

BTEP104 – Lab Oturumu

Tablo Birleřtirmeler – (JOIN)

Birincil anahtar – Yabancı anahtar

4 farklı Join Çeşidi bulunur:

1. Equi-Join (İki tablo arasında Birincil Anahtar(BA) ve Yabancı Anahtar(YA) kullanılarak kurulan bağlantıdır. Sadece eşleşen veriler görüntülenir)
2. Non-Equi Join (İki tablo arasında BA ile YA kullanılmadan oluşturulan bağlantı çeşididir.)
3. Self-Join (Bir tablonun kendi kendisi ile birleştirilmesidir. Sadece eşleşen veriler ilgili görüntülenir.)
4. Outer-Join (Harici Birleřtime – Eşleşmeyen veriler dahi görüntülenebilir.)

Self Join

Soru 1:

Çalışanların (employees) isimleri ile birlikte Yöneticilerinin (Managers) isimlerini görüntüleyiniz.

Yazım kuralı:

```
Select tablo1.sutun_adi, tablo2.sutun_adi
```

```
From tablo1 , tablo2
```

```
Where Birincil Anahtar-Yabancı anahtar;
```

Çalışanların(Employees) tablosunun, Birincil anahtarları→Çalışanların Yöneticilerini (Managers (aslında onlar da employee'dir) temsil eder, ve yine aynı tabloya Yabancı anahtar olarak eklenir.
Employee_Id – Manager_Id

1. calisan.employee_id = yönetici.manager_id X

2. calisan.manager_id=yönetici.manager_id X --yöneticiler de aslında birer çalışandır(!)

3. calisan.manager_id=yönetici.employee_id ✓

Yabancı Anahtar Birincil Anahtar

Geleneksel Yöntem:

```
select c.last_name CALISANLAR, y.last_name YONETICILERI  
from employees c, employees y  
where c.manager_id =y.employee_id;
```

ANSI Yöntemi:

```
select c.last_name CALISANLAR, y.last_name YONETICILERI  
from employees c JOIN employees y on(c.manager_id =y.employee_id);
```

Soru 2: 'KING' soyisimli yönetici tarafından yönetilen elemanların sayısını bulunuz.

Geleneksel Yöntem:

```
Select count(*)  
from employees c, employees y  
where upper(y.last_name)='KING' and c.manager_id=y.employee_id;
```

ANSI Yöntemi:

```
Select count(*)  
from employees c JOIN employees y on(c.manager_id=y.employee_id)  
and upper(y.last_name)='KING';
```

Soru 3: Her yöneticinin kaç çalışını yönettiğini gösteren bir sorgu yazalım. Ancak en az 3 çalışını olan yöneticileri görüntüleyelim.

Geleneksel Yöntem:

```
select y.first_name||' '||y.last_name YONETICILER, count(*) "CALISAN SAYISI"  
from employees y, employees c  
where c.manager_id=y.employee_id  
group by y.first_name||' '||y.last_name  
having count(*)>=3;
```

ANSI Yöntemi:

```
select y.first_name||' '||y.last_name YONETICILER, count(*) "CALISAN SAYISI"  
from employees y, employees c on(c.manager_id=y.employee_id)  
group by y.first_name||' '||y.last_name  
having count(*)>=3;
```

Harici birleştirme (Outer JOIN)

Soru 4: Çalışanları ve çalıştıkları bölümleri görüntüleyen bir sorgu yazalım. Herhangi bir bölüme dahil olmayan tüm çalışanları dahi sorgu sonucunda görüntüleyelim.

Geleneksel Yöntem:

```
select calisan.first_name, calisan.last_name, bolum.department_id  
from employees calisan, departments bolum  
where calisan.department_id=bolum.department_id(+);
```

ANSI Yöntemi:

```
select calisan.first_name, calisan.last_name, bolum.department_id  
from employees calisan LEFT OUTER JOIN departments bolum  
on(calisan.department_id=bolum.department_id);
```

Soru 5: Çalışanları ve çalıştıkları bölümleri görüntüleyen bir sorgu yazalım. Herhangi bir çalışını olmayan tüm bölümleri dahi sorgu sonucunda görüntüleyelim.

