
SQL

Structured Query Language
Yapılandırılmış Sorgu Dili

DERS 07

SUBQUERIES

EXISTS, IS NULL KOMUTLARI

ORDER BY, GROUP By

Mehmet Bulutluoz
Elektronik muh.

SUBQUERIES

SORU 4- Her şirketin ismini, personel sayısını ve personelin aldığı max. ve min. maaşı listeleyen bir QUERY yazın.

```
SELECT şirket, personel_sayisi, (SELECT MAX(maaş)
                                FROM personel
                                WHERE şirketler.şirket=personel.şirket
                                ) max_maaş,
    (SELECT MIN(maaş)
     FROM personel
     WHERE şirketler.şirket=personel.şirket
     ) min_maaş
FROM şirketler;
```

| SİRKET | PERSONEL_SAYISI | MAX_MAAS | MIN_MAAS |
|---------|-----------------|----------|----------|
| Honda | 12000 | 3000 | 2500 |
| Ford | 18000 | 1500 | 1000 |
| Hyundai | 10000 | 7000 | 7000 |
| Toyota | 21000 | 1500 | 1500 |

SUBQUERIES

SORU 5- Her sirketin ismini ve personel sayisini ve iscilere odedigi toplam maasi listeleyen bir QUERY yazin.

```
SELECT sirket,personel_sayisi,(SELECT SUM(maas)
                                FROM personel
                                WHERE sirketler.sirket=personel.sirket
                                ) toplam_maas
FROM sirketler;
```

| SIRKET | PERSONEL_SAYISI | TOPLAM_MAAS |
|---------|-----------------|-------------|
| Honda | 12000 | 8000 |
| Ford | 18000 | 2500 |
| Hyundai | 10000 | 7000 |
| Toyota | 21000 | 1500 |

TECHPROED

EXISTS CONDITION

EXISTS Condition subquery'ler ile kullanılır. IN ifadesinin kullanımına benzer olarak, EXISTS ve NOT EXISTS ifadeleri de alt sorgudan getirilen değerlerin içerisinde bir değerın olması veya olmaması durumunda işlem yapılmasını sağlar.

```
CREATE TABLE mart_satislar  
(  
    urun_id number(10),  
    musteri_isim varchar2(50),  
    urun_isim varchar2(50)  
);
```

```
INSERT INTO mart_satislar VALUES (10, 'Mark', 'Honda');  
INSERT INTO mart_satislar VALUES (10, 'Mark', 'Honda');  
INSERT INTO mart_satislar VALUES (20, 'John', 'Toyota');  
INSERT INTO mart_satislar VALUES (30, 'Amy', 'Ford');  
INSERT INTO mart_satislar VALUES (20, 'Mark', 'Toyota');  
INSERT INTO mart_satislar VALUES (10, 'Adem', 'Honda');  
INSERT INTO mart_satislar VALUES (40, 'John', 'Hyundai');  
INSERT INTO mart_satislar VALUES (20, 'Eddie', 'Toyota');
```

```
CREATE TABLE nisan_satislar  
(  
    urun_id number(10),  
    musteri_isim varchar2(50),  
    urun_isim varchar2(50)  
);
```

```
INSERT INTO nisan_satislar VALUES (10, 'Hasan', 'Honda');  
INSERT INTO nisan_satislar VALUES (10, 'Kemal', 'Honda');  
INSERT INTO nisan_satislar VALUES (20, 'Ayse', 'Toyota');  
INSERT INTO nisan_satislar VALUES (50, 'Yasar', 'Volvo');  
INSERT INTO nisan_satislar VALUES (20, 'Mine', 'Toyota');
```


EXISTS CONDITION

Her iki ayda da aynı id ile satılan ürünlerin ürün_id'lerini ve ürünleri mart ayında alanların isimlerini getiren bir query yazınız..

