



VERİ TABANI VE YÖNETİM SİSTEMLERİ

Dr. Öğretim Üyesi
Alper Talha KARADENİZ



GENEL
DERS
İÇERİĞİ

- 1.İç İçe Select Yapısı
- 2.Karşılaştırma İfadeleri
- 3.Distinct
- 4.Order By
- 5.Where
- 6.Between
7. Like
8. In-As
9. Tarih Fonksiyonları
10. Datediff

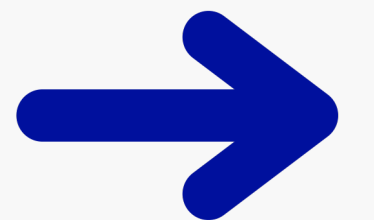




İç İçe Select Yapısı

Veritabanında bazen bir tablodan çektiğimiz bilgilerle başka bir tablodaki bilgileri sorgulamak isteyebiliriz. Bu tür sorgular iç içe sorgularla yapılabilir.

Bir sorgunun sonucunu başka bir sorguda filtreleme, karşılaştırma veya hesaplama yapmak için kullanabilirsin.





Örneğin elimizde iki tablo olsun. Birinde öğrenci bilgileri diğerinde ise öğrencilerin notları olsun.

ogr_id	ogr_ad	ogr_soyad	ogr_adres
1	Aylin	Durmaz	İstanbul
2	Beren	Aktaş	Ankara
3	Cem	Soylu	Bursa
4	Deniz	Aydın	Ankara

ogr_id	ogr_not
1	75
2	45
3	80
4	65

Notu 60'ın üzerinde olan öğrencilerin bilgileri istenirse yapılacak sorgu:

```
SELECT * FROM ogrenci
WHERE ogr_id
IN (select ogr_id from notlar where ogr_not > 60)
```



ÖRNEKLER

```
SELECT * FROM personel  
WHERE gorev_id = (SELECT id FROM gorev WHERE gorev_adi = 'Yazılım');
```

→ Ana sorgu personel tablosunda, ama alt sorgu bölüm adı üzerinden filtreleme yapıyor.

```
SELECT ortalama_maas FROM (  
SELECT AVG(maas) AS ortalama_maas  
FROM personel)  
AS maas_tablosu;
```

→ İçteki sorgu ortalama maaşı bulur, dıştaki sorgu bunu gösterir.



ÖRNEKLER

```
SELECT P_Adi,  
(SELECT gorev_adi FROM gorev  
WHERE gorev_id = personel.gorev_id)  
AS gorev FROM personel;
```

→ Her personelin bölümünü, alt sorguyla çekiyoruz.

```
SELECT * FROM personel  
WHERE maas > (SELECT AVG(maas) FROM personel);
```

→ Ortalama maaştan fazla maaşı olanları getirir.



ÖRNEKLER

```
SELECT * FROM personel p
WHERE EXISTS (
  SELECT * FROM izinler i
  WHERE i.personel_id = p.id );
```

Alt sorgunun sonuç üretip üretmediğini kontrol eder.
→ İzni olan personelleri listeler.

```
SELECT * FROM personel
WHERE sicil_no not
IN (select sicil_no from cezali_personeller)
```



ÖRNEK SORU

- 1) En yüksek maaşı alan personelin bilgilerini getiren sorguyu yazınız.**
- 2) Doğum yeri Samsun olan personellerin maaş ortalamasını getiren sorguyu yazınız.**



