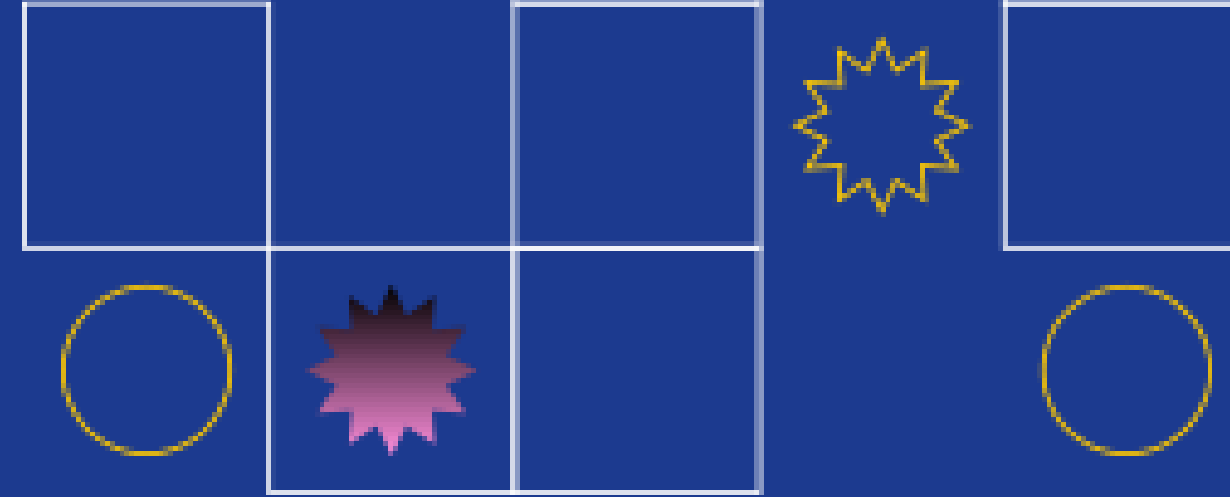


Veri Tabanı Yönetim Sistemleri

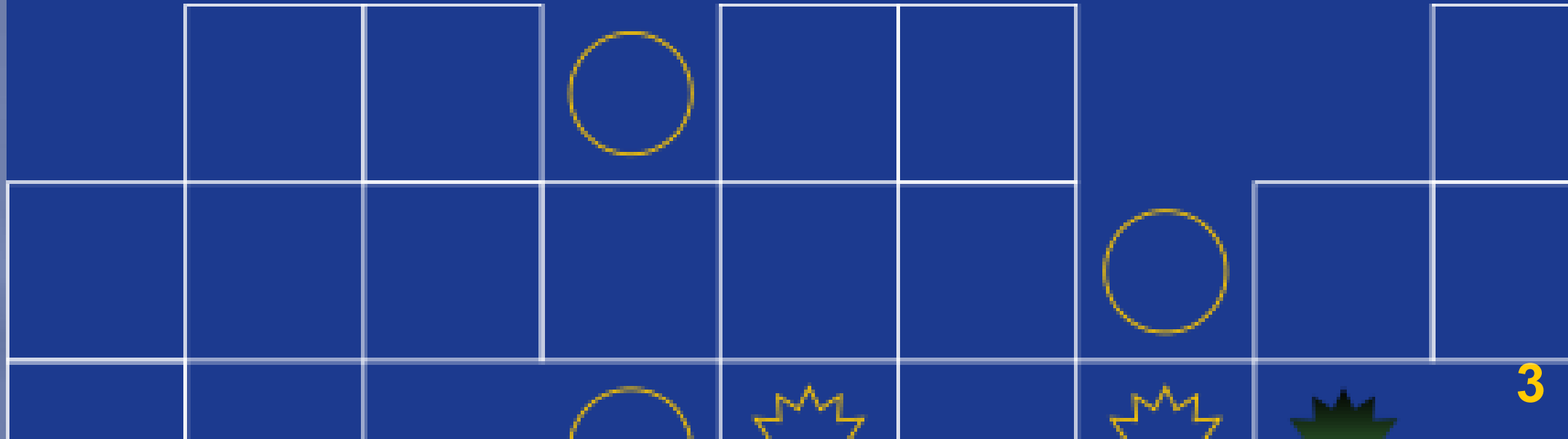
Öğr. Gör. Furkan DURMUŞ

Kazanımlar

- 01** GROUP BY Komutu
- 02** HAVING Komutu
- 03** INNER JOIN İfadesi



Önceki Hafta Tekrarı



SUM (Toplama) Fonksiyonu

Kullanım şekli :

```
SELECT  
    SUM(sutun_adi)  
FROM  
    tablo_adi  
WHERE  
    koşul;
```

SUM (Toplama) Fonksiyonu

[Notlar] tablosunda 20231004 numaralı öğrencinin aldığı tüm notların toplamını bulunuz.

```
1 SELECT SUM(Puan) AS OgrToplamNot
2 FROM Notlar WHERE OgrenciNo = '20231004';
```

149 %

✓ No issues found

Results Messages

	OgrToplamNot
1	125

↓	125
	OgrToplamNot

SUM (Toplama) Fonksiyonu

[Notlar] tablosunda Puanı 60 ile 80 arasında (dahil) olan notların toplam puanını hesaplayınız.

```
1  SELECT SUM(Puan) AS Puan60_80Toplam FROM Notlar  
2  WHERE Puan BETWEEN 60 AND 80;
```

149 % No issues found

Results Messages

	Puan60_80Toplam
1	1651.5

SUM (Toplama) Fonksiyonu

[Notlar] tablosunda DersId'si 1 olan dersin Vize - Ödev sınav türlerine ait tüm puanların toplamını hesaplayınız.

```
1  SELECT SUM(Puan) AS VizeOdevToplamPuan FROM Notlar
2  WHERE SinavTuru IN ('Vize', 'Odev') and DersID = 1;
```

149 % No issues found

Results Messages

	VizeOdevToplamPuan
1	777.5