| URUN_ID | MUSTERI_ISIM | URUN_ISIM |
|---------|--------------|-----------|
| 10 | Mark | Honda |
| 10 | Mark | Honda |
| 20 | John | Toyota |
| 30 | Amy | Ford |
| 20 | Mark | Toyota |
| 10 | Adem | Honda |
| 40 | John | Hyundai |
| 20 | Eddie | Toyota |

| URUN_ID | MUSTERI_ISIM | URUN_ISIM |
|---------|--------------|-----------|
| 10 | Hasan | Honda |
| 10 | Kemal | Honda |
| 20 | Ayşe | Toyota |
| 50 | Yasar | Volvo |
| 20 | Mine | Toyota |

| URUN_ID | MUSTERI_ISIM |
|---------|--------------|
| 10 | Mark |
| 10 | Mark |
| 10 | Adem |
| 20 | John |
| 20 | Mark |
| 20 | Eddie |

```
SELECT musteri_isim
FROM mart_satislar
WHERE EXISTS (SELECT ürün_id
               FROM nisan_satislar
               WHERE mart_satislar.ürün_id = nisan_satislar.ürün_id);
```

EXISTS CONDITION

Her iki ayda da satılan urun_isimleri aynı ürünlerin urun_isim'ini ve ürünleri nisan ayında alanların isimlerini getiren bir query yazınız..

```
SELECT urun_isim, musteri_isim
FROM nisan_satislar
WHERE EXISTS (SELECT urun_isim
                FROM mart_satislar
                WHERE mart_satislar.urun_isim = nisan_satislar.urun_isim);
```

| URUN_ISIM | MUSTERI_ISIM |
|-----------|--------------|
| Honda | Hasan |
| Honda | Kemal |
| Toyota | Ayşe |
| Toyota | Mine |

```
SELECT musteri_isim
FROM nisan_satislar
WHERE NOT EXISTS (SELECT urun_isim
                   FROM mart_satislar
                   WHERE mart_satislar.urun_isim = nisan_satislar.urun_isim);
```

| URUN_ISIM | MUSTERI_ISIM |
|-----------|--------------|
| Volvo | Yasar |

IS NULL CONDITION

Arama yapılan field'da NULL degeri almıs kayıtları getirir.

```
CREATE TABLE insanlar  
(  
  ssn char(9),  
  isim varchar2(50),  
  adres varchar2(50)  
);
```

```
INSERT INTO insanlar VALUES(123456789, 'Ali Can', 'Istanbul');  
INSERT INTO insanlar VALUES(234567890, 'Veli Cem', 'Ankara');  
INSERT INTO insanlar VALUES(345678901, 'Mine Bulut', 'Izmir');  
INSERT INTO insanlar (ssn, adres) VALUES(456789012, 'Bursa');  
INSERT INTO insanlar (ssn, adres) VALUES(567890123, 'Denizli');
```

| SSN | NAME | ADDRESS |
|-----------|------------|----------|
| 123456789 | Ali Can | Istanbul |
| 234567890 | Veli Cem | Ankara |
| 345678901 | Mine Bulut | Izmir |
| 456789012 | - | Bursa |
| 567890123 | - | Denizli |

```
SELECT *  
FROM insanlar  
WHERE isim IS NULL;
```

| SSN | NAME | ADDRESS |
|-----------|------|---------|
| 456789012 | - | Bursa |
| 567890123 | - | Denizli |

```
SELECT *  
FROM insanlar  
WHERE isim IS NOT NULL;
```

| SSN | NAME | ADDRESS |
|-----------|------------|---------|
| 123456789 | Ali Can | - |
| 234567890 | Veli Cem | Ankara |
| 345678901 | Mine Bulut | Izmir |

--isim field'i bos olan kayıtlara isim olarak "Isim Girilmemistir" yazdirin

```
UPDATE insanlar  
SET isim = 'Isim Girilmemis'  
WHERE name IS NULL;
```

| SSN | NAME | ADDRESS |
|-----------|-----------------|---------|
| 456789012 | Isim Girilmemis | Bursa |
| 567890123 | Isim Girilmemis | Denizli |

ORDER BY CLAUSE

ORDER BY komutu belli bir field'a gore NATURAL ORDER olarak siralama yapmak icin kullanilir

ORDER BY komutu sadece **SELECT** komutu ile kullanilir

```
CREATE TABLE insanlar  
(  
  ssn char(9),  
  isim varchar2(50),  
  soyisim varchar2(50),  
  adres varchar2(50)  
);
```

```
INSERT INTO insanlar VALUES(123456789, 'Ali','Can', 'Istanbul');  
INSERT INTO insanlar VALUES(234567890, 'Veli','Cem', 'Ankara');  
INSERT INTO insanlar VALUES(345678901, 'Mine','Bulut', 'Ankara');  
INSERT INTO insanlar VALUES(256789012, 'Mahmut','Bulut', 'Istanbul ');  
INSERT INTO insanlar VALUES(344678901, 'Mine','Yasa', 'Ankara');  
INSERT INTO insanlar VALUES(345678901, 'Veli','Yilmaz', 'Istanbul ');
```

