VERİ TABANI YÖNETİM SİSTEMLERİ

Yrd. Doç. Dr. Fırat YÜCEL
Akdeniz Üniversitesi
Enformatik Bölümü

Kompleks SQL Sorguları

Gruplama

İç İçe Sorgular

EXISTS, JOIN

Gruplama

- Bazen bir alana göre gruplandırıp, COUNT, SUM, MAX, MIN, AVG gibi fonksiyonların sonuçları gösterilmek istenebilir.

OGRENCILER

OgrNo	Adi	Soyadi

DERSLER

DersKodu	DersAdi

SINAVLAR

OgrNo	DersKodu	Notu	Donem

- Örneğin; SINAVLAR tablosunda her bir ders için not ortalamasını almak için,

```
SELECT DersAdi, AVG(Notu)
FROM DERSLER, SINAVLAR
WHERE DERSLER.DersKodu = SINAVLAR.DersKodu
GROUP BY DersAdi;
```

HAVING Deyimi

- Gruplama işleminde bir koşula bağlı olarak gruplama gerçekleştirilecekse, HAVING deyimi kullanılır.

- Örneğin; notu 60'tan yüksek en az 5 öğrenci bulunan dersler ve notu 60'tan yüksek olan öğrenci sayılarını bulan SQL sorgu deyimi,

```
SELECT DersAdi, COUNT(*)
FROM DERSLER, SINAVLAR
WHERE DERSLER.DersKodu = SINAVLAR.DersKodu AND Notu > 60
GROUP BY DersAdi
HAVING COUNT(*) >= 5;
```

Açık Değer Setleri ile Sorgulama

- Açık tanımlanan değer setleri kullanılarak sorgulama gerçekleştirilebilir.
- Örneğin; 1, 2 ve 3 numaralı öğrencilerin adı ve soyadı,

```
SELECT Adi, Soyadi
FROM OGRENCILER
WHERE OgrenciNo IN (1, 2, 3);
```

İç İçe Sorgular (*Nested Queries*)

- Kompleks sorgularda, bir sorgudan elde edilen değerler dizisi, diğer bir sorgudaki koşulda IN deyimi ile kullanılabilir.

- Örneğin, adı A ile başlayan öğrencilerin ENF101 kodlu dersten aldıkları notlar,

```
SELECT Adi, Soyadi, Notu
FROM OGRENCILER, SINAVLAR
WHERE OGRENCILER.OgrenciNo = SINAVLAR. OgrenciNo AND
DersKodu = 'ENF101' AND OgrenciNo IN
(       SELECT OgrenciNo
        FROM OGRENCILER
        WHERE Adi LIKE 'A%'
);
```

EXISTS Fonksiyonu

- EXITS fonksiyonu, bir iç içe sorguda, sorgu sonucunun boş olup olmadığını kontrol eder.
- Örneğin; notu girilen ders kodlarının listesi,

```
SELECT DISTINCT DERSLER.DersKodu
FROM DERSLER
WHERE EXISTS
(SELECT * FROM SINAVLAR
 WHERE SINAVLAR.DersKodu = DERSLER.DersKodu);
```

NOT EXISTS

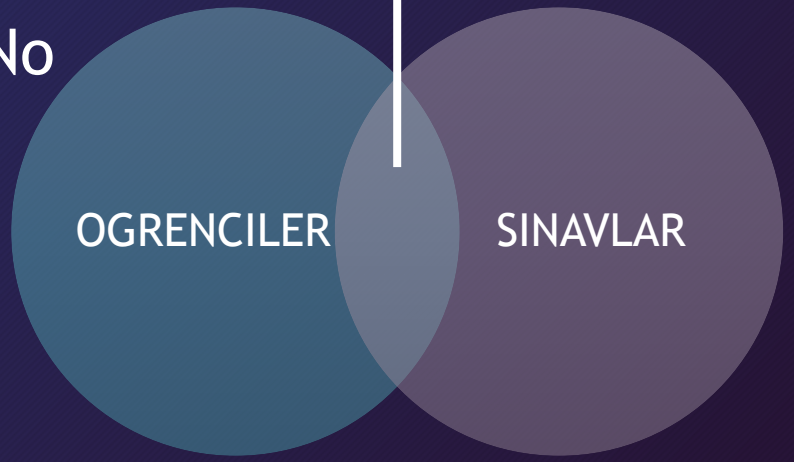
- Bir iç içe sorguda, sorgu sonucunun bulunmama durumunu kontrol eder.
- Örneğin; hiçbir notu girilmeyen ders kodlarının listesi,

```
SELECT DISTINCT DERSLER.DersKodu
FROM DERSLER
WHERE NOT EXISTS
(SELECT * FROM SINAVLAR
 WHERE SINAVLAR.DersKodu = DERSLER.DersKodu);
```

