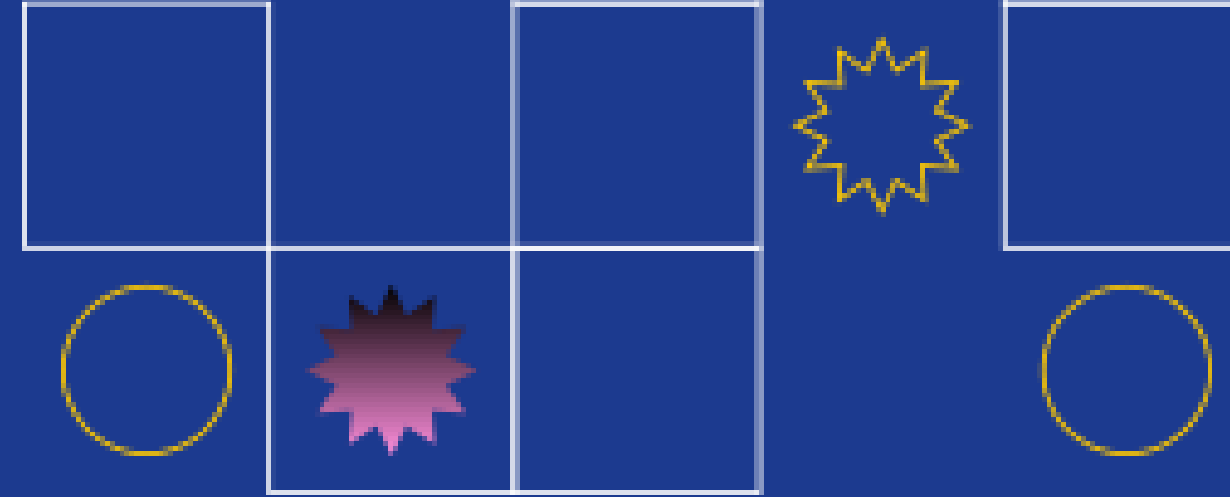


# Veri Tabanı Yönetim Sistemleri

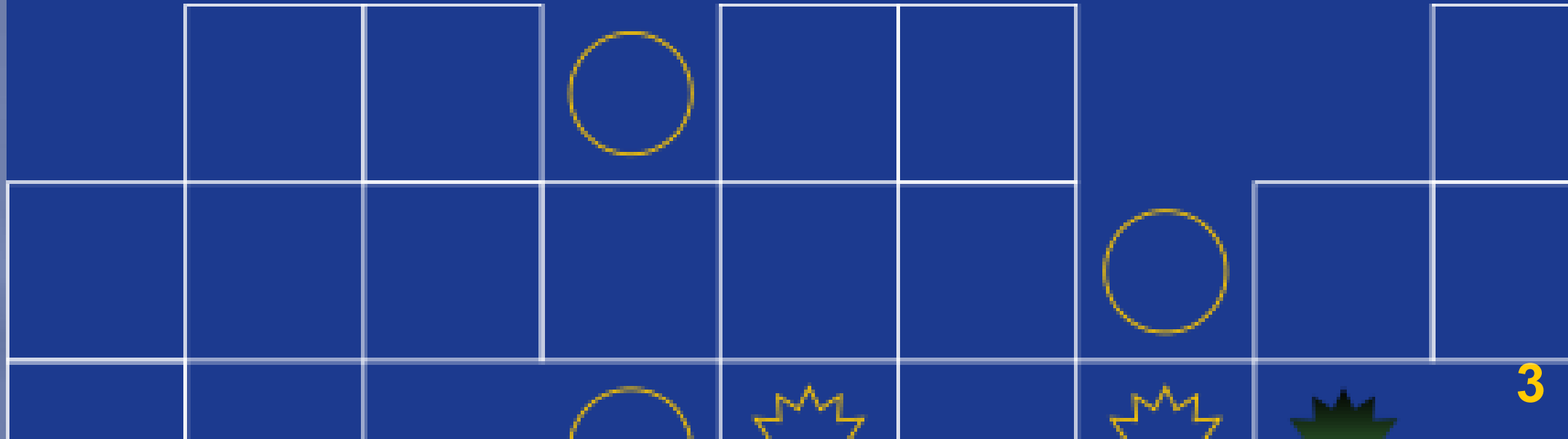
Öğr. Gör. Furkan DURMUŞ

# Kazanımlar

- 01** Önceki Hafta Tekrarı
- 02** LEFT JOIN İfadesi
- 03** RIGHT JOIN İfadesi
- 04** FULL OUTER JOIN
- 05** ALT SORGULAR



# Önceki Hafta Tekrarı



# GROUP BY Komutu

Kullanım şekli :

**SELECT**

    gruplama\_sutunu,

    TOPLAMSAL\_FONKSIYON(hesaplanacak\_sutun)

**FROM**

    tablo\_adi

**WHERE**

    koşul

**GROUP BY**

    gruplama\_sutunu\_1, gruplama\_sutunu\_2, ...

# GROUP BY Komutu

[Ogrenciler] tablosunda öğrencileri Cinsiyetlerine göre gruplandırarak toplam sayılarını bulunuz

```
1 SELECT Cinsiyet, COUNT(*) AS Sayi
2 FROM Ogrenciler GROUP BY Cinsiyet;
```

149 % No issues found

Results Messages

	Cinsiyet	Sayi
1	E	15
2	K	13

3	K	13
4	E	12

# GROUP BY Komutu

[Dersler] tablosunda her bir **OgretmenID**'nin üzerine tanımlı kaç farklı ders olduğunu listeleyin.

```
1 SELECT OgretmenID, COUNT(*) AS DersSayisi  
2 FROM Dersler GROUP BY OgretmenID
```

149 % No issues found

Results Messages

	OgretmenID	DersSayisi
1	1	2
2	2	2
3	3	2
4	4	1
5	5	2
6	6	2
7	7	1
8	8	1
9	9	2
10	10	1
11	11	1
12	12	1
13	13	1

13	13	1
15	15	1

# GROUP BY Komutu

[Notlar] tablosunda her bir **OgrenciNo** için sisteme girilmiş toplam not (Vize+Final+Ödev vb.) adedini bulun.

```
1 SELECT OgrenciNo, COUNT(*) AS ToplamSinavSayisi
2 FROM Notlar GROUP BY OgrenciNo;
```

149 % No issues found

Results Messages

	OgrenciNo	ToplamSinavSayisi
1	20231001	6
2	20231002	4
3	20231003	4
4	20231004	2
5	20231005	4
6	20231006	4
7	20231007	4
8	20231008	4
9	20231009	4
10	20231010	4
11	20231011	4
12	20231012	4
13	20231013	4
14	20231014	4
15	20231015	4
16	20231016	4
17	20231017	4
18	20231018	4
19	20231018	4
20	20231018	4
21	20231018	4
22	20231018	4

# GROUP BY Komutu

[Ogrenciler] tablosunda hem **Sinif** hem de **Cinsiyete** göre gruplandırma yaparak, örneğin "1. Sınıf Erkek" sayısını bulun.

```
1 SELECT Sinif, Cinsiyet, COUNT(*) AS Sayi
2 FROM Ogrenciler GROUP BY Sinif, Cinsiyet;
```

149 % No issues found

Results Messages

	Sinif	Cinsiyet	Sayi
1	1	E	9
2	2	E	6
3	1	K	6
4	2	K	7

# GROUP BY Komutu

[Dersler] tablosunda bölümlere göre ders çeşitliliğini görmek için her **BolumID**'deki ders sayısını bulun.

```
1  ✓ SELECT BolumID, COUNT(*) AS DersSayisi  
2  | FROM Dersler GROUP BY BolumID;
```

149 % No issues found

Results Messages

	BolumID	DersSayisi
1	1	6
2	2	4
3	3	4
4	4	2
5	5	1
6	6	1
7	7	1

# HAVING Komutu

Kullanım şekli :

**SELECT**

    gruplama\_sutunu,

    TOPLAMSAL\_FONKSIYON(hesaplanacak\_sutun)

**FROM**

    tablo\_adi

**WHERE**

    koşul

**GROUP BY**

    gruplama\_sutunu\_1, gruplama\_sutunu\_2, ...