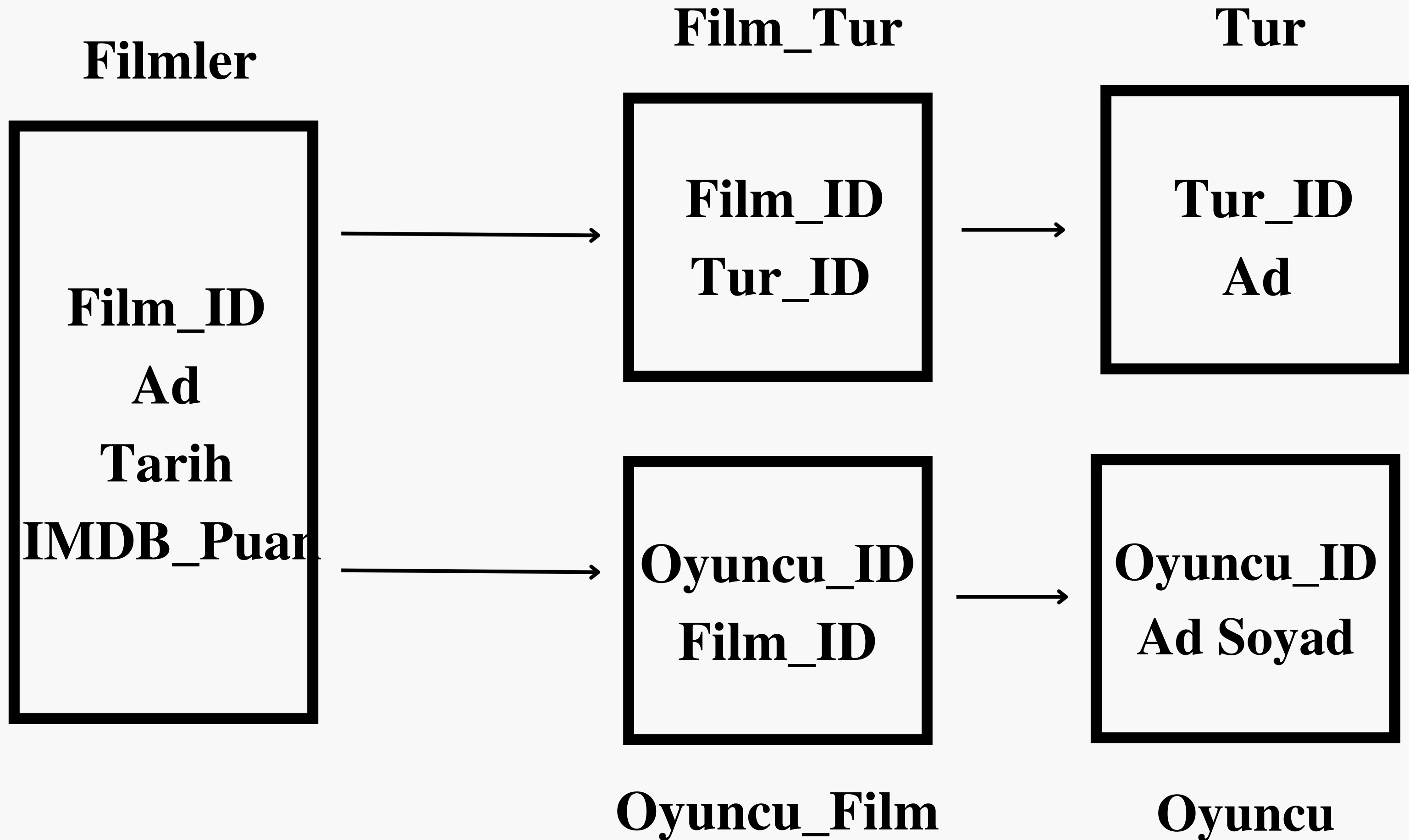
ÖRNEK SORU CEVAPLARI

1) En yüksek maaşı alan personelin bilgilerini getiren sorguyu yazınız.

```
SELECT * FROM personel  
WHERE maas = (SELECT MAX(maas) FROM personel);
```

2) Doğum yeri Samsun olan personellerin maaş ortalamasını getiren sorguyu yazınız.

```
SELECT AVG(maas) FROM personel  
WHERE dogum_yeri = 'Samsun';
```



ÖRNEK SORU



Nicolas Cage'nin oynadığı filmleri getirecek sorguyu yazınız.

```
SELECT Ad FROM Filmler
WHERE Film_ID in
(SELECT Film_ID From Oyuncu_Film
Where Oyuncu_ID=
(Select Oyuncu_ID From Oyuncular
Where Ad= 'Nicolas' AND Soyad= 'Cage'))
```



