

---

# SQL

Structured Query Language  
Yapılandırılmış Sorgu Dili

**DERS 09**

**JOINS**

**LIKE, NOT LIKE CONDITIONS**

Mehmet Bulutluoz  
Elektronik muh.

---

---

# JOINS

2 **T**ablodaki datalari Birleştirmek için kullanılır.

Su ana kadar gördüğümüz Union, Intersect ve Minus sorgu sonuçları için kullanılır  
Tablolar için ise **J**JOIN kullanılır

5 Cesi Join vardır

- 1) INNER JOIN iki Tablodaki ortak datalari gösterir
- 2) LEFT JOIN İlk datada olan tüm recordlari gösterir
- 3) RIGHT JOIN İkinci tabloda olan tüm recordlari gösterir
- 4) FULL JOIN İki tablodaki tüm recordlari gösterir
- 5) SELF JOIN Bir tablonun kendi içinde Join edilmesi ile oluşur.

# INNER JOINS

```
CREATE TABLE sirketler  
(  
  sirket_id number(9),  
  sirket_isim varchar2(20)  
);
```

```
INSERT INTO sirketler VALUES(100, 'Toyota');  
INSERT INTO sirketler VALUES(101, 'Honda');  
INSERT INTO sirketler VALUES(102, 'Ford');  
INSERT INTO sirketler VALUES(103, 'Hyundai');
```

SIRKET_ID	SIRKET_ISIM
100	Toyota
101	Honda
102	Ford
103	Hyundai

```
CREATE TABLE siparisler  
(  
  siparis_id number(9),  
  sirket_id number(9),  
  siparis_tarihi date  
);
```

```
INSERT INTO siparisler VALUES(11, 101, '17-Apr-2020');  
INSERT INTO siparisler VALUES(22, 102, '18-Apr-2020');  
INSERT INTO siparisler VALUES(33, 103, '19-Apr-2020');  
INSERT INTO siparisler VALUES(44, 104, '20-Apr-2020');  
INSERT INTO siparisler VALUES(55, 105, '21-Apr-2020');
```

SIPARIS_ID	SIRKET_ID	SIPARIS_TARIHI
11	101	17-APR-20
22	102	18-APR-20
33	103	19-APR-20
44	104	20-APR-20
55	105	21-APR-20

# INNER JOINS

TABLE 1

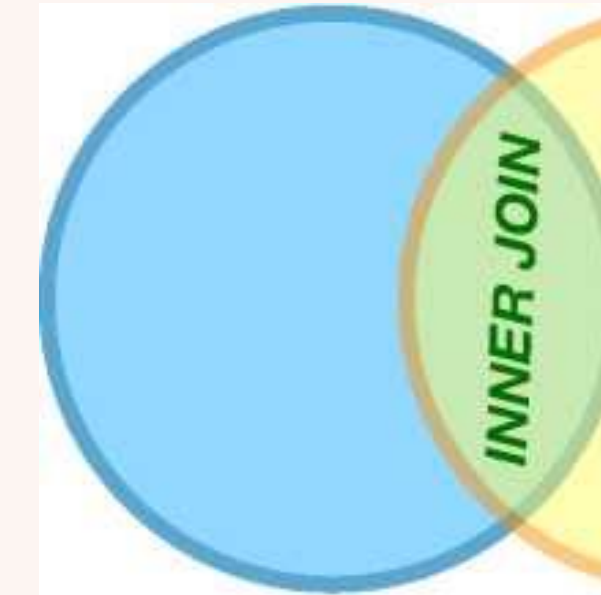


TABLE 2

**SORU)** İki Tabloda `sirket_id`'si aynı olanların `sirket_ismi`, `siparis_id` ve `siparis_tarihleri` ile yeni bir tablo oluşturun

```
SELECT sirketler.sirket_isim, siparisler. siparis_id, siparisler. siparis_tarihi  
FROM sirketler INNER JOIN siparisler  
ON sirketler.sirket_id = siparisler.sirket_id;
```