**HAVING**

    gruplanmis\_veri\_kosullari

# HAVING Komutu

[Notlar] tablosunda, Final sınavlarının ortalaması 85'in altında olan dersleri listeleysin.

```
1 SELECT DersID, AVG(Puan) AS FinalOrt FROM Notlar  
2 WHERE SınavTuru = 'Final' GROUP BY DersID HAVING AVG(Puan) < 85;
```

149 % No issues found

Results Messages

	DersID	FinalOrt
1	2	76.3125
2	5	83
3	10	84
3	10	84
5	2	83
1	5	76.3125

# HAVING Komutu

[Ogrenciler] tablosunda, bünyesinde 6'ten fazla kız (K) öğrenci olan sınıfları (**Sinif**) bulun.

```
1 SELECT Sinif, COUNT(*) AS KizSayisi FROM Ogrenciler
2 WHERE Cinsiyet = 'K' GROUP BY Sinif HAVING COUNT(*) > 6;
```

149 % No issues found

Results Messages

	Sinif	KizSayisi
1	2	7

	Sinif	KizSayisi
1	2	7

# HAVING Komutu

[Notlar] tablosunda, alınan en yüksek puanı 95'ten fazla olan dersleri listeleysin.

```
1 SELECT DersID, MAX(Puan) AS EnYuksek  
2 FROM Notlar GROUP BY DersID HAVING MAX(Puan) > 95;
```

149 % No issues found

Results Messages

	DersID	EnYuksek
1	1	96
2	11	99

3	11	99
4	11	99

# HAVING Komutu

[Dersler] tablosunda, bir öğretmene (**OgretmenID**) atanmış toplam ders kredisi **5'ten fazla** ise o öğretmenleri listeleyin.

```
1  SELECT OgretmenID, SUM(Kredi) AS AtananKredi FROM Dersler
2  GROUP BY OgretmenID HAVING SUM(Kredi) > 5
3
```

149 % ✓ No issues found

Results Messages

	OgretmenID	AtananKredi
1	1	7
2	2	8
3	5	6
4	9	7

# INNER JOIN

## İfadesi

Kullanım şekli :

**SELECT**

    gruplama\_sutunu,

    TOPLAMSAL\_FONKSIYON(hesaplanacak\_sutun)

**FROM**

    TabloA

**INNER JOIN** TabloB

**ON** TabloA.OrtakSutun = TabloB.OrtakSutun;

**WHERE**

    koşul

Aynı anda birden fazla inner join kullanılabilir.

# INNER JOIN Komutu

[Notlar] ve [Ogrenciler] tablolarını birleştirerek, not alan öğrencinin Ad, Soyad bilgisini ve aldığı Puan'ı listeleyin.

```
1 SELECT Ad, Soyad, Puan FROM Notlar AS N
2 INNER JOIN Ogrenciler AS O ON N.OgrenciNo = O.OgrenciNo;
```

149 % No issues found

Results Messages

	Ad	Soyad	Puan
1	Ali	AKGÜN	85.5
2	Ali	AKGÜN	90
3	Ali	AKGÜN	78
4	Ali	AKGÜN	82.5
5	Ali	AKGÜN	88
6	Ali	AKGÜN	95
7	Ayşe	ATAKUL	92
8	Ayşe	ATAKUL	88.5
9	Ayşe	ATAKUL	75
10	Ayşe	ATAKUL	80
11	Can	BAYRAM	70
12	Can	BAYRAM	75.5
13	Can	BAYRAM	85
14	Can	BAYRAM	90
15	Can	BAYRAM	80
16	Can	BAYRAM	82
17	Can	BAYRAM	78.5
18	Can	BAYRAM	85

# INNER JOIN Komutu

[Dersler] ve [Bolumler] tablolarını birleştirerek, her dersin hangi BolumAdi'na ait olduğunu listeleyin.

```

1 SELECT D.DersAdi, B.BolumAdi FROM Dersler AS D
2 INNER JOIN Bolumler AS B ON D.BolumID = B.BolumID
3 ORDER BY BolumAdi

```

149 % No issues found

Results Messages

	DersAdi	BolumAdi
1	Veritabanı Yönetim Sistemleri	Arka Yüz Yazılım Geliştirme
2	Algoritma ve Programlama	Arka Yüz Yazılım Geliştirme
3	Bilgisayar Ağları	Arka Yüz Yazılım Geliştirme
4	Temel Programlama 1	Arka Yüz Yazılım Geliştirme
5	Bulut Bilişim	Arka Yüz Yazılım Geliştirme
6	Nesne Yönelimli Programlama	Arka Yüz Yazılım Geliştirme
7	Dinamik	Havacılık
8	Güzel Ahlak	İlahiyat
9	Makine Elemanları	Makine Resim ve Konstrüksiyonu
10	Teknik Resim	Makine Resim ve Konstrüksiyonu
11	Web Programlama	Ön Yüz Yazılım Geliştirme
12	Kullanıcı Arayüzü Tasarımı	Ön Yüz Yazılım Geliştirme
13	Web Tasarım	Ön Yüz Yazılım Geliştirme
14	Mobil Uygulama Geliştirme	Ön Yüz Yazılım Geliştirme
15	İşletme Yönetimi	Sivil Havacılık Kabin Hizmetleri
16	İngilizce	Sivil Havacılık Kabin Hizmetleri
17	Havacılık Güvenliği	Sivil Havacılık Kabin Hizmetleri
18	Müşteri İlişkileri	Sivil Havacılık Kabin Hizmetleri
19	Nesne Tabanlı Programlama	Yazılım Mühendisliği

# INNER JOIN Komutu

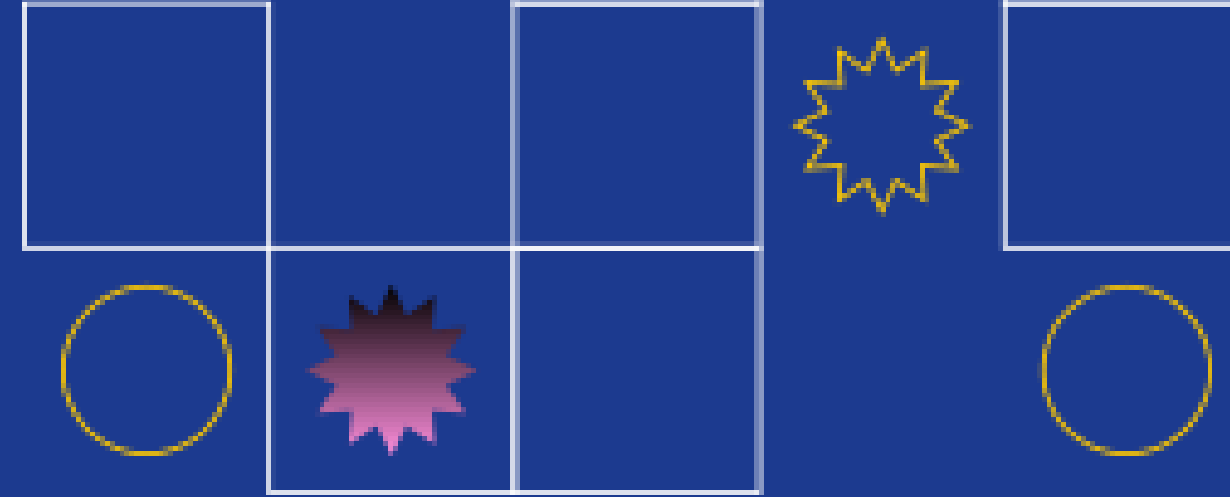
[Ogretmenler] tablosunda BolumID'si 1 olan öğretmenlerin adlarını ve bölüm adlarını listeleyin.

```
1 SELECT T.Ad, B.BolumAdi FROM Ogretmenler AS T
2 INNER JOIN Bolumler AS B ON T.BolumID = B.BolumID WHERE T.BolumID = 1;
```

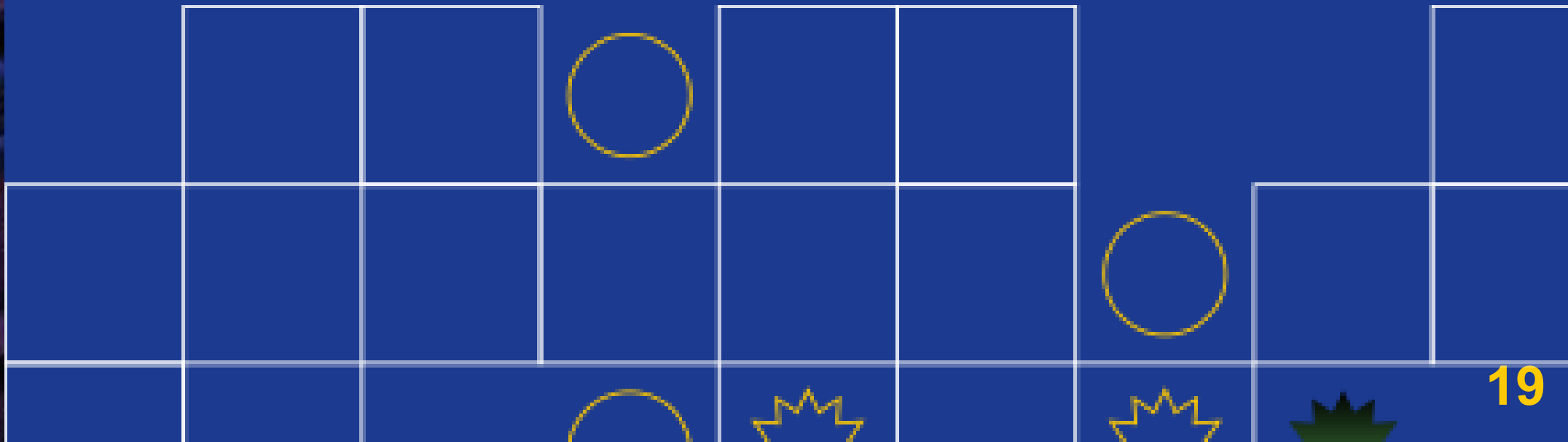
149 % No issues found

Results Messages

	Ad	BolumAdi
1	Furkan	Arka Yüz Yazılım Geliştirme
2	Kemal	Arka Yüz Yazılım Geliştirme
3	Mehmet	Arka Yüz Yazılım Geliştirme