AVG (Ortalama) Fonksiyonu

Kullanım şekli :

```
SELECT
    AVG(sutun_adi)
FROM
    tablo_adi
WHERE
    koşul;
```

AVG (Ortalama) Fonksiyonu

OgrenciNo'su 2003 ile başlayan öğrencilerin aldığı notların ortalama puanını hesaplayınız.

```
1 SELECT AVG(Puan) AS OgrNo2023Ortalama FROM Notlar
2 WHERE OgrenciNo LIKE '2023%';
```

149 % No issues found

Results Messages

	OgrNo2023Ortalama
1	85

↓	82
---	----

AVG (Ortalama) Fonksiyonu

Teknik Resim dersinin tüm notlarının ortalamasını bulunuz.

```
1 SELECT AVG(Puan) AS TeknikResimOrtalama  
2 FROM Notlar WHERE DersID = 12;
```

149 %

✓ No issues found

Results Messages

	TeknikResimOrtalama
1	86.5

↓	86.5
---	------

AVG (Ortalama) Fonksiyonu

Veri Tabanı Yönetim sistemi dersinin Final notlarının ortalama puanını hesaplayınız.

```
1 SELECT AVG(Puan) AS Ders1FinalOrtalamasi FROM Notlar
2 WHERE DersID = 1 AND SinavTuru = 'Final';
```

149 % No issues found

Results Messages

	Ders1FinalOrtalamasi
1	89.1666666666667

MAX (En Büyük) Fonksiyonu

Kullanım şekli :

```
SELECT  
    MAX(sutun_adi)  
FROM  
    tablo_adi  
WHERE  
    koşul;
```

MAX (En Büyük) Fonksiyonu

İşletme Yönetimi dersinin Vize'sinde alınan en yüksek puanı bulunuz.

```
1 SELECT MAX(Puan) AS EnYuksekVizePuanı  
2 FROM Notlar WHERE SinavTuru = 'Vize' and DersID = 6
```