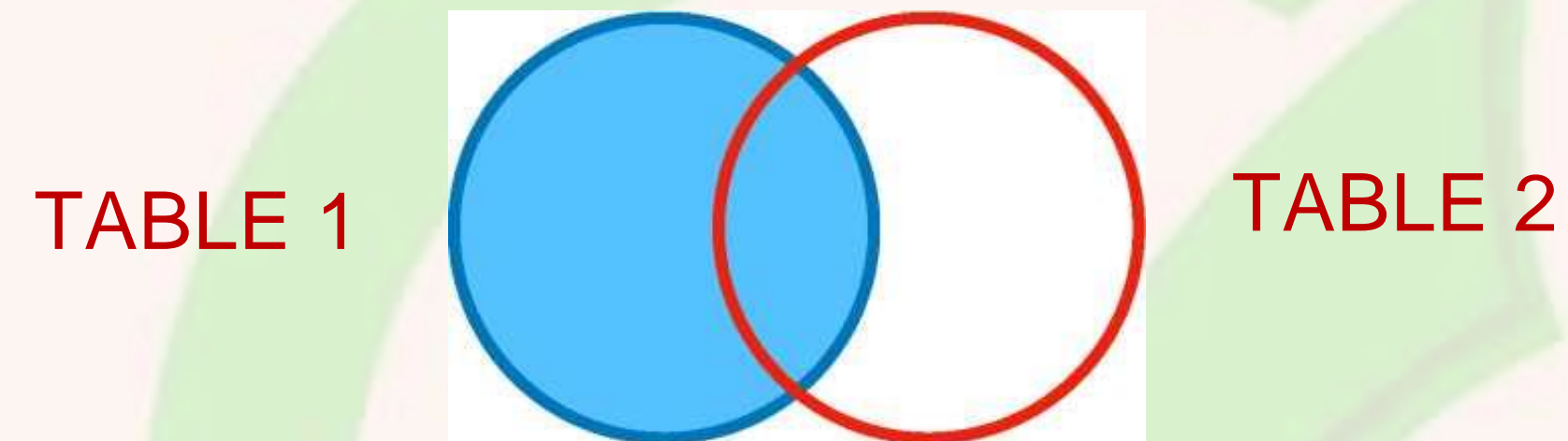
SIRKET_ISIM	SIPARIS_ID	SIPARIS_TARIHI
Honda	11	17-APR-20
Ford	22	18-APR-20
Hyundai	33	19-APR-20

**NOT :**

- 1) Select'ten sonra tabloda görmek istediğiniz sütunları yazarken **Tablo\_adi.field\_adi** şeklinde yazın
- 2) From'dan sonra tablo ismi yazarken **1.Tablo ismi + INNER JOIN + 2.Tablo ismi** yazmalıyız
- 3) Join'i hangi kurala göre yapacağınızı belirtmelisiniz. Bunun için **ON+ kuralımız** yazılmalı



# LEFT JOINS



```
SELECT sirketler.sirket_isim, siparisler. siparis_id, siparisler. siparis_tarihi  
FROM sirketler LEFT JOIN siparisler  
ON sirketler.sirket_id = siparisler.sirket_id;
```

SIRKET_ISIM	SIPARIS_ID	SIPARIS_TARIHI
Honda	11	17-APR-20
Ford	22	18-APR-20
Hyundai	33	19-APR-20
Toyota	-	-

## NOT :

- 1) Left Join'de ilk tablodaki tum record'lar gosterilir.
- 2) İlk tablodaki datalara 2.tablodan gelen ek datalar varsa bu ek datalar ortak datalar icin gosterilir ancak ortak olmayan datalar icin o kisimler bos kalir
- 3) İlk yazdiginiz Tablonun tamamini aldigi icin hangi tabloyu istedigimize karar verip once onu yazmaliyiz

