

---

# SQL

Structured Query Language  
Yapılandırılmış Sorgu Dili

19 Ocak 2021  
DERS 08

EXISTS, IS NULL  
ORDER BY, ALIASES

Mehmet Bulutluoz  
Elektronik muh.

---

---

## Önceki Dersten Hatırladıklarımız

1- Subquery : iç içe query demektir. Birden fazla tabloyu ilgilendiren update'ler için kullanabiliriz. Update yapılacak tablodan kodumuzu yazmaya başlıyoruz, sonra diğer tablodan bilgi almak için subquery ekleyip SELECT komutu ile ilgili datayı alıyoruz

2- SELECT komutu ile de subquery kullanılabilir

- Eğer WHERE den sonra subquery kullanacaksak sorgunun sonucunu değerlendirmek gerekir.

- Eğer sorgu 1 sonuc getirecekse = isareti kullanabiliriz

- Ama sorgunun kaç sonuc getireceğini bilmiyorsak veya birden fazla sonuc getireceğini biliyorsak bu durumda = yerine IN kullanılır

- Eğer SELECT komutunun olduğu satırda SUBQUERY kullanacaksak sonucun 1 tek değer dondurmeleri gerekir, bunun için AVG,MIN,MAX,COUNT,SUM gibi AGGREGATE fonksiyonları kullanılır. Aggregate fonksiyonları araya , konulmak şartıyla istendigi kadar kullanılabilir

- Aggregate fonksiyonu yeni bir sütun oluşturduğunda sütun ismini atamak için AS istenimsim komutu kullanılır.

---

# 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 ayni id ile satilan urunlerin urun\_id'lerini ve urunleri mart ayinda alanlarin isimlerini getiren bir query yaziniz..

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 urun_id
               FROM nisan_satislar
               WHERE mart_satislar.urun_id = nisan_satislar.urun_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)  
);
```

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

```
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
456789012	-	Bursa
567890123	-	Denizli

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

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

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

```
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 calisanlar  
(  
  calisan_id char(9),  
  calisan_isim varchar2(50),  
  calisan_dogdugu_sehir varchar2(50)  
);
```

```
INSERT INTO calisanlar VALUES(123456789, 'Ali Can', 'Istanbul');  
INSERT INTO calisanlar VALUES(234567890, 'Veli Cem', 'Ankara');  
INSERT INTO calisanlar 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 calisan_id AS id, calisan_isim AS isim, calisan_dogdugu_sehir AS dogum_yeri  
FROM calisanlar;
```

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

```
SELECT calisan_id AS id, calisan_isim || calisan_dogdugu_sehir AS isim_ve_dogum_yeri  
FROM calisanlar;
```

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

# GROUP BY CLAUSE

```
CREATE TABLE personel
(
  id number(9),
  isim varchar2(50),
  sehir varchar2(50),
  maas number(20),
  sirket varchar2(20)
);
```

```
INSERT INTO personel VALUES(123456789, 'Ali Yilmaz', 'Istanbul', 5500, 'Honda');
INSERT INTO personel VALUES(234567890, 'Veli Sahin', 'Istanbul', 4500, 'Toyota');
INSERT INTO personel VALUES(345678901, 'Mehmet Ozturk', 'Ankara', 3500, 'Honda');
INSERT INTO personel VALUES(456789012, 'Mehmet Ozturk', 'Izmir', 6000, 'Ford');
INSERT INTO personel VALUES(567890123, 'Mehmet Ozturk', 'Ankara', 7000, 'Tofas');
INSERT INTO personel VALUES(456789012, 'Veli Sahin', 'Ankara', 4500, 'Ford');
INSERT INTO personel VALUES(123456710, 'Hatice Sahin', 'Bursa', 4500, 'Honda');
```

1) Isme gore toplam maaslari bulun

```
SELECT isim, SUM(maas) AS toplam_maas
FROM personel
GROUP BY isim;
```

ISIM	TOPLAM_MAAS
Hatice Sahin	4500
Veli Sahin	9000
Ali Yilmaz	5500
Mehmet Ozturk	16500

2) sehre gore toplam personel sayisini bulun

```
SELECT sehir, COUNT(isim) AS calisan_sayisi
FROM personel
GROUP BY sehir;
```

SEHIR	CALISAN_SAYISI
Izmir	1
Bursa	1
Istanbul	2
Ankara	3

ID	ISIM	SEHIR	MAAS	SIRKET
123456789	Ali Yilmaz	Istanbul	5500	Honda
234567890	Veli Sahin	Istanbul	4500	Toyota
345678901	Mehmet Ozturk	Ankara	3500	Honda
456789012	Mehmet Ozturk	Izmir	6000	Ford
567890123	Mehmet Ozturk	Ankara	7000	Tofas
456789012	Veli Sahin	Ankara	4500	Ford
123456710	Hatice Sahin	Bursa	4500	Honda

# GROUP BY CLAUSE

3) Sirketlere gore maasi 5000 liradan fazla olan personel sayisini bulun

```
SELECT sirket, COUNT (*) AS calisan_sayisi  
FROM personel  
WHERE maas>5000  
GROUP BY sirket;
```

SIRKET	CALISAN_SAYISI
Honda	1
Ford	1
Tofas	1

4) Her sirket icin Min ve Max maasi bulun

```
SELECT sirket, MIN (maas) AS en_az_maas, MAX (maas) AS en_fazla_maas  
FROM personel  
GROUP BY sirket;
```

SIRKET	EN_AZ_MAAS	EN_FAZLA_MAAS
Honda	3500	5500
Ford	4500	6000
Toyota	4500	4500
Tofas	7000	7000