- Her bir alt select bir üsttekine değer gönderir.
- Alt sorgular parantez içine yazılır.
- Alt sorgularda da ORDER BY ifadesi kullanılmaz. Ancak ana sorgu içerisinde kullanılır.
- Alt sorgunun üstteki sorguya göndereceği değer tek de olabilir birden fazla da olabilir.
- Tek değer göndermesi için =, <, >, =<, => vb ifadeler olmalıdır.

NOT: Çok değer gönderen alt sorgularda ise ; IN, ANY, ALL gibi ifadeler olmalıdır.

KARŞILAŞTIRMA İFADELERİ



(NOT, OR, AND); Birden fazla koşula göre sıralama işlemlerinden bu ifadeler kullanılır.


İfade	Sembol
>	Büyük
<	Küçük
=	Eşit
>=	Büyük Eşit
<=	Küçük Eşit
<>	Eşit Değil

DISTINCT İFADESİ




Sorgu sonucunda tekrar eden kayıtları gizler. Sadece farklı (benzersiz) olanları gösterir.

SELECT sehir FROM personel;

 Sonuç:
sehir
Ankara
İstanbul
Ankara
İzmir

SELECT DISTINCT sehir FROM personel;

 Sonuç:
sehir
Ankara
İstanbul
İzmir

ORDER BY İFADESİ



ORDER BY, sorgu sonuçlarını belirli bir sütuna göre sıralar.

- Varsayılan olarak artan (ASC) sıralar.
- Azalan (DESC) sıralamak istersen belirtmen gerekir.

```
SELECT * FROM personel  
ORDER BY maas;  
SELECT * FROM personel  
ORDER BY maas DESC;
```

AD	SOYAD	ŞEHİR	MAAŞ
ALİ	AKSOY	ANKARA	18000
AYŞE	SOYLU	İSTANBUL	27000
EGE	DEMİR	BURSA	23000

ORDER BY İFADESİ



```
SELECT * FROM personel
ORDER BY maas;
```

AD	SOYAD	ŞEHİR	MAAŞ
ALİ	AKSOY	ANKARA	18000
EGE	DEMİR	BURSA	23000
AYŞE	SOYLU	İSTANBUL	27000

```
SELECT * FROM personel
ORDER BY maas DESC;
```

AD	SOYAD	ŞEHİR	MAAŞ
AYŞE	SOYLU	İSTANBUL	27000
EGE	DEMİR	BURSA	23000
ALİ	AKSOY	ANKARA	18000

WHERE İFADESİ



Tablo içerisinde bilgilerin bir koşula göre sıralanması için kullanılır.

SELECT * FROM tablo_adi WHERE koşul

```
SELECT * FROM personel WHERE sehir = 'Ankara';  
SELECT * FROM personel WHERE maas > 15000;  
SELECT * FROM personel WHERE cinsiyet = 'K';  
SELECT * FROM personel WHERE sehir != 'İstanbul';  
SELECT * FROM personel WHERE maas = 12000 AND cinsiyet = 'E';
```

NOT: != 'İstanbul' ifadesi İstanbul'da olmayanları listelemeyi sağlar.

BETWEEN İFADESİ



Belirli bir aralıkta olan değerleri seçmek için kullanılır.

