

2016 – 2017 GÜZ DÖNEMİ
VERİTABANI SİSTEMLERİNİN GERÇEKLENMESİ
2. VİZE LABORATUVAR SORULARI

Öğrenci Numarası : **CEVAP ANAHTARI**
Ad-Soyad :
Gr : (IOS) Grup 1,2 ☐ (GA) Grup 3,4 ☐ (PC) Grup 5,6 ☐ (FÇ) Grup 7,8 ☐

Açıklamalar

1. Bu bölüm laboratuvarda işlenen konulardan hazırlanmıştır.
2. Bu bölümü, sadece, laboratuvar dersinden muaf olmayan öğrenciler çözeceklerdir.
3. Bu bölüm 100 puan üzerinden değerlendirilecektir.
4. Değerlendirme sonucu laboratuvar puanına eklenecektir.
5. Grup bilginizi işaretlemeyi unutmayınız.
6. Süre 30 dk.

CompanyDB

Aşağıdaki bilgiler Company veritabanındaki tabloları ve tablolardaki sütunları göstermektedir. Tüm sorular bu tablolara göre yanıtlanacaktır. Altı çizili ifadeler birincil anahtarları (primary key) göstermektedir.

Department(dname, <u>dnumber</u> , mgrssn, mgrstartdate)	Şirket bünyesindeki departmanları göstermektedir. Her departmanın ismi, numarası, yönetici kimlik numarası ve yöneticinin yönetici olarak işe başladığı tarih tutulmaktadır.
Project(pname, <u>pnumber</u> , plocation, dnum)	Şirketin yürüttüğü projeleri göstermektedir. Her projenin ismi, numarası, yapıldığı yer ve hangi departmanın projesi olduğu bilgisi tutulmaktadır.
Dept_location(dnumber, dlocation)	Hangi departmanın nerelerde ofisi olduğunu göstermektedir. Bu bilgiler departmanın numarası ve ofisin yeri ile belirtilmektedir.
Employee(fname, minit, lname, <u>ssn</u> , bdate, address, sex, salary, superssn, dno)	Şirket bünyesindeki çalışanların bilgilerini göstermektedir. Her çalışan için; ad, soyad, kimlik numarası, doğum tarihi, açık adres, cinsiyet, maaş, amirinin kimlik numarası ve bağlı olduğu departman numarası tutulmaktadır.
Works_on(essn, pno, hours)	Şirket bünyesindeki projelerde çalışanların, çalışma bilgisi tutulmaktadır. Çalışanın kimlik numarası, projenin numarası ve kişinin bu projede kaç saat çalıştığı bilgisi tutulmaktadır.
Dependent(essn, dependent_name, sex, bdate, relationship)	Şirket bünyesinde çalışanların akraba bilgileri tutulmaktadır. Her akraba için, çalışanın kimlik numarası (kimin akrabası ise), akrabanın ismi, cinsiyeti, doğum tarihi ve çalışan ile olan akrabalık ilişkisi bilgileri yer almaktadır.

1. (10 puan) Aşağıdaki sorgulardan hangisi tüm kadın akrabaların ilişki durumunu 'Görümce' olarak güncelleyen sorgudur?

- a) UPDATE dependent SET relationship = 'Görümce' WHERE sex = 'F'
b) UPDATE FROM dependent SET relationship = 'Görümce' WHERE sex = 'F'
c) UPDATE relationship = 'Görümce' FROM dependent WHERE sex = 'F'
d) UPDATE relationship = 'Görümce' SET dependent WHERE sex = 'F'
e) UPDATE FROM dependent SET relationship = 'Görümce' WHERE sex = 'F'

2. (10 puan) Aşağıdaki sorgulardan hangisi supervisor olmayan çalışanları silen sorgudur?

- a) DELETE FROM employee WHERE ssn <> superssn
b) DELETE FROM employee e, employee s WHERE s. ssn <> e.superssn
c) DELETE FROM employee WHERE NOT EXISTS (SELECT null FROM employee WHERE superssn <> ssn)
d) DELETE FROM employee WHERE ssn NOT IN (SELECT s.ssn FROM employee e, employee s WHERE e.superssn = s.ssn);
e) DELETE FROM employee WHERE ssn NOT EXISTS (SELECT s.ssn FROM employee e, employee s WHERE e.superssn = s.ssn)

3. (10 puan) Aşağıda verilen sorgunun sorusu hangisidir?