# RIGHT JOINS

TABLE 1

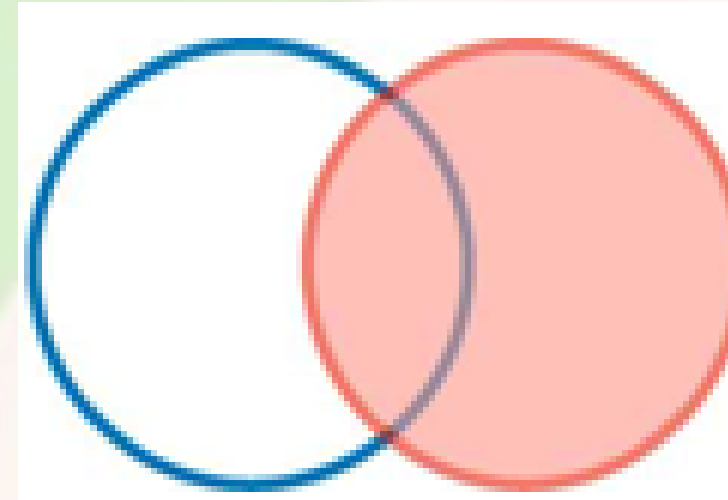


TABLE 2

```
SELECT sirketler.sirket_isim, siparisler. siparis_id, siparisler. siparis_tarihi  
FROM sirketler RIGHT JOIN siparisler  
ON sirketler.sirket_id = siparisler.sirket_id;
```

SIRKET_ISIM	SIPARIS_ID	SIPARIS_TARIHI
Honda	11	17-APR-20
Ford	22	18-APR-20
Hyundai	33	19-APR-20
-	55	21-APR-20
-	44	20-APR-20

NOT :

- 1) Right Join'de ikinci tablodaki tum record'lar gosterilir.
- 2) Ikinci tablodaki datalara 1.tablodan gelen ek datalar varsa bu ek datalar ortak datalar icin gosterilir ancak ortak olmayan datalar icin o kisimler bos kalir

# FULL JOINS

```
SELECT sirketler.sirket_isim, siparisler. siparis_id, siparisler. siparis_tarihi  
FROM sirketler FULL JOIN siparisler  
ON sirketler.sirket_id = siparisler.sirket_id;
```

NOT :

- 1) FULL Join'de iki tabloda var olan tum record'lar gosterilir.
- 2) Bir tabloda olup otekinde olmayan data'lar bos kalir

SIRKET_ISIM	SIPARIS_ID	SIPARIS_TARIHI
Honda	11	17-APR-20
Ford	22	18-APR-20
Hyundai	33	19-APR-20
-	44	20-APR-20
-	55	21-APR-20
Toyota	-	-

TECHPROED

# SELF JOINS

```
CREATE TABLE personel
```

```
(  
  id number(2),  
  isim varchar2(20),  
  title varchar2(60),  
  yonetici_id number(2)  
);
```

```
INSERT INTO personel VALUES(1, 'Ali Can', 'SDET', 2);  
INSERT INTO personel VALUES(2, 'Veli Cem', 'QA', 3);  
INSERT INTO personel VALUES(3, 'Ayse Gul', 'QA Lead', 4);  
INSERT INTO personel VALUES(4, 'Fatma Can', 'CEO', 5);
```

ID	ISIM	TITLE	YONETICI_ID
1	Ali Can	SDET	2
2	Veli Cem	QA	3
3	Ayse Gul	QA Lead	4
4	Fatma Can	CEO	5

Her personelin yanina yonetici ismini yazdiran bir tablo olusturun

```
SELECT p1.isim AS personel_ismi, p2.isim AS yonetici_ismi  
FROM personel p1 INNER JOIN personel p2  
ON p1.yonetici_id = p2.id;
```

PERSONEL_ISMI	YONETICI_ISMI
Ali Can	Veli Cem
Veli Cem	Ayse Gul
Ayse Gul	Fatma Can



# LIKE Condition

LIKE condition WHERE ile kullanılarak SELECT, INSERT, UPDATE, veya DELETE statement ile calisan wildcards'a izin verir.. Ve bize pattern matching yapma imkani verir.

```
CREATE TABLE musteriler  
(  
id number(10) UNIQUE,  
isim varchar2(50) NOT NULL,  
gelir number(6)  
);
```

```
INSERT INTO musteriler (id, isim, gelir) VALUES (1001, 'Ali', 62000);  
INSERT INTO musteriler (id, isim, gelir) VALUES (1002, 'Ayse', 57500);  
INSERT INTO musteriler (id, isim, gelir) VALUES (1003, 'Feride', 71000);  
INSERT INTO musteriler (id, isim, gelir) VALUES (1004, 'Fatma', 42000);  
INSERT INTO musteriler (id, isim, gelir) VALUES (1005, 'Kasim', 44000);
```