Geleneksel Yöntem:

```
select calisan.first_name, calisan.last_name, bolum.department_id  
from employees calisan, departments bolum  
where calisan.department_id(+)=bolum.department_id;
```

ANSI Yöntemi:

```
select calisan.first_name, calisan.last_name, bolum.department_id  
from employees calisan RIGHT OUTER JOIN departments bolum  
on(calisan.department_id=bolum.department_id);
```

Önemli Not: Geleneksel Yöntemde, her iki tablonun eşleşmeyen satırlarını sorgu sonucuna eklemek mümkün olmaz.

```
select calisan.first_name, calisan.last_name, bolum.department_id
from employees calisan, departments bolum
where calisan.department_id(+) = bolum.department_id(+); //HATALI JOIN CUMLESİ
```

Ancak ANSI Yöntemi ile mümkündür.

```
select calisan.first_name, calisan.last_name, bolum.department_id
from employees calisan FULL OUTER JOIN departments bolum
on(calisan.department_id = bolum.department_id);
```

NON-EQUI JOIN (BA YA dışındaki sütunları kullanarak birleştirme)

Soru 6: Zlotkey isimli çalışandan önce işe alınan tüm çalışanları listeleyelim.

Geleneksel Yöntem:

```
select calisan.first_name, calisan.last_name
from employees calisan, employees zlotkey
where calisan.hire_date < zlotkey.hire_date and upper(zlotkey.last_name) = 'ZLOTKEY';
```

ANSI Yöntemi:

```
select calisan.first_name, calisan.last_name
from employees calisan JOIN employees zlotkey
on(calisan.hire_date < zlotkey.hire_date) and upper(zlotkey.last_name) = 'ZLOTKEY';
```

Soru 7: Zlotkey isimli çalışandan daha fazla maaş alan tüm çalışanları listeleyelim.

Geleneksel Yöntem:

```
select calisan.first_name, calisan.last_name
from employees calisan, employees zlotkey
where calisan.salary > zlotkey.salary and upper(zlotkey.last_name) = 'ZLOTKEY';
```

ANSI Yöntemi:

```
select calisan.first_name, calisan.last_name
from employees calisan JOIN employees zlotkey
on(calisan.salary > zlotkey.salary) and upper(zlotkey.last_name) = 'ZLOTKEY';
```

ALT SORGULAR (SUB-QUERY)

İç içe sorgu yazabileceğimiz bir yöntemdir. Bazı durumlar için JOIN yönteminin alternatifi olarak da kullanılabilir.

Yazım Kuralı:

```
Select sutun....
From tablo
Where sutun OPERATOR (Select sutun
                        From tablo
                        Where alt sorgu / koşul);
```

Soru 8: IT bölümünde çalışanların isim, soyisim ve maaşlarını görüntüleyiniz.

ALTSORGU

```
select first_name,last_name,salary
from employees
where department_id = (select department_id from departments
                        where upper(department_name)='IT');
```

JOIN

```
select first_name,last_name,salary
from employees c, departments b
where c.department_id = b.department_id and upper(b.department_name)='IT';
```

Soru 9: En düşük maaş alan çalışanı (isim, soyisim ve maas) görüntüleyiniz.

BUNU YAPMAYIN!!!

```
select first_name, last_name, salary from employees where salary = min(salary);
```

GRUP FONKSİYONLARINI WHERE CÜMLESİNDE KULLANAMAYIZ.

```
select first_name, last_name, salary
from employees
where salary = (select min(salary) from employees);
```

Soru 10: 7. Soruyu ALT sorgu yöntemi kullanarak tekrardan oluşturunuz.

```
select first_name, last_name
from employees
where salary >(select salary
                from employees
                where upper(last_name)='ZLOTKEY');
```