Insanlar tablosundaki datalari adres'e gore siralayin

```
SELECT *  
FROM insanlar  
ORDER BY adres;
```

| SSN | ISIM | SOYISIM | ADRES | SSN | ISIM | SOYISIM | ADRES |
|-----------|--------|---------|----------|-----------|--------|---------|----------|
| 123456789 | Ali | Can | Istanbul | 345678901 | Mine | Bulut | Ankara |
| 234567890 | Veli | Cem | Ankara | 344678901 | Mine | Yasa | Ankara |
| 345678901 | Mine | Bulut | Ankara | 234567890 | Veli | Cem | Ankara |
| 256789012 | Mahmut | Bulut | Istanbul | 123456789 | Ali | Can | Istanbul |
| 344678901 | Mine | Yasa | Ankara | 345678901 | Veli | Yilmaz | Istanbul |
| 345678901 | Veli | Yilmaz | Istanbul | 256789012 | Mahmut | Bulut | Istanbul |

ORDER BY CLAUSE

Insanlar tablosundaki ismi Mine olanlari SSN sirali olarak listeleyin

```
SELECT *  
FROM insanlar  
WHERE isim='Mine'  
ORDER BY ssn;
```

| SSN | ISIM | SOYISIM | ADRES |
|-----------|------|---------|--------|
| 344678901 | Mine | Yasa | Ankara |
| 345678901 | Mine | Bulut | Ankara |

NOT : Order By komutundan sonra field ismi yerine field numarasi da kullanilabilir

Insanlar tablosundaki soyismi Bulut olanlari isim sirali olarak listeleyin

```
SELECT *  
FROM insanlar  
WHERE soyisim='Bulut'  
ORDER BY 2;
```

| SSN | ISIM | SOYISIM | ADRES |
|-----------|--------|---------|----------|
| 256789012 | Mahmut | Bulut | Istanbul |
| 345678901 | Mine | Bulut | Ankara |

ORDER BY field_name DESC CLAUSE

Insanlar tablosundaki tum kayitlari SSN numarasi buyukten kucuge olarak siralayin

```
SELECT *  
FROM insanlar  
ORDER BY ssn DESC;
```

| SSN | ISIM | SOYISIM | ADRES |
|-----------|--------|---------|----------|
| 345678901 | Mine | Bulut | Ankara |
| 345678901 | Veli | Yilmaz | Istanbul |
| 344678901 | Mine | Yasa | Ankara |
| 256789012 | Mahmut | Bulut | Istanbul |
| 234567890 | Veli | Cem | Ankara |
| 123456789 | Ali | Can | Istanbul |

Insanlar tablosundaki tum kayitlari isimler Natural sirali, Soyisimler ters sirali olarak listeleyin

```
SELECT *  
FROM insanlar  
ORDER BY isim ASC, soyisim DESC;
```

| SSN | ISIM | SOYISIM | ADRES |
|-----------|--------|---------|----------|
| 123456789 | Ali | Can | Istanbul |
| 256789012 | Mahmut | Bulut | Istanbul |
| 344678901 | Mine | Yasa | Ankara |
| 345678901 | Mine | Bulut | Ankara |
| 345678901 | Veli | Yilmaz | Istanbul |
| 234567890 | Veli | Cem | Ankara |

ALIASES

Aliases kodu ile tablo yazdirilirken, field isimleri sadece o cikti icin degistirilebilir

```
CREATE TABLE employees  
(  
  employee_id char(9),  
  employee_name varchar2(50),  
  employee_birth_city varchar2(50)  
);
```

```
INSERT INTO employees VALUES(123456789, 'Ali Can', 'Istanbul');  
INSERT INTO employees VALUES(234567890, 'Veli Cem', 'Ankara');  
INSERT INTO employees VALUES(345678901, 'Mine Bulut', 'Izmir');
```

