Veritabanı uygulama dersi - 4: Tablolarda Kısıt, View, Sequences işlemleri

Şirkette çalışan işçiler kendi aralarında basketbol takımları kurmuşlardır. Her bir işçi belli hafta boyunca bu takımlarda oynayabilir. Işçilerin tek bir takımda oynama zorunluluğu yoktur. Takım tablosu aşağıdaki gibidir:

TEAM

TNUMBER (PK)	NOT NULL	NUMERIC(2)
TNAME	NOT NULL	VARCHAR(15)

Örnek:
1. Team tablosu ve Employee tablolarının ilişki tablosu olan Team_employee tablosunu aşağıdaki özelliklere sahip olacak şekilde yaratınız:

Nitelikler: Tno (numeric(2)), essn (char(9)), play_time (numeric (2))
Tno: takım no'su; essn: çalışan ssn'si; play_time: bu çalışanın, bu takımda kaç hafta oynadığı.
Kısıtlar:

- Tno ve essn birlikte belirleyici nitelik (primary key) olmalıdır.
- **Tno** Team tablosundaki **tnumber**'i referans almalıdır ve Team tablosundan bir satır silindiğinde bu satıra ait tno'lu satırlar da silinmelidir.
- Essn Employee tablosundaki ssn'yi referans almalıdır ve Employee tablosundan bir satır silindiğinde bu satıra ait ssn'li satırlar da silinmelidir.
- Play_time alanının 12'den çok olması engellenmelidir. (bir çalışan bir takımda 12 haftadan uzun süre oynayamaz!)

Cevap:

Team tablosu:
```
create table team(tnumber numeric(2), tname varchar(15), constraint pk_team primary key(tnumber))
```

Employee - team tabloları arasındaki ilişki tablosu:
```
create table team_employee (Tno numeric(2), essn char(9) , play_time numeric(2), constraint
pk_team_emp primary key(tno, essn) ,
constraint fk_team foreign key(tno) references team(tnumber) on delete cascade,
constraint fk_emp foreign key(essn) references employee (ssn) on delete cascade,
constraint play_time_ck check (play_time <13)  );
```

Açıklama: Cevabın 2. satırında primary key tanımı yapılmaktadır.
Üçüncü satırda tno'nun, team tabosundaki tnumber'ı referans gösterdiği söyleniyor ve "**on delete cascade**" eklenerek team tablosunda bir takım (tnumber) silindiğinde, team_employee tablosunda karşılık gelen tno'ların bulunduğu satırlar da silinsin deniyor.
Dördüncü satırda essn'nin employee tablosundaki ssn'ye referans verdiği söyleniyor. Ve employee'den bir satır silinince, team_employee tablosunda karşılık gelen satırların da silinmesi isteniyor.
Son satırda ise play_time isimli numeric kolonun sadece 13'ten küçük sayılar içermek zorunda olduğu söyleniyor.

View nedir?

View'ler bir sorgu sonucunda oluşan sanal tablo yada sanal veri kümesi gibi düşünülebilir. Tablolar gibi veritabanında fiziksel olarak bulunmazlar, sorgu olarak saklanırlar.

View'leri oluşturan sorgularda adı geçen tabloların verileri değişince view'lerin verisi de kendiliğinden değişir.

CREATE VIEW *viewname*
AS
SELECT *select-list*
FROM table-list
[WHERE search-condition]
Şeklinde yaratılır.

Örnekler:

1. Maaşı 20000 ile 40000 arasında olan çalışanların isimlerini ve maaşlarını gösteren bir view yazın.
2. "Sales" isimli departmanda çalışanların ad, soyad ve cinsiyetlerini gösteren bir view yazın.

Cevaplar:

1. İstenen view oluşturulurken employee tablosu kullanılacaktır. View'in ismi maas_araligi olsun:

create view maas_araligi as select fname, lname, salary from employee where salary between 20000 and 40000;

NOT: Örneğin maaşı 30000 olan çalışanları employee tablosundan silersek, ilgili çalışanlar bu view'den de silinir.

2. View'in ismi sales_calisanlari olsun:

create view sales_calisanlari as select fname, lname, sex from employee e, department d where e.dno=d.dnumber and d.dname='Sales'

Sequence nedir?

