

## VERİ TABANI YÖNETİMİ DERSİ ÖRNEK SORULAR – 2

### 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 vardır. 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.

a. İsmi 'Pro' ile başlamayan projelerin isimlerini listeleyen sorguyu yazınız.

```
SELECT pname
FROM project
WHERE pname NOT LIKE 'Pro%'
```

Açıklama: Sadece proje isimlerinin listelenmesi gerektiğinden "project" tablosu yeterli olacaktır. Şart olarak ise nasıl başlamaması gerektiğini belirttiğinden "NOT LIKE" kullanılmıştır.

b. İsmi 'Pro' ile başlamayan projelerde kaç kişinin çalıştığını ve proje ismini gösteren sorguyu yazınız.

```
SELECT pname, count(*)
FROM project, works_on
WHERE pname NOT LIKE 'Pro%' AND pnumber = pno
GROUP BY pname
```

Açıklama: Önceki soruya ek olarak projelerde çalışanların sayısı da istendiğinden "works\_on" tablosu da sorguya eklenmelidir. İki tablo olduğundan birleştirme şartı soruya eklenir. "GROUP BY" komutu ile projeler proje isimlerine göre gruplandırılır. "SELECT" kısmına yazılan "count(\*)" fonksiyonu ile 'Pro' ile başlamayan her projedeki kişilerin sayısı hesaplanır.

c. İsmi 'Pro' ile başlamayan projelerde kaç kişinin çalıştığını ve proje ismini; proje ismine göre ters alfabetik olacak şekilde gösteren sorguyu yazınız.

```
SELECT pname, count(*)
FROM project, works_on
WHERE pname NOT LIKE 'Pro%' AND pnumber = pno
GROUP BY pname
ORDER BY pname DESC
```

Açıklama: Önceki soruya, sıralama şartı eklenmiştir. Tersten sıralanacağı için "ORDER BY" komutunda "DESC" kullanılmıştır. Normal sıralanması istenseydi "ASC" yazılması ( bir şey yazılmadığı durumunda kabul edilen "ASC" dir) yeterli olacaktı.

- d. İsmi 'Pro' ile başlamayan projelerde kaç kişinin çalıştığını ve proje ismini; proje ismine göre ters alfabetik olacak şekilde gösteren sorguyu yazınız. Çalışan sayısının 3'ten az olduğu projeleri dahil etmeyiniz.

```
SELECT pname, count(*)
FROM project, works_on
WHERE pname NOT LIKE 'Pro%' AND pnumber = pno
GROUP BY pname
HAVING count(*)>=3
ORDER BY pname DESC
```

Açıklama: Önceki soruya ek olarak çıktıda gruplanarak gösterilen projelerden, çalışan sayısı 3 ve fazlası olanların gösterilmesi istenilmiştir. Dolayısıyla "HAVING" komutunda her bir projedeki çalışanlar "count(\*)" ile bulunarak ilgili şartı sağlayanları listelenmesi sağlanmıştır.

- e. İsmi 'Pro' ile başlamayan projelerde kaç kişinin çalıştığını ve proje ismini; proje ismine göre ters alfabetik olacak şekilde gösteren sorguyu yazınız. Çalışan sayısının 3'ten az olduğu projeleri dahil etmeyiniz. Ayrıca çıktıda gösterilecek tablonun sadece 5. Satırından sonraki 3 satırı gösteriniz.

```
SELECT pname, count(*)
FROM project, works_on
WHERE pname NOT LIKE 'Pro%' AND pnumber = pno
GROUP BY pname
HAVING count(*)>=3
ORDER BY pname DESC
LIMIT 3 OFFSET 5
```

Açıklama: Önceki soruya ek olarak sorgunun çıktısı olan tablodaki belirli bir yerin yazdırılması istenilmiştir. 5. Satırdan sonra denildiğinde "OFFSET 5" ifadesi kullanılmıştır. 3 satır yazılması istenildiğinden komutumuz "LIMIT 3" şeklinde başlamıştır. "LIMIT 5, 3" ifadesi de farklı sorgu dillerinde geçerli cevap olacaktır.

2. İki veya daha fazla Ođlu (Son) olan alıřanın/alıřanların ssn numaralarını listeleyen view'i yazınız. View'i "depend\_calisan" řeklinde isimlendiriniz.

```
CREATE VIEW depend_calisan
AS
SELECT ssn
FROM employee, dependent
WHERE ssn = essn AND relationship = 'Son'
GROUP BY ssn
HAVING count(*) >= 2
```

Aıklama: alıřanların "ssn" numaraları ve akrabaları ile ilgili bilgiler kullanılacađından "employee, dependent" tabloları sorguya eklenmiřtir. "WHERE" řartı ile tablolar birleřtirilmiř ve sadece ođlu olan alıřanların kalması sađlanmıřtır. Kalan satırlar alıřanın "ssn" numarasın gre gruplandırılıp "HAVING" řartı ile 2 veya daha fazla ođlu olanların bulunması sađlanmıřtır.