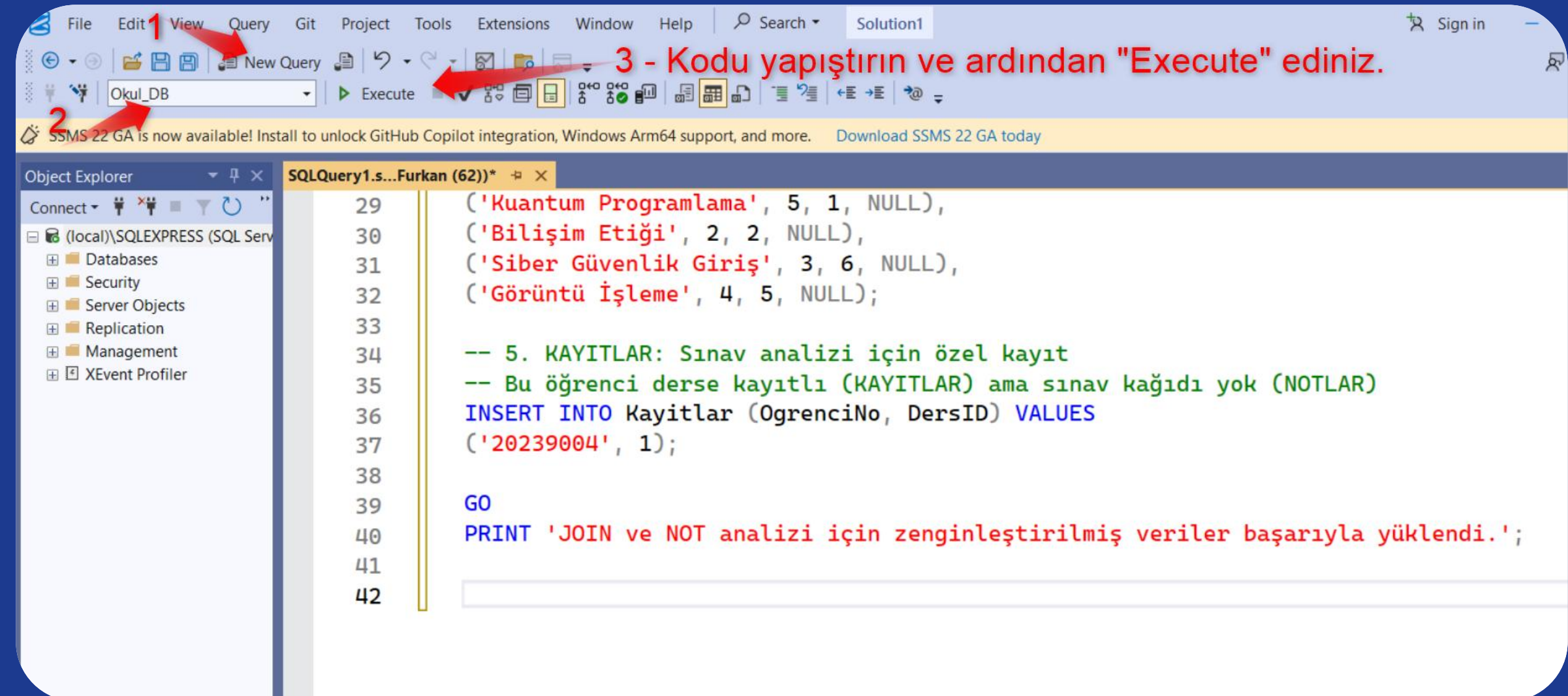
# LEFT JOIN İfadesi





# LEFT JOIN İfadesi

SSMS içerisinde yeni bir sorgu oluşturunuz, okul\_db'yi seçiniz ve kodu yapıştırıp execute ediniz.



3 - Kodu yapıştırın ve ardından "Execute" ediniz.

```
29 ('Kuantum Programlama', 5, 1, NULL),
30 ('Bilişim Etiği', 2, 2, NULL),
31 ('Siber Güvenlik Giriş', 3, 6, NULL),
32 ('Görüntü İşleme', 4, 5, NULL);
33
34 -- 5. KAYITLAR: Sınav analizi için özel kayıt
35 -- Bu öğrenci derse kayıtlı (KAYITLAR) ama sınav kağıdı yok (NOTLAR)
36 INSERT INTO Kayitlar (OgrenciNo, DersID) VALUES
37 ('20239004', 1);
38
39 GO
40 PRINT 'JOIN ve NOT analizi için zenginleştirilmiş veriler başarıyla yüklendi.';
41
42
```

# LEFT JOIN İfadesi

**LEFT JOIN**, bir SQL sorgusunda soldaki (ilk yazılan) tablonun tüm kayıtlarını getirir. Sağdaki (ikinci yazılan) tabloda bu kayıtlara karşılık gelen bir veri varsa onları yan yana koyar. Eğer sağdaki tabloda bir karşılık yoksa, o sütunlar için sonuç kümesine NULL değerini yazar.

Basitçe ifade etmek gerekirse:

"Soldaki tablo benim için önceliklidir; sağda karşılığı olsun ya da olmasın hepsini listele."

# LEFT JOIN İfadesi

## Dikkat Edilmesi Gereken Hususlar

Hangi Tablo Sol Hangisi Sağ?: FROM ifadesinden hemen sonra yazdığınız tablo "Sol", LEFT JOIN ifadesinden sonra yazdığınız tablo "Sağ" tablodur. Tabloların yerini değiştirirseniz sonuç tamamen değişir.

Analiz Gücü: Veritabanındaki eksiklikleri bulmak için birebirdir. Örneğin; "Derse kayıtlı ama notu girilmemiş öğrenciler" veya "Öğretmeni olmayan dersler" gibi durumları LEFT JOIN ve WHERE ... IS NULL kombinasyonu ile bulabilirsiniz.

# LEFT JOIN İfadesi

Kullanım şekli :

**SELECT**

TabloA.kolon1

TabloB.kolon2

**FROM**

TabloA

**LEFT JOIN** TabloB

**ON** TabloA.OrtakSutun = TabloB.OrtakSutun;

**WHERE**

koşul

Aynı anda birden fazla join kullanılabilir.

# LEFT JOIN İfadesi

[Dersler] tablosundaki tüm dersleri ve varsa öğretmenlerini listeleyin.  
(Öğretmeni olmayan dersler NULL gelecek.)

```
1 SELECT D.DersAdi, O.Ad, O.Soyad
2 FROM Dersler AS D
3 LEFT JOIN Ogretmenler AS O
4 ON D.OgretmenID = O.OgretmenID;
```

	DersAdi	Ad	Soyad
0	İngilizce	Sümeyye	ÖZER
1	Makine Elemanları	Tuba	YILDIZ KAPLAN
2	Teknik Resim	Ali Samet	AYVAZ
3	Havacılık Güvenliği	Serdar	ÜNVER
4	Müşteri İlişkileri	Sümeyye	ÖZER
5	Nesne Yönelimli Programlama	Mehmet	YILMAZ
6	Mobil Uygulama Geliştirme	Ayşe	KAYA
7	Nesne Tabanlı Programlama	Ahmet ...	DURSUN
8	Güzel Ahlak	Tuncay	ALTUN
9	Dinamik	Ahmet	YILDIRIM
20	Kuantum Programlama	NULL	NULL
21	Bilişim Etiği	NULL	NULL
22	Siber Güvenlik Giriş	NULL	NULL
	Görüntü İşleme	NULL	NULL

# LEFT JOIN İfadesi

[Ogretmenler] tablosundaki tüm öğretmenleri ve verdikleri dersleri listeleyin.

```
1 SELECT O.Ad, O.Soyad, D.DersAdi
2 FROM Ogretmenler AS O
3 LEFT JOIN Dersler AS D
4 ON O.OgretmenID = D.OgretmenID;
```

149 % No issues found

Results Messages

	Ad	Soyad	DersAdi
10	Sümeyye	ÖZER	İngilizce
11	Sümeyye	ÖZER	Müşteri İlişkileri
12	Tuba	YILDIZ KAPLAN	Makine Elemanları
13	Ali Samet	AYVAZ	Teknik Resim
14	Mehmet	YILMAZ	Bulut Bilişim
15	Mehmet	YILMAZ	Nesne Yönelimli Programlama
16	Ayşe	KAYA	Mobil Uygulama Geliştirme
17	Ahmet	YILDIRIM	Dinamik
18	Tuncay	ALTUN	Güzel Ahlak
19	Ahmet ...	DURSUN	Nesne Tabanlı Programlama
20	Hakan	Hocaoğlu	NULL
21	Meltem	Bilgin	NULL
22	Deniz	Yılmaz	NULL
23	Kemal	Sönmez	NULL

# LEFT JOIN İfadesi

[Ogrenciler] ve [Kayitlar] tablolarını kullanarak, tüm öğrencileri ve kayıt tarihlerini listeleyin.

```
SQLQuery1.s...Furkan (65))*  
1 SELECT O.Ad, O.Soyad, K.KayitTarihi  
2 FROM Ogrenciler AS O  
3 LEFT JOIN Kayitlar AS K  
4 ON O.OgrenciNo = K.OgrenciNo;
```

149 % No issues found

Results Messages

	Ad	Soyad	KayitTarihi
1	Ali	Koç	NULL
2	Zeynep	Arslan	NULL
3	Mert	Yıldız	NULL
4	Ali	AKGÜN	2025-11-05 19:01:58.380
5	Ali	AKGÜN	2025-11-05 19:01:58.380
6	Ali	AKGÜN	2025-11-05 19:01:58.380
7	Ayşe	ATAKUL	2025-11-05 19:01:58.380
8	Ayşe	ATAKUL	2025-11-05 19:01:58.380

## LEFT JOIN Komutu

[Kayıtlar] ve [Notlar] tablolarını birleştirerek; derse kayıtlı her öğrencinin numarasını, aldığı sınav puanını ve türünü listeleyin.