1) % => 0 veya birden fazla karakter belirtir

**SORU** : Ismi A harfi ile baslayan musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM musteriler  
WHERE isim LIKE 'A%';
```

ID	ISIM	GELIR
1001	Ali	62000
1002	Ayse	57500

ID	ISIM	GELIR
1001	Ali	62000
1002	Ayse	57500
1003	Feride	71000
1004	Fatma	42000
1005	Kasim	44000

## LIKE Condition

**SORU :** Ismi e harfi ile biten musterilerin isimlerini ve gelir'lerini yazdıran QUERY yazın

```
SELECT isim,gelir  
FROM musteriler  
WHERE isim LIKE '%e';
```

ISIM	GELIR
Ayşe	57500
Feride	71000

**SORU :** Isminin icinde er olan musterilerin isimlerini ve gelir'lerini yazdıran QUERY yazın

```
SELECT isim,gelir  
FROM musteriler  
WHERE isim LIKE '%er%';
```

ISIM	GELIR
Feride	71000

## LIKE Condition

2) \_ => sadece bir karakteri gösterir.

**SORU** : Ismi 5 harfli olup son 4 harfi atma olan musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM musteriler  
WHERE isim LIKE '_atma';
```

ID	ISIM	GELIR
1004	Fatma	42000

**SORU** : Ikinci harfi a olan musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM musteriler  
WHERE isim LIKE '_a%';
```

ID	ISIM	GELIR
1004	Fatma	42000
1005	Kasim	44000

**SORU** : Ucuncu harfi s olan musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM musteriler  
WHERE isim LIKE '___s%';
```

ID	ISIM	GELIR
1002	Ayşe	57500
1005	Kasim	44000

## LIKE Condition

**SORU** : Ucuncu harfi s olan ismi 4 harfli musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM musteriler  
WHERE isim LIKE '___s_';
```

ID	ISIM	GELIR
1002	Ayşe	57500

**SORU** : İlk harfi F olan en az 4 harfli musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM musteriler  
WHERE isim LIKE 'F_%_%_%';
```

ID	ISIM	GELIR
1003	Feride	71000
1004	Fatma	42000

**SORU** : Ikinci harfi a,4.harfi m olan musterilerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM musteriler  
WHERE isim LIKE '_a_m%';
```

ID	ISIM	GELIR
1004	Fatma	42000



## LIKE Condition

3) [ ] REGEXP\_LIKE => sadece bir karakteri gösterir.

```
CREATE TABLE kelimeler  
(  
  id number(10) UNIQUE,  
  kelime varchar2(50) NOT NULL,  
  Harf_sayisi number(6)  
);
```

```
INSERT INTO kelimeler VALUES (1001, 'hot', 3);  
INSERT INTO kelimeler VALUES (1002, 'hat', 3);  
INSERT INTO kelimeler VALUES (1003, 'hit', 3);  
INSERT INTO kelimeler VALUES (1004, 'hbt', 3);  
INSERT INTO kelimeler VALUES (1008, 'hct', 3);  
INSERT INTO kelimeler VALUES (1005, 'adem', 4);  
INSERT INTO kelimeler VALUES (1006, 'selim', 5);  
INSERT INTO kelimeler VALUES (1007, 'yusuf', 5);
```

**SORU :** İlk harfi h, son harfi t olup 2. harfi a veya i olan 3 harfli kelimelerin tüm bilgilerini yazdıran QUERY yazın

```
SELECT *  
FROM kelimeler  
WHERE REGEXP_LIKE (kelime, 'h[ai]t');
```

# LIKE Condition

**SORU** : İlk harfi h,son harfi t olup 2.harfi a ile k arasinda olan 3 harfli kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE REGEXP_LIKE (kelime, 'h[a-k]t');
```

ID	KELIME	HARF_SAYISI
1002	hat	3
1003	hit	3
1004	hbt	3
1008	hct	3

**SORU** : Icinde m veya i olan kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE REGEXP_LIKE (kelime, '[mi](*) '); [a|n] de olur
```

ID	KELIME	HARF_SAYISI
1003	hit	3
1005	adem	4
1006	selim	5

**SORU** : a veya s ile baslayan kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE REGEXP_LIKE (kelime, '^[as] ');
```

