

1) Aşağıdaki tanımları yapınız.

a) Veritabanı tasarımında kullanılan anahtar türlerinin üçünü açıklayınız.

Süper Anahtar, ilişkideki tüm nitelik değerlerini belirleyen ve iki farklı satırda aynı değeri alamayan nitelik alt kümesidir.

Aday anahtar: Kaydı belirleyen minimum alan kümeleridir.

Birincil Anahtar: Anahtarlar arasında kaydı tanımlamak için en uygun aday anahtar kümesidir.

b) Fonksiyonel bağımlılık adaya anahtar ilişkisi nasıl kurulmaktadır. Belirtiniz.

Bir nitelik değeri diğer bir niteliğin değerini kesin belirliyorsa aralarında fonksiyonel bağımlılık vardır denir. Fonksiyonel bağımlılık gerçek hayatta nitelikler arası ilişkilerden çıkartılır.

c) Satır tabanlı ilişkisel hesaba kısaca açıklayınız? SQL'e en temel katkısı nedir? Açıklayınız.

Satır tabanlı ilişkisel hesap, ilişkiler için satır değişkenleri tanımlamayı sağlar. Bu değişkenler üzerinde belli koşullar yazılarak istenen sorgu sonuçları bulunur. İlişkisel cebir gibi matematiksel ifadeler üzerine kuruludur ve SQL diline katkı sağlamıştır. Ancak satır tabanlı ilişkisel hesap, ilişkisel cebir gibi prosedürel değildir. Basit bir satır hesap ifadesi satır değişken tanımları ve şartlardan oluşur.

2) Aşağıdaki soruları verilen veri modeli göre cevaplandırınız.

a) `SELECT mt.adi, mt.soyadi FROM musteriler mt, musteriler m
WHERE mt.personel_id=m.temsilci_id
AND m.toplam_gelir >= 100;`

b) `SELECT mt.adi, mt.soyadi FROM musteriler mt, musteriler m
WHERE mt.personel_id=m.temsilci_id
AND mt.adres_il="İstanbul"
AND NOT EXISTS (SELECT * FROM siparis s
WHERE s.musteri_id=m.musteri_id
AND s.durum="Açık");`

c) `SELECT m.adi, m.soyadi FROM musteriler m, siparis s
WHERE s.musteri_id=m.musteri_id
GROUP BY musteriler_id
HAVING SUM(s.adet*s.satis_fiyat) != m.toplam_gelir;`

d) `SELECT mt.adi, mt.soyadi FROM musteriler mt
WHERE (SELECT toplam_gelir FROM musteriler m
WHERE mt.ilmusteri=m.musteri_id) >
ALL (SELECT toplam_gelir FROM musteriler m
WHERE mt.personel_id=m.temsilci_id);`

e) `UPDATE musteriler mt
SET ilmusteri = (SELECT musteriler_id FROM musteriler m1
WHERE m1.temsilci_id=mt.personel_id
AND toplam_gelir > ALL (SELECT musteriler_id FROM musteriler m2
WHERE m2.temsilci_id=mt.personel_id
AND m2.musteri_id!=m1.musteri_id
AND toplam_gelir));`

3) 2a ve 2c. Şıklarını ilişkisel cebirle çözünüz.

a) $\pi_{mt.adi,mt.soyadi} ((\rho_{mt} (musteri_temsilcisi)) \bowtie_{mt.personel_id=m.temsilci_id} (\rho_m (\sigma_{musteri.toplam_gelir \geq 100} (musteri))))$

b) $\sigma_{-(m.toplam_gelir = SUM(s.adet * s.satis_fiyat))} ((\rho_m (musteri)) \bowtie_{m.musteri_id=s.musteri_id} (\rho_s (siparis))))$

4) $\{mt | muster_temsilcisi(mt) \text{ AND } (\exists m)(musteri(m) \text{ AND } m.temsilci_id=mt.temsilci_id \text{ AND } (\forall s)(siparis(s) \text{ AND } s.musteri_id=m.musteri_id \text{ AND } s.urun_id \neq 'Silgi')))\}$