# LEFT JOIN Komutu

[Kayitlar] ve [Notlar] tablolarını birleştirerek; derse kayıtlı her öğrencinin numarasını, aldığı sınav puanını ve türünü listeleyin.

```

1 SELECT K.OgrenciNo, N.Puan, N.SinavTuru
2 FROM Kayitlar AS K
3 LEFT JOIN Notlar AS N
4 ON K.OgrenciNo = N.OgrenciNo AND K.DersID = N.DersID;

```

149 % No issues found

Results Messages

	OgrenciNo	Puan	SinavTuru
107	20231023	94	Final
108	20231023	82	Vize
109	20231023	86	Final
110	20231023	NULL	NULL
111	20231024	91	Vize
112	20231024	94	Final
113	20231024	85	Vize
114	20231024	89	Final
115	20231025	86	Vize
116	20231025	90	Final
117	20231025	83	Vize
118	20231025	88	Odev
119	20231025	NULL	NULL
120	20239004	NULL	NULL

Query executed successfully

(local) SQL EXPLORER (16.0 BIT) | DESKTOP-GES81HE0 | End

## LEFT JOIN Komutu

[Bolumler] ve [Ogrenciler] tablolarını birleştirerek; tüm bölümlerin adını ve o bölümlere kayıtlı öğrencilerin ad ve soyadlarını listeleyin.

# LEFT JOIN Komutu

[Bolumler] ve [Ogrenciler] tablolarını birleştirerek; tüm bölümlerin adını ve o bölümlere kayıtlı öğrencilerin ad ve soyadlarını listeleyin.

```

1 SELECT B.BolumAdi, O.Ad, O.Soyad
2 FROM Bolumler AS B
3 LEFT JOIN Ogrenciler AS O
4 ON B.BolumID = O.BolumID;

```

149 % No issues found

Results Messages

	BolumAdi	Ad	Soyad
26	Sivil Havacılık Kabin Hizm...	Gizem	KURT
27	Sivil Havacılık Kabin Hizm...	Tolga	ÇAKIR
28	Makine Resim ve Konstrü...	Zeynep	ÇETİN
29	Makine Resim ve Konstrü...	Oğuz	POLAT
30	Makine Resim ve Konstrü...	Deniz	ACAR
31	Makine Resim ve Konstrü...	Ebru	ÖZER
32	Havacılık	Alperen	Güneş
33	İlahiyat	NULL	NULL
34	Yazılım Mühendisliği	NULL	NULL
35	Yapay Zeka Mühendisliği	NULL	NULL
36	Veri Bilimi	NULL	NULL
37	Uzay Bilimleri	NULL	NULL
38	Gömülü Sistemler	NULL	NULL
39	Endüstriyel Tasarım	NULL	NULL

Query executed successfully. (local)\SQLEXP

## LEFT JOIN Komutu

[Dersler] ve [Ogretmenler] tablolarını birleřtirerek; hiçbir öđretmeni olmayan derslerin isimlerini ve kredilerini listeleyin.

# LEFT JOIN Komutu

[Dersler] ve [Ogretmenler] tablolarını birleştirerek; hiçbir öğretmeni olmayan derslerin isimlerini ve kredilerini listeleyin.

```
SQLQuery1.s...Furkan (65))* # X
1 SELECT D.DersAdi, D.Kredi
2 FROM Dersler AS D
3 LEFT JOIN Ogretmenler AS O
4 ON D.OgretmenID = O.OgretmenID
5 WHERE O.OgretmenID IS NULL;
```

135 % No issues found

Results Messages

	DersAdi	Kredi
1	Kuantum Programlama	5
2	Bilişim Etiği	2
3	Siber Güvenlik Giriş	3
4	Görüntü İşleme	4

## LEFT JOIN Komutu

[Bolumler] ve [Dersler] tablolarını birleştirerek; tüm bölümlerin adını ve o bölümlerde açılmış olan derslerin isimlerini listeleyin.

# LEFT JOIN Komutu

[Bolumler] ve [Dersler] tablolarını birleştirerek; tüm bölümlerin adını ve o bölümlerde açılmış olan derslerin isimlerini listeleyin.

```
SQLQuery1.s...Furkan (65)* x
1 SELECT B.BolumAdi, D.DersAdi
2 FROM Bolumler AS B
3 LEFT JOIN Dersler AS D
4 ON B.BolumID = D.BolumID;
```

135 % No issues found

Results Messages

	BolumAdi	DersAdi
16	Sivil Havacılık Kabin Hizmetleri	Müşteri İlişkileri
17	Makine Resim ve Konstrüks...	Makine Elemanları
18	Makine Resim ve Konstrüks...	Teknik Resim
19	Havacılık	Dinamik
20	Havacılık	Görüntü İşleme
21	İlahiyat	Güzel Ahlak
22	İlahiyat	Siber Güvenlik Giriş
23	Yazılım Mühendisliği	Nesne Tabanlı Programlama
24	Yapay Zeka Mühendisliği	NULL
25	Veri Bilimi	NULL
26	Uzay Bilimleri	NULL
27	Gömülü Sistemler	NULL
28	Endüstriyel Tasarım	NULL

Query executed successfully (local) SOLEXP

## LEFT JOIN Komutu

[Kayıtlar] ve [Notlar] tablolarını birleştirerek; derse kaydı olduğu halde henüz hiçbir sınav notu girilmemiş öğrencileri listeleyin.

# LEFT JOIN Komutu

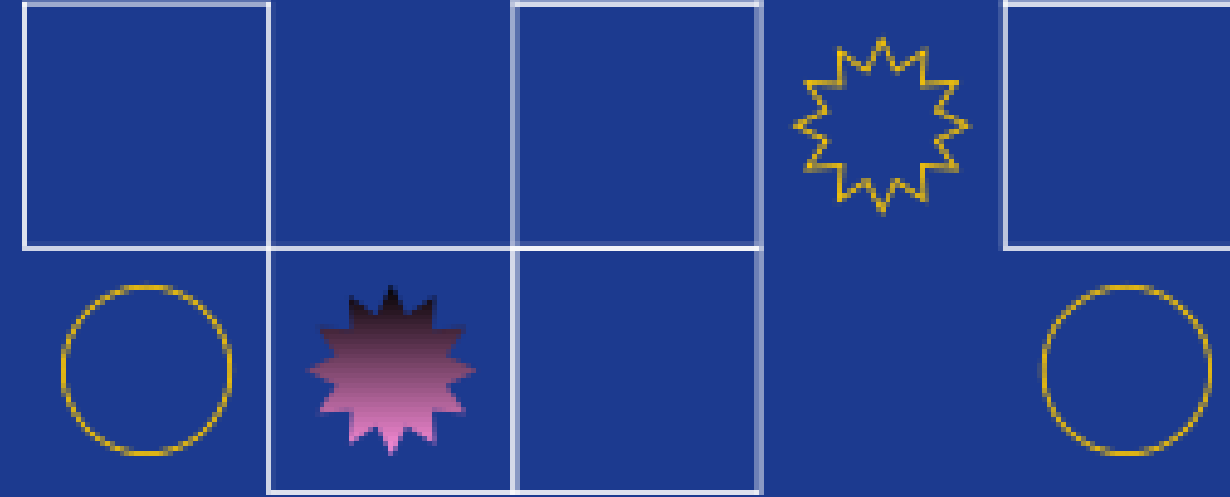
[Kayitlar] ve [Notlar] tablolarını birleştirerek; derse kaydı olduğu halde henüz hiçbir sınav notu girilmemiş öğrencileri listeleyin.

```
1 SELECT K.OgrenciNo
2 FROM Kayitlar AS K
3 LEFT JOIN Notlar AS N
4 ON K.OgrenciNo = N.OgrenciNo WHERE N.NotID IS NULL;
```