149 % No issues found

Results Messages

	EnYuksekVizePuanı
1	92

↓ 85

MAX (En Büyük) Fonksiyonu

Vize veya Final sınav türüne ait en yüksek puanı bulunuz.

```
1 SELECT MAX(Puan) AS MaxVizeFinalPuan FROM Notlar
2 WHERE SinavTuru IN ('Vize', 'Final');
```

149 %

✖ 1

⚠ 0

↑

↓

◀

Results Messages

	MaxVizeFinalPuan
1	99

aa

MIN (En Küçük) Fonksiyonu

Kullanım şekli :

```
SELECT  
    MIN(sutun_adi)  
FROM  
    tablo_adi  
WHERE  
    koşul;
```

MIN (En Küçük) Fonksiyonu

Algoritma ve Programlama dersi için verilen en düşük puanı bulunuz.

```
1  SELECT MIN(Puan) AS Ders2EnDusukPuan  
2  FROM Notlar WHERE DersID = 3
```

149 %



1



0



Results

Messages

	Ders2EnDusukPuan
1	75



12

MIN (En Küçük) Fonksiyonu

1. Sınıf Kız öğrencileri arasındaki en erken (en yaşlı) doğum tarihini bulunuz.

```
1  SELECT MIN(DogumTarihi) AS Sinif1KizEnYasliDT FROM Ogrenciler
2  WHERE Sinif = '1' AND Cinsiyet = 'K'
```

149 % 1 0

Results Messages

	Sinif1KizEnYasliDT
1	2003-04-12

COUNT (Sayı) Fonksiyonu

Kullanım şekli :

```
SELECT  
    COUNT(sutun_adi | *)  
FROM  
    tablo_adi  
WHERE  
    koşul;
```

COUNT (Sayı) Fonksiyonu

1. Sınıf Erkek öğrenci sayısını bulunuz.

```
1  SELECT COUNT(*) AS Sinif1ErkekAdedi FROM Ogrenciler  
2  WHERE Sinif = '1' AND Cinsiyet = 'E';
```

149 % No issues found

Results Messages

	Sinif1ErkekAdedi
1	9

↓	0
---	---

COUNT (Sayı) Fonksiyonu

Dersler tablosunda kaç farklı kredi miktarı kullanıldığını sayınız.

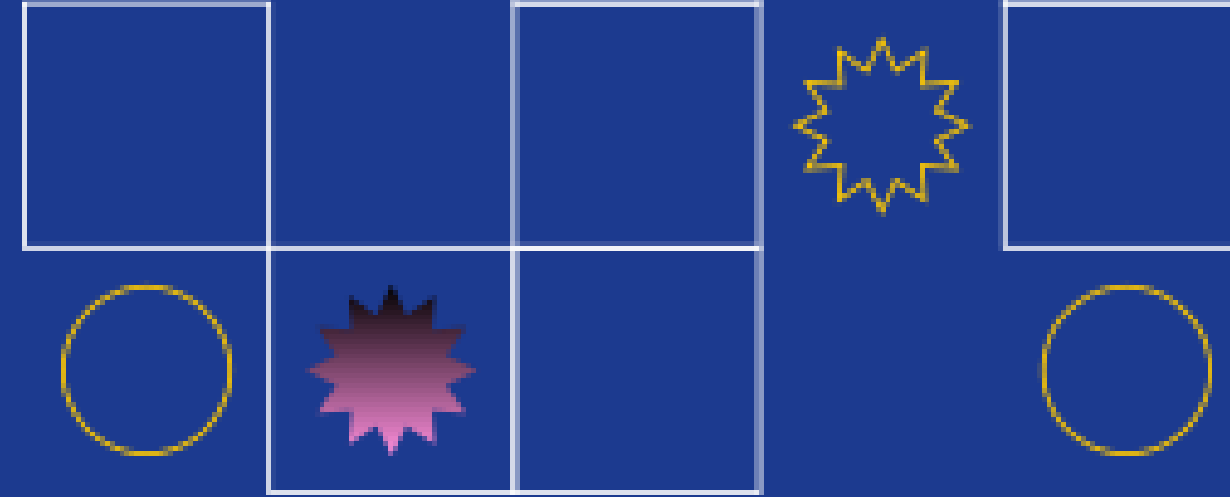
```
1 SELECT COUNT(DISTINCT Kredi) AS  
2 FarkliKrediSayisi FROM Dersler;
```