Her iki sınır dahildir.

select * from tablo_adi where sütun_adi between 'tarih_1' and 'tarih_2'

```
SELECT * FROM personel WHERE maas  
BETWEEN 13000 AND 17000;
```

```
SELECT * FROM personel WHERE id BETWEEN 2 AND 5;
```

```
SELECT * FROM personel WHERE sehir = 'Ankara'  
AND maas BETWEEN 12000 AND 15000;
```

```
SELECT * FROM personel WHERE id NOT BETWEEN 1 AND 3;
```

LIKE İFADESİ



Belirli bir kalıba uyan verileri bulmak için kullanılır.

```
/*"A" harfiyle başlayan adlar*/  
SELECT * FROM personel WHERE ad LIKE 'A%';
```

```
/* "p" ile biten şehirler*/  
SELECT * FROM personel  
WHERE şehir LIKE '%p';
```

```
/*Ortasında "met" geçen adlar*/  
SELECT * FROM personel  
WHERE ad LIKE '%met%';
```

```
/*5 harfli adlar*/  
SELECT * FROM personel  
WHERE ad LIKE '_ _ _ _ _';
```

% → sıfır veya daha fazla karakter

_ → tek bir karakter



IN İFADESİ

Bir sütun değerinin birden fazla olasılıkla karşılaştırılması

```
/*Sadece Ankara ve İzmir'deki personeller*/  
SELECT * FROM personel  
WHERE sehir IN ('Ankara', 'İzmir');
```

```
/*Maaşı 12000, 14000 veya 16000 olanlar */  
SELECT * FROM personel  
WHERE maas IN (12000, 14000, 16000);
```

```
/*ID'si 1, 3, 5 olanlar*/  
SELECT * FROM personel WHERE id IN (1, 3, 5);
```

AS İFADESİ



Sorgu sonucunda sütunlara ad verme veya alt sorguya isim verme. Bu isim geçici bir isimdir tabloda herhangi bir değişikliğe yol açmaz.

```
/*Maaş ortalamasına isim verme*/  
SELECT AVG(maas) AS ortalama_maas  
FROM personel;
```

```
/*ad ve soyad'ı birleştirip tam ad olarak göster*/  
SELECT ad || ' ' || soyad  
AS tam_ad FROM personel;
```

```
/*Şehir'e takma ad ver*/  
SELECT şehir AS konum FROM personel;
```

```
/*Alt sorguya isim verme*/  
SELECT * FROM (SELECT * FROM personel  
WHERE şehir = 'Ankara')  
AS ankaralilar;
```

TARİH FONKSİYONLARI



Select getdate() : Anlık tarih ve saati verir. Select

datepart(YEAR,GETDATE()) as 'Yıl',

datepart(Month,GETDATE()) as 'Ay',

datepart(Day,GETDATE()) as 'Gün',

datepart(week,GETDATE()) as 'Hafta'

Bu ay içinde işe başlayanları bul:

```
SELECT * FROM personel
WHERE
MONTH(ise_giris_tarihi) = MONTH(CURDATE())
AND
YEAR(ise_giris_tarihi) = YEAR(CURDATE());
```

TARİH FONKSİYONLARI



```
/*Son 30 gün içinde alınan izinleri göster:*/  
SELECT * FROM izinler  
WHERE izin_tarihi BETWEEN DATE_SUB(CURDATE(), INTERVAL 30 DAY) AND CURDATE();  
  
/*Bugün doğum günü olan personeller:*/  
SELECT * FROM personel  
WHERE  
DAY(dogum_tarihi) = DAY(CURDATE())  
AND  
MONTH(dogum_tarihi) = MONTH(CURDATE());  
  
/*2023 yılında işe başlayanlar:*/  
SELECT * FROM personel  
WHERE YEAR(ise_giris_tarihi) = 2023;
```

DATEDIFF İFADESİ



İki tarih arasındaki farkı (gün) verir.

```
/*İki sabit tarih arasındaki fark*/
```

```
SELECT DATEDIFF('2025-12-31', '2025-01-01') AS gun_farki;
```

```
/*Personelin işe başlama süresine göre sıralama*/
```

```
SELECT ad, DATEDIFF(CURDATE(), ise_giris_tarihi)
```

```
AS kac_gun
```

```
FROM personel
```

```
ORDER BY kac_gun DESC;
```

```
/*30 günden önce işe başlayanlar*/
```

```
SELECT * FROM personel
```

```
WHERE DATEDIFF(CURDATE(), ise_giris_tarihi) > 30;
```