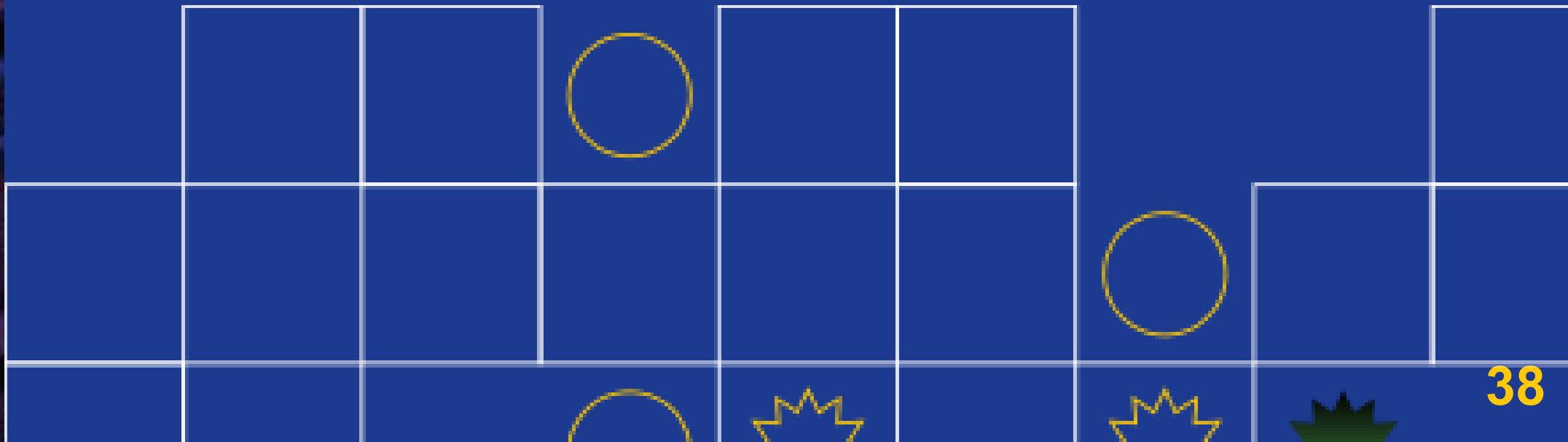
135 % No issues found

Results Messages

	OgrenciNo
1	20239004



# RIGHT JOIN İfadesi



# RIGHT JOIN

## İfadesi

**RIGHT JOIN**, sorguda ikinci sırada yazılan (SAĞ) tablodaki tüm kayıtları korur. Birinci (SOL) tabloda bu kayıtlara karşılık gelen bir veri varsa birleştirir; eğer sol tabloda bir karşılık yoksa, sol tablodan gelen sütunlar için sonuç kümesine NULL yazar.

# RIGHT JOIN İfadesi

## Dikkat Edilmesi Gereken Hususlar

**Pratik Kullanım:** Yazılım dünyasında genellikle tablolar soldan sağa okunduğu için LEFT JOIN daha sık kullanılır. Herhangi bir RIGHT JOIN sorgusu, tabloların yerini değiştirerek LEFT JOIN ile de yazılabilir.

**NULL Değerler:** Sol tablodan gelen sütunlarda NULL değerleri gördüğümüzde, bu sağdaki kaydın solda bir eşleşmesi olmadığını gösterir.

# RIGHT JOIN İfadesi

Kullanım şekli :

**SELECT**

TabloA.kolon1

TabloB.kolon2

**FROM**

TabloA

**RIGHT JOIN** TabloB

**ON** TabloA.OrtakSutun = TabloB.OrtakSutun;

**WHERE**

koşul

Aynı anda birden fazla join kullanılabilir.

# LEFT JOIN İfadesi

[Ogretmenler] ve [Bolumler] tablolarını birleştirerek; tüm bölümleri ve o bölümlerde (varsa) görevli öğretmenleri listeleyin.

```

1  SELECT B.BolumAdi, O.Ad, O.Soyad
2  FROM Ogretmenler AS O
3  RIGHT JOIN Bolumler AS B
4  ON O.BolumID = B.BolumID;

```

135 % No issues found

Results Messages

	BolumAdi	Ad	Soyad
9	Sivil Havacılık Kabin Hizmetleri	Serdar	ÜNVER
10	Sivil Havacılık Kabin Hizmetleri	Sümeyye	ÖZER
11	Makine Resim ve Konstrüksiyonu	Tuba	YILDIZ KAPLAN
12	Makine Resim ve Konstrüksiyonu	Ali Samet	AYVAZ
13	Havacılık	Ahmet	YILDIRIM
14	Havacılık	Deniz	Yılmaz
15	İlahiyat	Tuncay	ALTUN
16	Yazılım Mühendisliği	Ahmet ...	DURSUN
17	Yapay Zeka Mühendisliği	NULL	NULL
18	Veri Bilimi	Kemal	Sönmez
19	Uzay Bilimleri	NULL	NULL
20	Gömülü Sistemler	NULL	NULL
21	Endüstriyel Tasarım	NULL	NULL

Successfully executed

## RIGHT JOIN Komutu

[Dersler] ve [Bolumler] tablolarını birleştirerek hiç dersi olmayan bölümleri bulun ve **Bölüm Adına göre** sıralayın.

# RIGHT JOIN Komutu

[Dersler] ve [Bolumler] tablolarını birleştirerek hiç dersi olmayan bölümleri bulun ve **Bölüm Adına göre** sıralayın.

```
1 SELECT B.BolumAdi
2 FROM Dersler AS D
3 RIGHT JOIN Bolumler AS B
4 ON D.BolumID = B.BolumID
5 WHERE D.DersID IS NULL
6 ORDER BY B.BolumAdi;
```

135 % No issues found

Results Messages

	BolumAdi
1	Endüstriyel Tasarım
2	Gömülü Sistemler
3	Uzay Bilimleri
4	Veri Bilimi
5	Yapay Zeka Mühendisliği

## RIGHT JOIN Komutu

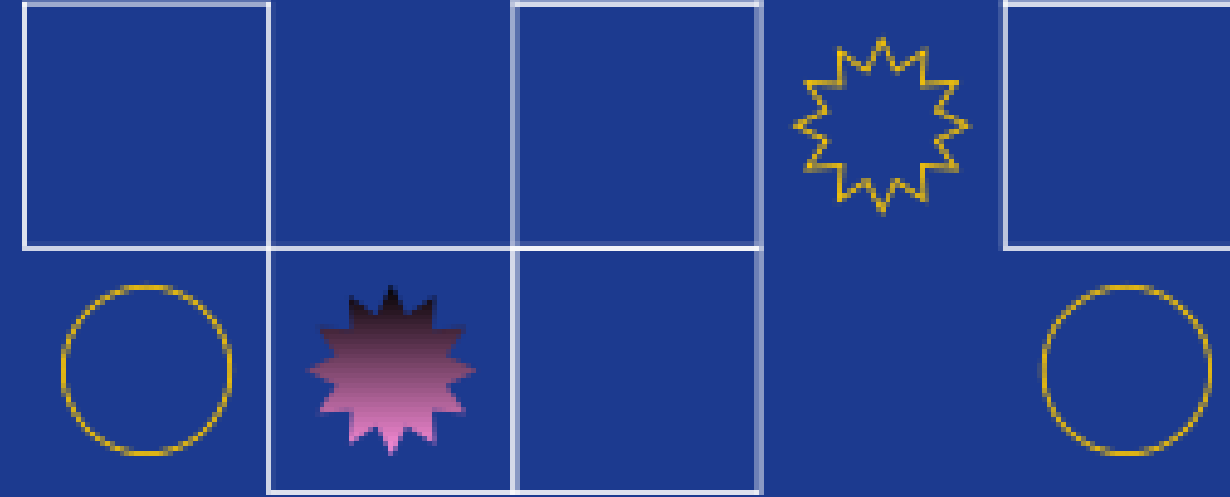
[Kayitlar] ve [Dersler] tablolarını birleştirerek tüm dersleri ve kayıt tarihlerini listeleyin. Kaydı olmayan dersleri en sonda görmek için **Tarihe göre azalan** sıralayın.

# RIGHT JOIN Komutu

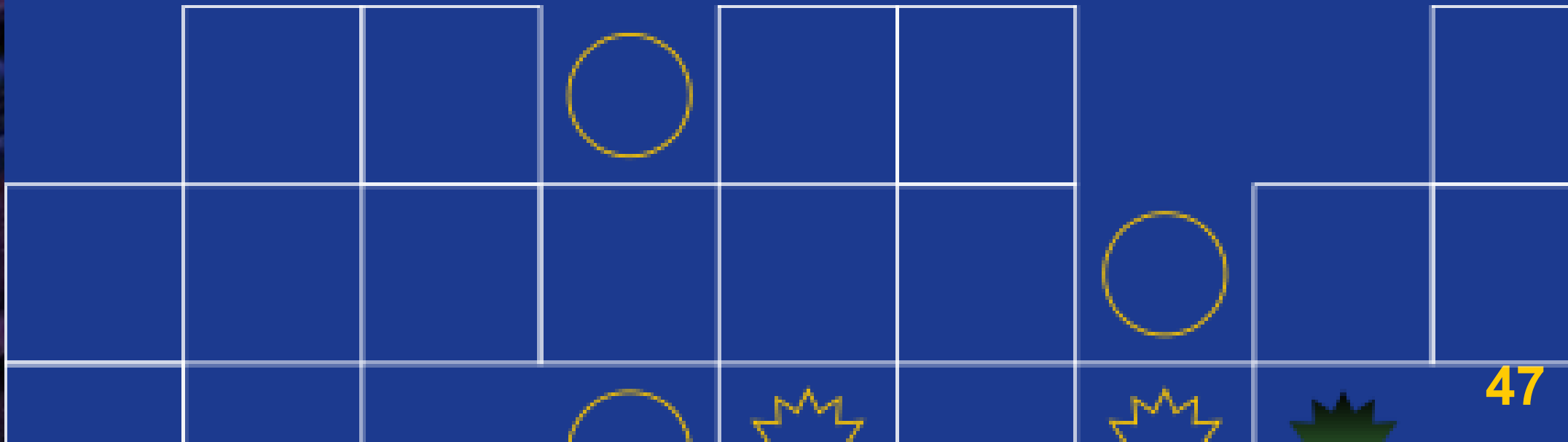
[Kayıtlar] ve [Dersler] tablolarını birleştirerek tüm dersleri ve kayıt tarihlerini listeleyin. Kaydı olmayan dersleri en sonda görmek için **Tarihe göre azalan** sıralayın.

```
SQLQuery1.s...Furkan (65)*
1 SELECT D.DersAdi, K.KayitTarihi
2 FROM Kayitlar AS K
3 RIGHT JOIN Dersler AS D
4 ON K.DersID = D.DersID
5 ORDER BY K.KayitTarihi DESC;
```