SELECT dname, COUNT(*) FROM project, department WHERE dnum = dnumber GROUP BY dname HAVING COUNT(*) <= 2;

- a) Her bir projenin kaç departmanda yapıldığını bulup departman isimlerine göre listeleyiniz. Departman sayısı 2'den fazla olan projeleri dâhil etmeyiniz.
b) Her bir departmanda kaç proje yapıldığını departman isimlerine göre listeleyiniz. Proje sayısı 2'den fazla olan projeleri dâhil etmeyiniz.
c) Her bir departmanda kaç proje yapıldığını proje isimlerine göre listeleyiniz. Departman sayısı 2'den fazla olan projeleri dâhil etmeyiniz.
d) Her bir projenin kaç tane departmanda yapıldığını proje isimlerine göre listeleyiniz. Departman sayısı 2'den fazla olan projeleri dâhil etmeyiniz.
e) Her bir projenin kaç tane departmanda yapıldığını proje isimlerine göre listeleyiniz. Proje sayısı 2'den fazla olan projeleri dâhil etmeyiniz.

4.) (10 puan) Aşağıdaki sorgulardan hangisi erkek akrabası olan çalışanların sayısını ve ortalama maaşlarını listeleyen sorgudur?

- a) SELECT COUNT(ssn), AVG(salary) FROM employee WHERE ssn NOT IN (SELECT ssn FROM dependent d WHERE ssn = essn AND d.sex = 'F')
- b) SELECT COUNT(ssn), AVG(salary) FROM employee WHERE sex = 'M' AND ssn IN (SELECT ssn FROM dependent d WHERE ssn = essn)
- c) SELECT COUNT(ssn), AVG(salary) FROM employee, dependent d WHERE ssn = essn AND d.sex = 'M'
- d) SELECT COUNT(ssn), AVG(salary) FROM employee WHERE EXISTS (SELECT null FROM dependent d WHERE ssn = essn AND d.sex = 'M')**
- e) SELECT COUNT(ssn), AVG(salary) FROM employee e, dependent d WHERE ssn = essn AND d.sex = 'M' AND e.sex = 'M'

5. (10 puan) Şirketin sınırları içerisine bir lojman yapılmıştır. Lojman yerleşimi için çocuğu olan çalışanlara öncelik verilecektir. Buna göre çocuğu olan çalışanların isim ve soyisimlerini listeleyen view'i yazınız.

```
CREATE VIEW liste AS
SELECT fname, lname
FROM employee e, dependent d
WHERE e.ssn = d.essn AND (relationship = 'Daughter' OR relationship = 'Son');
```

5. (20 puan) Şirkette en çok akrabası olan çalışanın/(çalışanlardan birisinin) ssn numarasını bulunuz.

```
SELECT essn FROM dependent GROUP BY essn ORDER BY COUNT(*) DESC LIMIT 1;
```

7. (30 puan) Numarası verilen bir projede en yüksek maaşı alan kişinin/(kişilerden birisinin) SSN'ini MAX ve LIMIT fonksiyonlarını kullanmadan bulan ve OUT değişkeni üzerinden döndüren fonksiyonu yazınız (Verilen fonksiyon tanımını içerisinde ve ilk parametresini doldurunuz).

```
CREATE FUNCTION soru7(proje_no project.pnumber%TYPE, OUT max_maas_ssn employee.ssn%TYPE, OUT maks_maas integer) AS $$
```

```
DECLARE
```

```
    maaslar CURSOR FOR SELECT ssn, salary FROM employee, works_on WHERE essn = ssn AND pno = proje_no;
```

```
BEGIN
```

```
    SELECT AVG(salary) INTO maks_maas FROM employee, works_on WHERE essn = ssn AND pno = proje_no;
    FOR maas IN maaslar LOOP
```

```
        IF maas.salary >= max_maas THEN
```

```
            maks_maas = maas.salary;
```

```
            max_maas_ssn = maas.ssn;
```

```
        END IF;
```

```
    END LOOP;
```

```
END; $$
LANGUAGE plpgsql;
```