| EMPLOYEE_ID | EMPLOYEE_NAME | EMPLOYEE_BIRTH_CITY |
|-------------|---------------|---------------------|
| 123456789 | Ali Can | Istanbul |
| 234567890 | Veli Cem | Ankara |
| 345678901 | Mine Bulut | Izmir |

```
SELECT employee_id AS id, employee_name AS isim, employee_birth_city AS dogum_yeri  
FROM employees;
```

| ID | ISIM | DOGUM_YERI |
|-----------|------------|------------|
| 123456789 | Ali Can | Istanbul |
| 234567890 | Veli Cem | Ankara |
| 345678901 | Mine Bulut | Izmir |

```
SELECT employee_id AS id, employee_name || employee_birth_city AS isim_ve_dogum_yeri  
FROM employees;
```

| ID | ISIM_VE_DOGUM_YERI |
|-----------|--------------------|
| 123456789 | Ali CanIstanbul |
| 234567890 | Veli CemAnkara |
| 345678901 | Mine BulutIzmir |

GROUP BY CLAUSE

Group By komutu sonuçları bir veya daha fazla sütuna göre gruplamak için **SELECT** komutuyla birlikte kullanılır

```
CREATE TABLE manav
(  
  isim varchar2(50),  
  Urun_adi varchar2(50),  
  Urun_miktar number(9)  
);
```

```
INSERT INTO manav VALUES( 'Ali', 'Elma', 5);  
INSERT INTO manav VALUES( 'Ayse', 'Armut', 3);  
INSERT INTO manav VALUES( 'Veli', 'Elma', 2);  
INSERT INTO manav VALUES( 'Hasan', 'Uzum', 4);  
INSERT INTO manav VALUES( 'Ali', 'Armut', 2);  
INSERT INTO manav VALUES( 'Ayse', 'Elma', 3);  
INSERT INTO manav VALUES( 'Veli', 'Uzum', 5);  
INSERT INTO manav VALUES( 'Ali', 'Armut', 2);  
INSERT INTO manav VALUES( 'Veli', 'Elma', 3);  
INSERT INTO manav VALUES( 'Ayse', 'Uzum', 2);
```

| ISIM | URUN_ADI | URUN_MIKTAR |
|-------|----------|-------------|
| Ali | Elma | 5 |
| Ayse | Armut | 3 |
| Veli | Elma | 2 |
| Hasan | Uzum | 4 |
| Ali | Armut | 2 |
| Ayse | Elma | 3 |
| Veli | Uzum | 5 |
| Ali | Armut | 2 |
| Veli | Elma | 3 |
| Ayse | Uzum | 2 |

1) Isme gore alinan toplam urunleri bulun

```
SELECT isim, SUM(urun_miktar) AS Alinan_Toplam_Meyve  
FROM manav  
GROUP BY isim;
```

| ISIM | ALINAN_TOPLAM_MEYVE |
|-------|---------------------|
| Veli | 10 |
| Ayse | 8 |
| Ali | 9 |
| Hasan | 4 |

GROUP BY CLAUSE

| ISIM | URUN_ADI | URUN_MIKTAR |
|-------|----------|-------------|
| Ali | Elma | 5 |
| Ayşe | Armut | 3 |
| Veli | Elma | 2 |
| Hasan | Uzum | 4 |
| Ali | Armut | 2 |
| Ayşe | Elma | 3 |
| Veli | Uzum | 5 |
| Ali | Armut | 2 |
| Veli | Elma | 3 |
| Ayşe | Uzum | 2 |

2) Urun ismine gore urunu alan toplam kisi sayisi

```
SELECT urun_adi, COUNT(isim) AS Urunu_Alan_Kisi_Sayisi  
FROM manav  
GROUP BY urun_adi;
```

| URUN_ADI | URUNU_ALAN_KISI_SAYISI |
|----------|------------------------|
| Elma | 4 |
| Uzum | 3 |
| Armut | 3 |

3) Alinan kilo miktarina gore musteri sayisi

```
SELECT urun_miktar, COUNT(isim) AS Urun_Miktarini_Alan_Kisi_Sayisi  
FROM manav  
GROUP BY urun_miktar;
```

| URUN_MIKTAR | URUN_MIKTARINI_ALAN_KISI_SAYISI |
|-------------|---------------------------------|
| 2 | 4 |
| 5 | 2 |
| 4 | 1 |
| 3 | 3 |

TECHPROED