5) $ABC \rightarrow D, AC \rightarrow E, A \rightarrow B, A \rightarrow D$

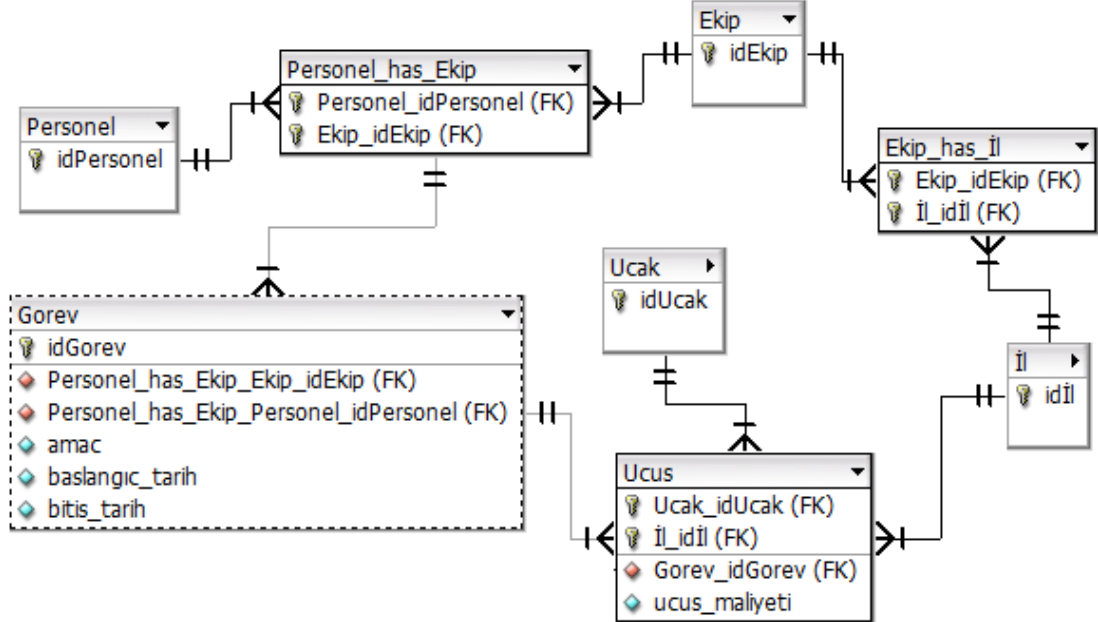
ABC aday anahtardır. Tam fonksiyonel bağımlı olmadığı için 2NF'den itibaren normalizasyon başlamalıdır.

$R1=\{ABCE\}$

$R2=\{AD\}$

Geçişli bağıllık olmadığı için 3NF'a uygun haldedir. Fakat aday anahtar içinde olan B'nin her zaman determinant olması gerekir. Bu yüzden BCNF'ye göre de düzenlemek gerekir.

6) Bir havalimanı tamir ekibinin görev takip sistemi tasarlanacaktır. Bu sistemde personeller çeşitli ekiplerde görev alarak tamir işlemlerini yürütmektedir. Görevlerin amacı ve başlangıç bitiş tarihi bulunmaktadır. Her ekibin sorumlu olduğu iller bulunmaktadır. Bir personel birden fazla ekipte görev alabilir. Ancak içinde bulunduğu ekibin tamir sahası dışında çalışamaz. Ekipler görevlere çeşitli uçaklarla girmektedir. Bir ile hangi uçakla uçulduğuna göre uçuş maliyeti değişmektedir.



```
7) DECLARE
    CURSOR c1 IS
        SELECT musteri_id,adi,soyadi FROM musteri;
    r1 c1%ROWTYPE;
BEGIN
    OPEN c1;
    FOR r1 IN c1 LOOP
        DBMS_OUTPUT.PUT_LINE(r1.adi || r1.soyadi || get_Toplam(r1.musteri_id));
    END LOOP;
    CLOSE c1;
END;
```