Belli bir sırada nümerik değer üretmemizi sağlayan veritabanı nesnesidir. Genelde primary key'ler gibi unique olması gereken alanlara, otomatik artan değerler atamakta kullanılır.

CREATE SEQUENCE sequence_name
[INCREMENT BY #]
[START WITH #]
[MAXVALUE # | NOMAXVALUE]
[MINVALUE # | NOMINVALUE]
[CYCLE | NO CYCLE] =>Son değere ulaşılınca başa dönülebilir / dönülemez
[CACHE #] =>İşlemleri hızlandırmak için belli sayıda sequence belleğe alınabilir

Şeklinde yaratılır.
"Select * from *sequence_name*" denilerek önceden yaratılmış bir sequence hakkında bilgi alınabilir.

Örnek:
1. 9'dan başlayıp 99'a kadar birer birer artan bir sequence yaratın. Bu sequence'ı Team tablosunun tnumber'larının numaralandırılmasında kullanın (bunun için team tablosuna veri girmeniz gerekecektir).

Cevap: Sequence'in ismi seq olsun
create sequence seq minvalue 9 maxvalue 99 increment by 1;

insert into team values(nextval('seq'), 'Kedicikler')

nextval('seq') ile sequence'in sıradaki değerini alıp tnumber'a atıyoruz.

Tablolarda Union, Intersect, Except İşlemleri

1. "OperatingSystems" isimli projede çalışanların **ve** "Software" departmanında çalışanların ad, soyad bilgilerini bulunuz. (UNION ? / INTERSECT ? / EXCEPT ?)

2. "OperatingSystems" isimli projede çalışanların **veya** "Software" departmanında çalışanların ad, soyad bilgilerini bulunuz. (UNION ? / INTERSECT ? / EXCEPT ?)

3. "OperatingSystems" isimli projede çalışanların ve "Software" departmanında **çalışmayanların** ad, soyad bilgilerini bulunuz. (UNION ? / INTERSECT ? / EXCEPT ?)

4. Hiçbir departmanın veya hiçbir çalışanın yöneticisi olmayan çalışanların isimleri. (EXISTS ? / NOT EXISTS?)

5. İsmi 'John' olan işçilerin çalıştıkları departmanların isimlerini 'IN' kullanarak bulunuz.

6. 'Sales' departmanında kaç kişinin çalıştığını, en düşük, en yüksek, ortalama ve toplam maaşı bulunuz.

Cevaplar
1. select fname, lname from employee e, project p, works_on w where pname = 'OperatingSystems' and e.ssn=w.essn and p.pnumber=w.pno
INTERSECT
select fname, lname from employee e, department d where dname = 'Software' and e.dno=d.dnumber;

Açıklama: Intersect ifadesi "ve" sözcüğünün karşılığıdır; iki kümenin kesişimini verir.

2. select fname, lname from employee e, project p, works_on w where pname = 'OperatingSystems' and e.ssn=w.essn and p.pnumber=w.pno
UNION
select fname, lname from employee e, department d where dname = 'Software' and e.dno=d.dnumber;

Açıklama: Union ifadesi "veya" sözcüğünün karşılığıdır; iki kümenin birleşimini verir.

3. select fname, lname from employee e, project p, works_on w where pname = 'OperatingSystems' and e.ssn=w.essn and p.pnumber=w.pno
EXCEPT
select fname, lname from employee e, department d where dname = 'Software' and e.dno=d.dnumber;

Açıklama: Except ifadesi "fark/hariç" sözcüğünün karşılığıdır; iki ayrı kümenin farkını verir.

4. select fname, lname from employee e where not exists (select null from department d where d.mgrssn = e.ssn) and not exists (select null from employee e2 where e2.superssn = e.ssn);

Açıklama: Not exists ifadesinden sonra gelen alt sorgunun sonuç kümesi, dış sorgunun (asıl sorgu) sonucunda dönen satırların içinde varsa bu satırlar elenerek sonuç elde edilir.

5. select dname from department where dnumber in (select dno from employee where fname ='John')

6. select count(*) as toplam, min(salary), max(salary), avg(salary), sum(salary) from employee, department where e.dno = d.dnumber and dname = 'Sales'