3. İsmi 'Pro' ile bařlayan projelerde alıřanların ssn numaralarını IN kullanarak bulan sorguyu yazınız.

```
SELECT essn
FROM works_on
WHERE pno IN ( SELECT pnumber
                FROM project
                WHERE pname LIKE 'Pro%' )
```

Aıklama: Projede alıřanların "ssn" numaraları "works\_on" tablosunda mevcuttur. Haliyle bu sadece bu tablonun kullanılması yeterlidir. Ancak "ssn" numarasını yazdırılması iin "works\_on" tablosundaki "pno" ya karřılık gelen proje isminin 'Pro' ile bařlaması gerekmektedir. Bunun iin "IN" kullanılarak "project" tablosunda ismi 'Pro' ile bařlayanların numaraları bulunur.

4. Herhangi bir departmanın yneticisi olmayan alıřanların isim ve soyisimlerini EXISTS / NOT EXISTS (uygun olanı ile) bulan sorguyu yazınız.

```
SELECT fname, lname
FROM employee e
WHERE NOT EXISTS ( SELECT null
                   FROM department
                   WHERE mgrssn = e.ssn )
```

Aıklama: Kiřilerin "department" tablosundaki "mgrssn" stununda "ssn" numarası yazıyorsa, bir departmanın yneticisidir. Bu durumda "employee" tablosundaki bir alıřanın ismi yazdırılırken "ssn" numarası "department" tablosunda olmadıđı durumda yazdırılacaktır. Bu yzden dıřtaki sorgudan gelen "e.ssn" iteki sorguda "mgrssn" e eřit olması istenilmiřtir.

5. 'Software' projesinde çalışıp oğlu olmayan çalışanların isim ve soyisimlerini gösteren sorguyu yazınız. (INTERSECT/UNION/EXCEPT ten uygun olanı kullanınız)

```
SELECT fname, lname
FROM project, works_on, employee
WHERE pnumber = pno AND essn = ssn AND pname='Software'
```

EXCEPT

```
SELECT fname, lname
FROM employee, dependent
WHERE essn = ssn AND relationship='Son'
```

Açıklama: Soruyu 'Software' projesinde çalışanlar (A kümesi) ve oğlu olanlar (B kümesi) olarak iki kümeye ayırabiliriz. Soruda istenilen ise A kümesinde olup B kümesinde olmayanlar, yani A-B. Bu durumda "EXCEPT" kullanılmalıdır. A kümesi için yazılan sorgu 'Software' projesinde çalışanları vermektedir. B kümesi için yazılan sorgu ise oğlu olan çalışanları vermektedir.

### Örnek Diğer Sorular

1. Şirketin sınırları içerisinde bir lojman yapılmıştır. Lojman yerleşimi için çocuğu olan çalışanlara öncelik verilecektir. Buna göre çocuğu olan çalışanların isim ve soyisimlerini listeleyen view'i yazınız.
2. Her bir projede toplam kaç saat çalışıldığını bulup listeleyiniz. Toplamda 50 saatten az çalışılan projeleri listeye dahil etmeyin. Proje ismine göre ters alfabetik olarak listeleyiniz.
3. Her bir projede kaç kadın (F) çalışan olduğunu bulunuz. İki'den az sayıda kadın çalışanı olan projeyi listeye dahil etmeyiniz. Proje ismine göre ters alfabetik olarak sıralayınız. Çıktıda sadece ilk 4 satırı gösteriniz.
4. Çalışan sayısının en az 5 olduğu departmanların isimlerini ve toplam maaş bilgisini listeleyen view'i yazınız. View'i "dep\_maas" şeklinde isimlendirin.
5. Hiçbir projede görev almayan işçilerin ad, soyad bilgilerini listeleyin.
6. Ortalama 20 saatten çok çalışılan projelerin isimlerini ve çalışan sayısını bulan sorguyu yazınız.

### Bilgilendirme: LIMIT

Sorgu sonucu oluşacak tabloda belirli bir kısmın seçilmesini sağlar.

"LIMIT A": Bu kullanımda ilk A satır gösterilmektedir.

"LIMIT A OFFSET B": Bu kullanımda A satır gösterilmektedir. Ancak B. satırdan sonraki A satırı ifade etmektedir.

"LIMIT B, A": Bu kullanım "LIMIT A OFFSET B" kullanımı ile aynıdır. Bu kullanım bazı sorgu dillerinde geçerli olmayabilir.

Örnekler:

- "LIMIT 1": İlk satır
- "LIMIT 3": İlk üç satır
- "LIMIT 3 OFFSET 5": Beşinci satırdan sonraki ilk üç satır. Yani 6., 7. Ve 8. Satırlar
- "LIMIT 5, 3": Beşinci satırdan sonraki ilk üç satır. Yani 6., 7. Ve 8. Satırlar