevirin:
 - Sınavların adı, türü('arasınav', 'final' yada 'quiz'), sınavdaki soru sayısı, sınavın toplam puanı (sınavdaki soruların puan değerleri toplamından hesaplanır), ve geçme notuna etki yüzdesi veritabanında saklanmalıdır.
 - Sınavlardaki test sorularının numarası, sorunun metni, 5 tane cevap şıkkının metinleri, doğru cevap şıkkı ('A','B','C','D' yada 'E' olabilir), sorunun puan olarak değeri veritabanında saklanmalıdır.
 - Bir sınavda birçok soru sorulabilir. Bir soru birçok sınavda sorulabilir. Her sınavda en az bir soru sorulmalı ve her soru en az bir sınavda sorulmalıdır.
- Sarkici tablosundaki albumSayisi alanını gerektiğinde güncelleyen trigger(lar)ı yazınız. [Not: Şarkıcıların albümlerinin sayısı album tablosundaki sarkicino alanı kullanılarak bulunabilir.]
- Verilen bir doğum yeri için (sozyazari tablosundaki dogumYer alanını kullanınız) o doğum yerinde doğan yazarların yazarno, sözünü yazdığı şarkıların sayısı ve bu şarkıların toplam süresini bir tablo olarak döndüren stored fonksiyonu yazınız. [Not: Bu fonksiyon ...sarkilar('Ankara')... şeklinde çağırılacaktır]
- Hiç şarkı bestelememiş olan bestecilerin kayıtlarını html tablosu olarak oluşturan(yazdıran) bir PHP kodunu veriniz. [Not: Fonksiyonda mysqli_connect, mysqli_query, mysqli_fetch_assoc / mysqli_fetch_array, mysqli_close, mysqli_error kullan(abilirs)iniz.]
- Student adlı MongoDB collection'da aşağıdaki dokümanlar olsun. Student collection üstünde aşağıdaki işlemleri yapan MongoDD komutlarını veriniz.
 - No'su 5 olan öğrencinin ismini 'Ömer KURT' olarak değiştirin.
 - 101 numaralı dersi alan (yani "courses" alanındaki arrayda "cid":101 elemanı olan) öğrencileri student collection'dan siliniz. ["courses" alanındaki arrayda "cid":101 elemanı olup olmadığını kontrol etmek için hayali yani kendi kafanızdan uyduracağınız bir fonksiyon kullanabilirsiniz. Ama burada sözdizimin JSON ve mongodb formatına uygun olmasına dikkat ediniz.
 - Bölümlerin adını ve o bölümdeki öğrenci sayısı listeleyen sorguyu yazınız [list the department and the number of students in the department for the departments in student collection]

```
db.student.find()
```

```
{ "_id": ObjectId(...), "no": 5, "name": "Ali KURT", "courses": [{"cid": 101, "grade": 80}, {"cid": 202, "grade": 90}],  
"dept": "Comp Eng"}  
{ "_id": ObjectId(...), "no": 6, "name": "Ayşe KURT", "courses": [{"cid": 202, "grade": 100}, {"cid": 302, "grade":  
50}], "dept": "Elec Eng"}  
{ "_id": ObjectId(...), "no": 7, "name": "Esme KURT", "courses": [{"cid": 301, "grade": 60}, {"cid": 101,  
"grade": 100}], "dept": "Comp Eng"}
```