	DersAdi	KayitTarihi
66	Veritabanı Yönetim Sistemleri	2025-11-05 19:01:58.380
67	Veritabanı Yönetim Sistemleri	2025-11-05 19:01:58.380
68	Veritabanı Yönetim Sistemleri	2025-11-05 19:01:58.380
69	Veritabanı Yönetim Sistemleri	2025-11-05 19:01:58.380
70	Veritabanı Yönetim Sistemleri	2025-11-05 19:01:58.380
71	Veritabanı Yönetim Sistemleri	2025-11-05 19:01:58.380
72	Nesne Tabanlı Programlama	NULL
73	Güzel Ahlak	NULL
74	Dinamik	NULL
75	Kuantum Programlama	NULL
76	Bilişim Etiği	NULL
77	Siber Güvenlik Giriş	NULL
78	Görüntü İşleme	NULL



# FULL OUTER JOIN İfadesi



# FULL OUTER JOIN

## İfadesi

**FULL OUTER JOIN**, birleştirilen her iki tablodaki tüm kayıtları sonuç kümesine dahil eder.

Tablolar arasında eşleşen kayıtlar yan yana getirilir.

- \* Sol tabloda olup sağda karşılığı olmayan kayıtlar için sağ taraf NULL döner.
- \* Sağ tabloda olup solda karşılığı olmayan kayıtlar için sol taraf NULL döner.

# FULL OUTER JOIN

## İfadesi

### Dikkat Edilmesi Gereken Hususlar

**Veri Denetimi:** Genellikle veri tabanındaki kopuklukları (hem öğretmeni olmayan dersleri hem de dersi olmayan öğretmenleri aynı anda görmek gibi) tespit etmek için kullanılır.

**Performans:** Diğer JOIN türlerine göre daha fazla kaynak tüketebilir çünkü her iki tarafın da taranması ve boşlukların NULL ile doldurulması gerekir.

**Matematiksel Mantık:** Sol ve Sağ tabloların birleşimi (union) gibi düşünülebilir.

# FULL OUTER JOIN

## İfadesi

Kullanım şekli :

**SELECT**

TabloA.kolon1

TabloB.kolon2

**FROM**

TabloA

**FULL OUTER JOIN** TabloB

**ON** TabloA.OrtakSutun = TabloB.OrtakSutun;

**WHERE**

koşul

Aynı anda birden fazla join kullanılabilir.

## FULL OUTER JOIN Komutu

[Ogretmenler] ve [Dersler] tablolarını birleştirerek; tüm öğretmenleri ve tüm dersleri tek listede görün.

# FULL OUTER JOIN Komutu

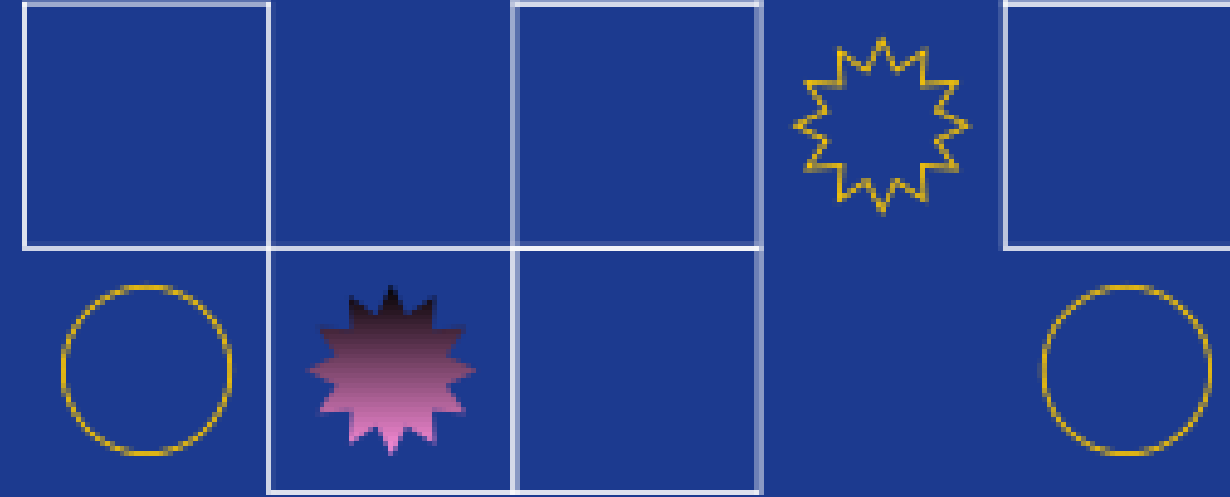
[Ogretmenler] ve [Dersler] tablolarını birleştirerek; tüm öğretmenleri ve tüm dersleri tek listede görün.

```

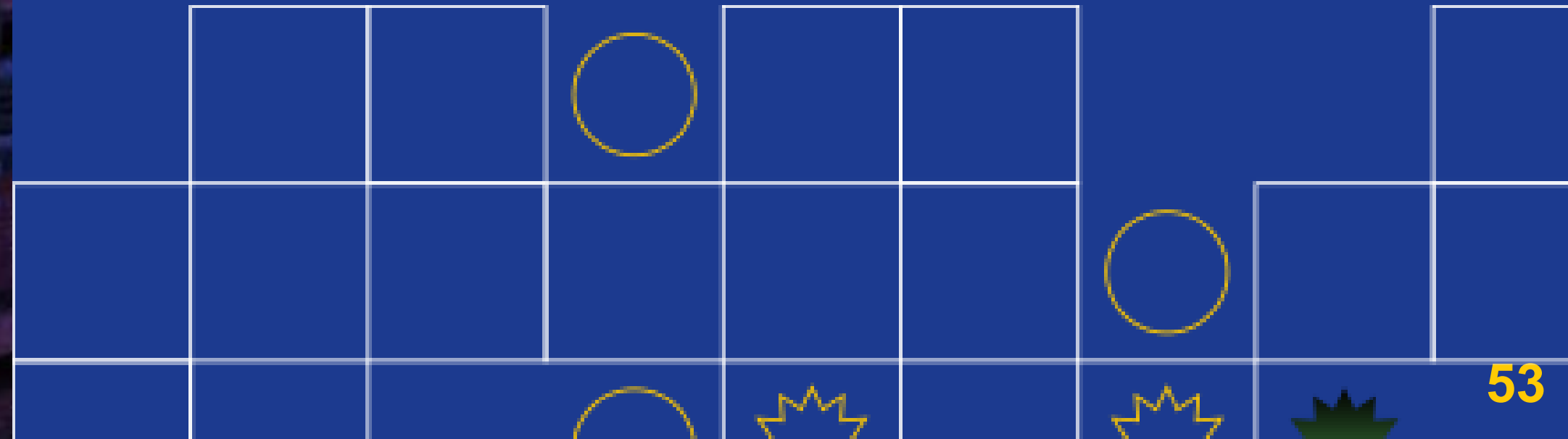
SQLQuery1.s...Furkan (65))*
1 SELECT O.Ad, O.Soyad, D.DersAdi
2 FROM Ogretmenler AS O
3 FULL OUTER JOIN Dersler AS D
4 ON O.OgretmenID = D.OgretmenID ORDER BY O.Ad;

```

	Ad	Soyad	DersAdi
1	NULL	NULL	Kuantum Programlama
2	NULL	NULL	Bilişim Etiği
3	NULL	NULL	Siber Güvenlik Giriş
4	NULL	NULL	Görüntü İşleme
5	Ahmet	YILDIRIM	Dinamik
6	Ahmet Faruk	DURSUN	Nesne Tabanlı Programlama
7	Ali Samet	AYVAZ	Teknik Resim
8	Ayşe	KAYA	Mobil Uygulama Geliştirme
9	Deniz	Yılmaz	NULL
10	Ferhat	PERÇİN	Kullanıcı Arayüzü Tasarımı
11	Furkan	DURMUŞ	Veritabanı Yönetim Sistemleri
12	Furkan	DURMUŞ	Bilgisayar Ağları
13	Hakan	Hocaoğlu	NULL



# Alt Sorgular



# Alt Sorgular (WHERE)

Ana sorgudaki **WHERE** koşulunun içine yerleştirilen ve ana sorguya bir "referans değeri" üreten sorgulardır. Genellikle tek bir değer (Scalar) döndüren alt sorgular için kullanılır. Dış sorgu, bu içten gelen değere göre satırları filtreler.

# Alt Sorgular (WHERE)

## Dikkat Edilmesi Gereken Hususlar

**Tek Değer Zorunluluğu:** Eğer =, >, < gibi operatörler kullanıyorsanız, alt sorgu yalnızca tek bir satır ve tek bir sütun döndürmelidir. Aksi halde hata alırsınız.