ID	KELIME	HARF_SAYISI
1005	adem	4
1006	selim	5

# LIKE Condition

**SORU** : m veya f ile biten kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE REGEXP_LIKE (kelime, '[ea]$');
```

ID	KELIME	HARF_SAYISI
1005	adem	4
1006	selim	5
1007	yusuf	5

# NOT LIKE Condition

**SORU 1** : ilk harfi h olmayan kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE kelime NOT LIKE 'h%';
```

ID	KELIME	HARF_SAYISI
1005	adem	4
1006	selim	5
1007	yusuf	5

**SORU 2** : a harfi icermeyen kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE kelime NOT LIKE '%a%';
```

ID	KELIME	HARF_SAYISI
1001	hot	3
1003	hit	3
1004	hbt	3
1008	hct	3
1006	selim	5
1007	yusuf	5



# NOT LIKE Condition

**SORU 3** : ikinci ve ucuncu harfi 'de' olmayan kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE kelime NOT LIKE '_de%';
```

ID	KELIME	HARF_SAYISI
1001	hot	3
1002	hat	3
1003	hit	3
1004	hbt	3
1008	hct	3
1006	selim	5
1007	yusuf	5

**SORU 4** : 2. harfi e,i veya o olmayan kelimelerin tum bilgilerini yazdiran QUERY yazin

```
SELECT *  
FROM kelimeler  
WHERE NOT REGEXP_LIKE (kelime, '[_eio]');
```

ID	KELIME	HARF_SAYISI
1002	hat	3
1004	hbt	3
1008	hct	3
1007	yusuf	5

# UPPER – LOWER - INITCAP

Tabloları yazdırırken büyük harf, küçük harf veya ilk harfleri büyük diğerleri küçük harf yazdırmak için kullanırız

```
SELECT UPPER(kelime)
FROM kelimeler;
```

UPPER(KELIME)
HOT
HAT
HIT
HBT
HCT
ADEM
SELIM
YUSUF

```
SELECT LOWER(kelime)
FROM kelimeler;
```

LOWER(KELIME)
hot
hat
hit
hbt
hct
adem
selim
yusuf

```
SELECT INITCAP(kelime)
FROM kelimeler;
```

INITCAP(KELIME)
Hot
Hat
Hit
Hbt
Hct
Adem
Selim
Yusuf

# DISTINCT

URUN_ID	MUSTERI_ISIM	URUN_ISIM
10	Ali	Portakal
10	Ali	Portakal
20	Veli	Elma
30	Ayşe	Armut
20	Ali	Elma
10	Adem	Portakal
40	Veli	Kaysi
20	Elif	Elma

```
SELECT DISTINCT urun_isim  
FROM muster_iurun;
```

URUN_ISIM
Elma
Portakal
Kaysi
Armut

```
SELECT DISTINCT muster_iisim  
FROM muster_iurun;
```

MUSTERI_ISIM
Veli
Ayşe
Elif
Adem
Ali

Tabloda kaç farklı meyve vardır ?

```
SELECT COUNT(DISTINCT urun_isim) AS urun_cesit_sayisi  
FROM muster_iurun;
```

URUN_CESIT_SAYISI
4

TECHPR

# FETCH NEXT (SAYI) ROW ONLY- OFFSET

1) Tabloyu urun\_id ye gore siralayiniz

URUN_ID	MUSTERI_ISIM	URUN_ISIM
10	Ali	Portakal
10	Ali	Portakal
10	Adem	Portakal
20	Veli	Elma
20	Elif	Elma
20	Ali	Elma
30	Ayşe	Armut
40	Veli	Kaysi

2) Sirali tablodan ilk 3 kaydi listeleyin

```
SELECT *  
FROM musteri_urun  
ORDER BY urun_id  
FETCH NEXT 3 ROW ONLY;
```

URUN_ID	MUSTERI_ISIM	URUN_ISIM
10	Ali	Portakal
10	Adem	Portakal
10	Ali	Portakal

3) Sirali tablodan 4. kayittan 7.kayida kadar olan kayitlari listeleyin

```
SELECT *  
FROM musteri_urun  
ORDER BY urun_id  
OFFSET 3 ROW  
FETCH NEXT 4 ROW ONLY;
```

URUN_ID	MUSTERI_ISIM	URUN_ISIM
20	Veli	Elma
20	Elif	Elma
20	Ali	Elma
30	Ayşe	Armut