İç Birleşim (*Inner Join*)

- Tablolardaki ortak alanı eşleşen sonuçları döndüren SQL sorgusudur. Tabloların kesişim kümesini gösterir.
- Örneğin; ENF101 kodlu ders için öğrenci adı, soyadı ve notunu gösteren sorgu deyimi,

```
SELECT Adi, Soyadi, Notu
FROM OGRENCILER INNER JOIN SINAVLAR
ON OGRENCILER.OgrenciNo = SINAVLAR.OgrenciNo
WHERE DersKodu = 'ENF101';
```



Doğal Birleşim (*Natural Join*)

- INNER JOIN deyimi yerine yalnızca JOIN deyimi de yazılabilir. Bir önceki örnek için;

```
SELECT Adi, Soyadi, Notu
FROM OGRENCILER JOIN SINAVLAR
ON OGRENCILER.OgrenciNo = SINAVLAR.OgrenciNo
WHERE DersKodu = 'ENF101';
```

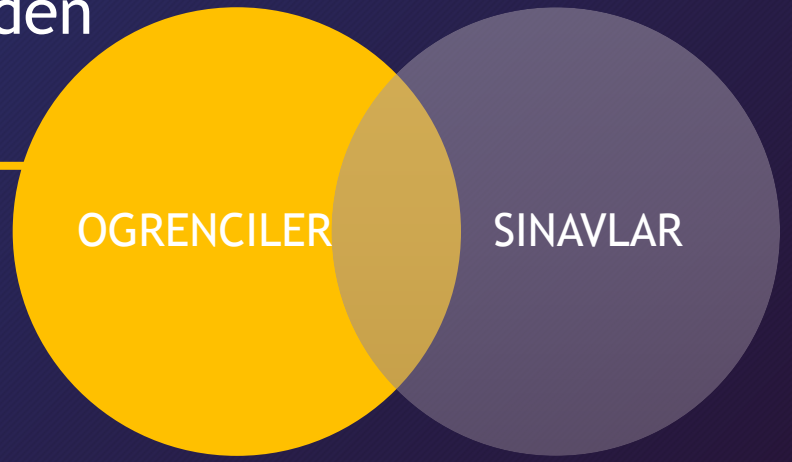
- **NATURAL JOIN**, her iki tablodaki ortak alanı otomatik olarak algılar ve ona göre ortak kayıtları görüntüler.
- Bir önceki örneği buna göre yaparsak;

```
SELECT Adi, Soyadi, Notu
FROM OGRENCILER NATURAL JOIN SINAVLAR
WHERE DersKodu = 'ENF101';
```

Sağ ve Sol Dış Birleşim (*RIGHT / LEFT OUTER JOIN*)

- Eğer bir tablodaki tüm kayıtlar ve diğer tablodaki koşula uyanlar döndürülmek istenirse sağ veya sol dış birleşim RIGHT JOIN ya da LEFT JOIN kullanılır.
- Örneğin; tüm öğrencilerin Adı ve Soyadı ve ENF101 dersinden aldıkları notları görüntülemek için,

```
SELECT Adi, Soyadi, Notu
FROM OGRENCILER LEFT OUTER JOIN SINAVLAR
ON OGRENCILER.OgrenciNo = SINAVLAR.OgrenciNo
 WHERE DersKodu = 'ENF101';
```



- **Dikkat: Burada dersi almayan öğrenciler de görüntülenecektir.**