**Parantez Kullanımı:** Alt sorgu her zaman parantez ( ) içinde yazılmalıdır.

**Boş Sonuç:** Eğer alt sorgu hiçbir değer döndürmezse (NULL), ana sorgu da genellikle hiçbir sonuç döndürmez.

# Alt Sorgular (WHERE)

Kullanım şekli :

**SELECT**

kolonlar

**FROM**

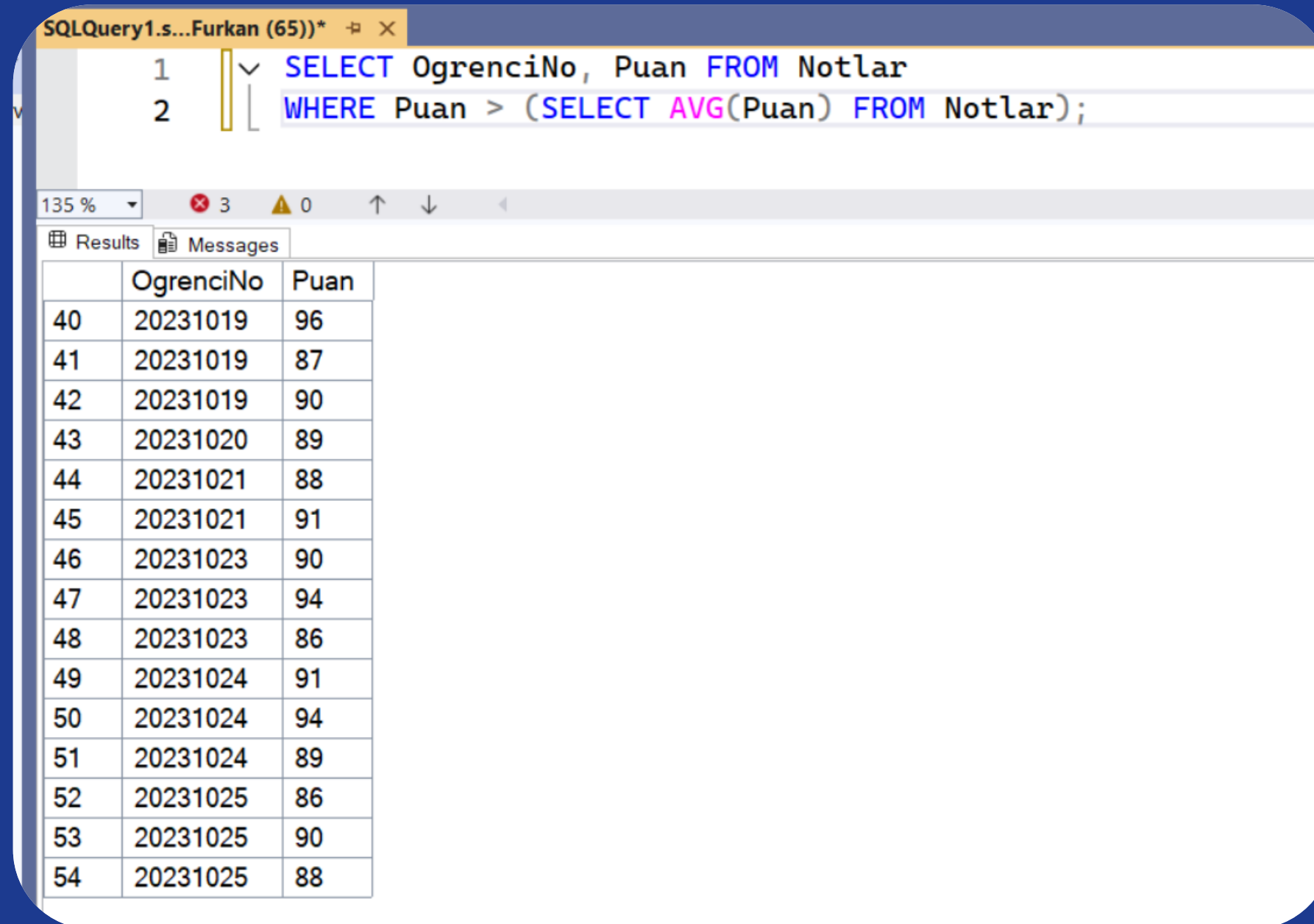
TabloA

**WHERE**

sutun\_adi [Operatör] (SELECT tek\_sutun FROM Tablo2 WHERE kosul);

# Alt Sorgular (WHERE)

Sınav ortalamasının üzerinde puan alan öğrencilerin numaralarını ve puanlarını listeleyin.



The screenshot shows a SQL query window with the following text:

```
1 SELECT ÖğrenciNo, Puan FROM Notlar
2 WHERE Puan > (SELECT AVG(Puan) FROM Notlar);
```

The results window displays the following data:

	OğrenciNo	Puan
40	20231019	96
41	20231019	87
42	20231019	90
43	20231020	89
44	20231021	88
45	20231021	91
46	20231023	90
47	20231023	94
48	20231023	86
49	20231024	91
50	20231024	94
51	20231024	89
52	20231025	86
53	20231025	90
54	20231025	88

# Alt Sorgular (WHERE)

Kredisi en yüksek olan dersin adını ve kredisini listeleyin.

```
SQLQuery1.s...Furkan (65))*  
1 SELECT DersAdi, Kredi FROM Dersler  
2 WHERE Kredi = (SELECT MAX(Kredi) FROM Dersler);
```

135 %

3

0

↑

↓

←

Results

Messages

	DersAdi	Kredi
1	Güzel Ahlak	5
2	Kuantum Programlama	5

## Alt Sorgular (WHERE)

En düşük notu alan öğrencinin numarasını ve aldığı puanı listeleysin.

## Alt Sorgular (WHERE)

En düşük notu alan öğrencinin numarasını ve aldığı puanı listeleysin.

```
SQLQuery1.s...Furkan (65))*  X
1  SELECT OgrenciNo, Puan FROM Notlar
2  WHERE Puan = (SELECT MIN(Puan) FROM Notlar);
```

135 % 3 0

Results Messages

	OgrenciNo	Puan
1	20231022	55

## Alt Sorgular (WHERE)

'Matematik' dersinden alınan en yüksek puana sahip kaydı listeleyin.

## Alt Sorgular (WHERE)

'Matematik' dersinden alınan en yüksek puana sahip kaydı listeleyin.

```
SQLQuery1.s...Furkan (65))*  ×
1  SELECT * FROM Notlar
2  WHERE Puan = (SELECT MAX(Puan) FROM Notlar WHERE DersID = 1);
```

135 % 3 0

Results Messages

	NotID	OgrenciNo	DersID	SinavTuru	Puan
1	65	20231017	11	Vize	96
2	74	20231019	1	Final	96

## Alt Sorgular (WHERE)

'Furkan Durmuş' isimli öğretmenin verdiği dersleri listeleyin.

# Alt Sorgular (WHERE)

'Furkan Durmuş' isimli öğretmenin verdiği dersleri listeleyin.

```
SQLQuery1.s...Furkan (65))*  X
1  SELECT DersAdi FROM Dersler
2  WHERE OgretmenID = (SELECT OgretmenID FROM Ogretmenler
3  WHERE Ad = 'Furkan' AND Soyad = 'Durmuş');
```

135 % 3 0

Results Messages

	DersAdi
1	Veritabanı Yönetim Sistemleri
2	Bilgisayar Ağları

# Alt Sorgular (IN / NOT IN)

**IN:** Dış sorgudaki bir değer, alt sorgudan dönen liste içerisinde olup olmadığını kontrol eder. Eğer değer listede varsa o satırı getirir.

**NOT IN:** Değerin, alt sorgudan dönen liste içerisinde olmamasını şart koşar. Listede bulunmayan kayıtları ayıklamak için kullanılır.

# Alt Sorgular (IN / NOT IN)

## Dikkat Edilmesi Gereken Hususlar

**Sütun Sayısı:** Alt sorgu (SELECT kısmı) mutlaka tek bir sütun döndürmelidir. Birden fazla sütun döndürürseniz "subquery has too many columns" hatası alırsınız.

**NULL Tehlikesi (Kritik):** Eğer NOT IN kullandığınız alt sorgunun sonuç listesinde tek bir tane bile NULL değer varsa, tüm sorgu boş dönebilir. Bu yüzden alt sorguda WHERE sütun IS NOT NULL kullanmak güvenlidir.

# Alt Sorgular (IN / NOT IN)

Kullanım şekli :

**SELECT**

kolonlar

**FROM**

TabloA

**WHERE**

sutun\_adi **[NOT] IN** (SELECT ortak\_sutun FROM Tablo2 WHERE kosul);

# Alt Sorgular (IN / NOT IN)

[Kayıtlar] tablosunda kaydı bulunan (derse yazılmış) öğrencilerin isimlerini listeleyin.

```
SQLQuery1.s...Furkan (65)* -> X
1  SELECT Ad, Soyad FROM Ogrenciler
2  WHERE OgrenciNo IN (SELECT OgrenciNo FROM Kayitlar);
```

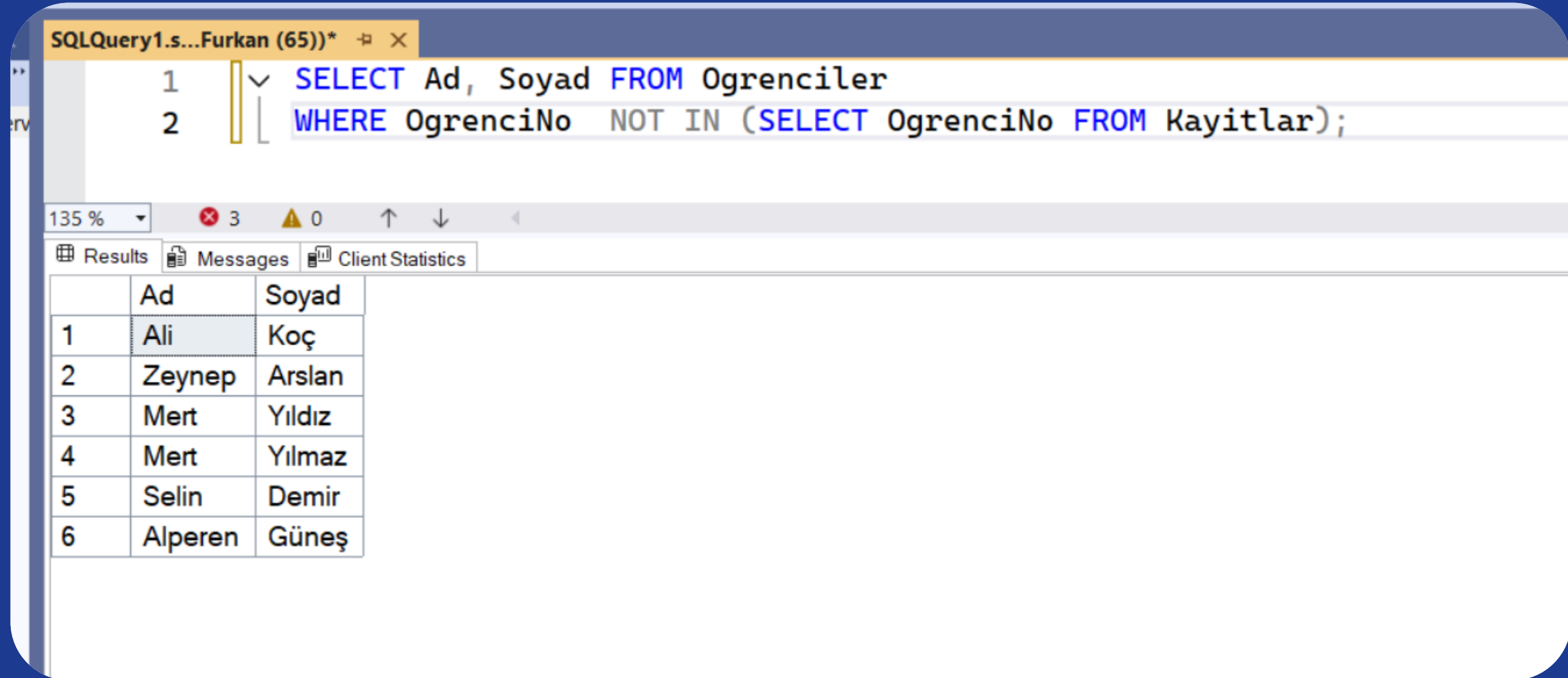
135 % 3 0

Results Messages

	Ad	Soyad
12	Fatma	ŞEN
13	Mehmet	YILDIZ
14	Esra	AKIN
15	Hakan	ÇOBAN
16	Gizem	KURT
17	Oğuz	POLAT
18	Deniz	ACAR
19	Cem	ÖZKAN
20	Seda	GÜNEŞ
21	Barış	YAVUZ
22	Pınar	ERDO...
23	Tolga	ÇAKIR
24	Ebru	ÖZER
25	Serkan	ASLAN
26	Buse	Kaya

# Alt Sorgular (IN / NOT IN)

Hiçbir derse kayıt yaptırmamış (boştaki) öğrencileri listeleyin.

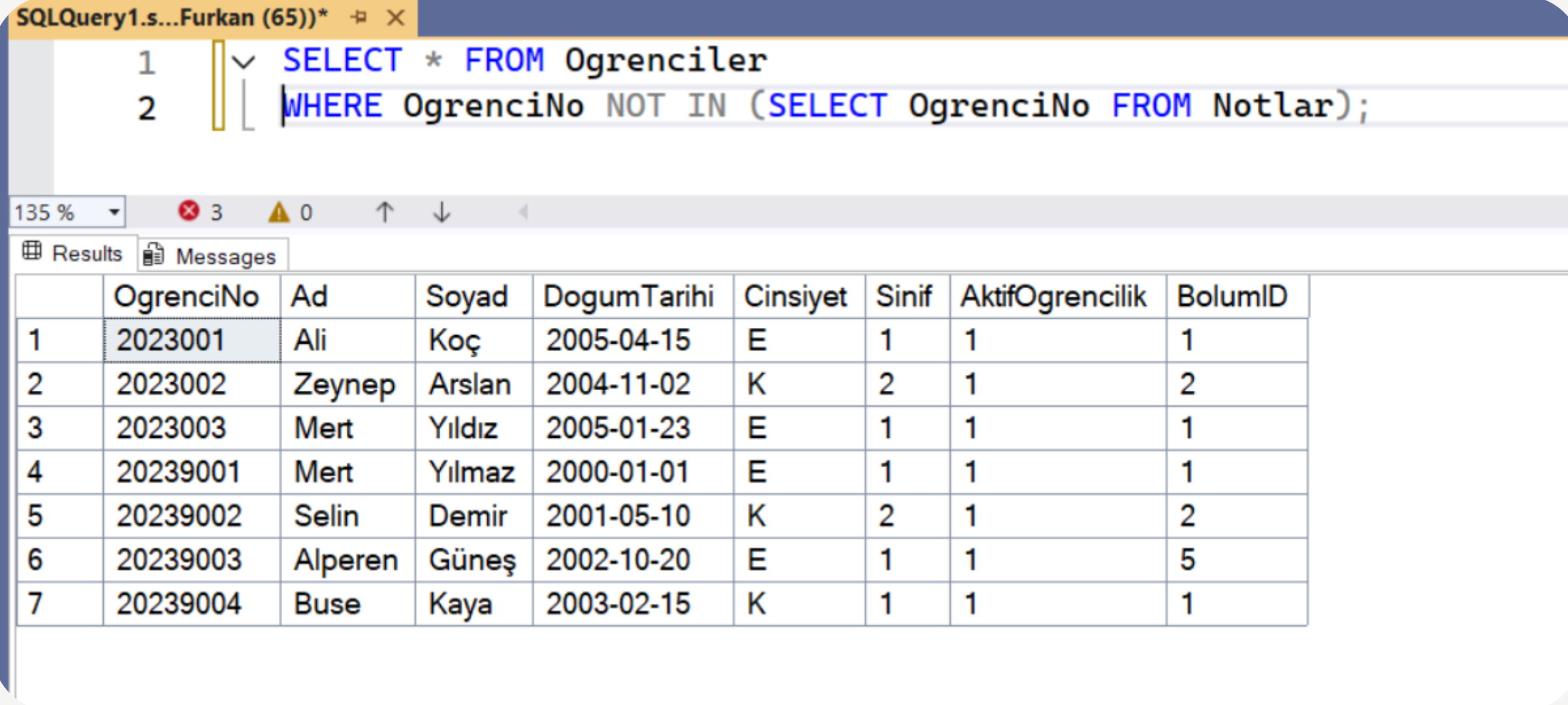


```
1 SELECT Ad, Soyad FROM Ogrenciler
2 WHERE OgrenciNo NOT IN (SELECT OgrenciNo FROM Kayitlar);
```

	Ad	Soyad
1	Ali	Koç
2	Zeynep	Arslan
3	Mert	Yıldız
4	Mert	Yılmaz
5	Selin	Demir
6	Alperen	Güneş

## Alt Sorgular (IN / NOT IN)

Henüz **hiçbir sınav notu girilmemiş** olan öğrencilerin bilgilerini getirin.



The screenshot shows a SQL query editor window titled "SQLQuery1.s...Furkan (65))". The query is as follows:

```
1 SELECT * FROM Ogrenciler
2 WHERE OgrenciNo NOT IN (SELECT OgrenciNo FROM Notlar);
```

The query results are displayed in a table with the following columns: OgrenciNo, Ad, Soyad, DogumTarihi, Cinsiyet, Sinif, AktifOgrencilik, and BolumID. The results are as follows:

	OgrenciNo	Ad	Soyad	DogumTarihi	Cinsiyet	Sinif	AktifOgrencilik	BolumID
1	2023001	Ali	Koç	2005-04-15	E	1	1	1
2	2023002	Zeynep	Arslan	2004-11-02	K	2	1	2
3	2023003	Mert	Yıldız	2005-01-23	E	1	1	1
4	20239001	Mert	Yılmaz	2000-01-01	E	1	1	1
5	20239002	Selin	Demir	2001-05-10	K	2	1	2
6	20239003	Alperen	Güneş	2002-10-20	E	1	1	5
7	20239004	Buse	Kaya	2003-02-15	K	1	1	1

Diğer Alt Sorgu Çeşitleri :

SELECT İfadesinde Alt Sorgular (Hesaplanmış Kolon Oluşturma)

FROM İfadesinde Alt Sorgular (Derived Tables / Türetilmiş Tablolar)

HAVING İfadesinde Alt Sorgular (Grup Verisi Filtreleme)

UNION - EXCEPT Kullanımı

EXISTS ve NOT EXISTS Kullanımı

NE ÖĞRENEBİLİRİM ?

STORE PROCEDURES

TRIGGERS

INDEXES

TRANSACTIONS