149 %

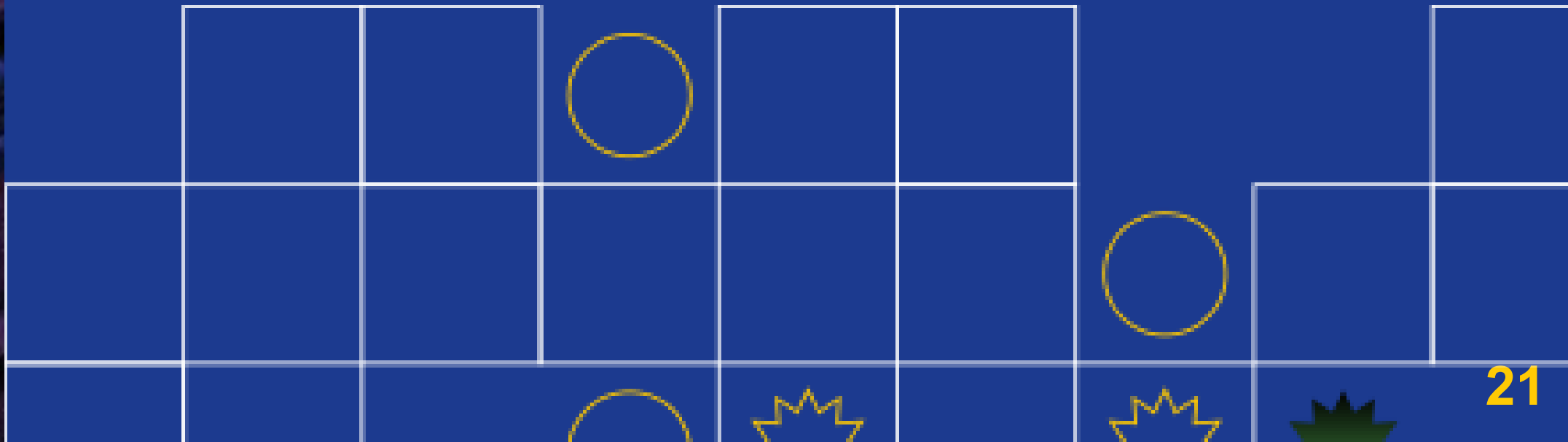
✓ No issues found

Results Messages

	FarkliKrediSayisi
1	4



GROUP BY Komutu



GROUP BY

Komutu

GROUP BY ifadesi, bir tablodaki satırları bir veya daha fazla sütundaki değerlerine göre gruplara ayırmak için kullanılır.

Genellikle SUM, AVG, COUNT, MAX ve MIN gibi toplamsal (aggregate) fonksiyonlarla birlikte kullanılır.

Temel amacı, tüm tablo yerine her bir grup (örneğin; her bölüm, her sınıf veya her cinsiyet) için ayrı ayrı hesaplamalar yapmaktır.

GROUP BY Komutu

Dikkat Edilmesi Gereken Hususlar

SELECT Kuralı: SELECT listesinde yer alan ancak toplamsal bir fonksiyonun içinde olmayan tüm sütunlar, mutlaka GROUP BY yan cümlesinde de yer almalıdır. Aksi takdirde SQL hata döndürür.

Filtreleme Farkı: Gruplama yapılmadan önce satırları filtrelemek için WHERE, gruplama yapıldıktan sonra gruplanmış sonuçları filtrelemek için HAVING kullanılır.

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 her bir Sinif düzeyinde (1. sınıf ve 2. sınıf) toplam kaç öğrenci olduğunu bulunuz.

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

149 % No issues found

Results Messages

	Sinif	OgrenciSayisi
1	1	15
2	2	13

GROUP BY Komutu

[Dersler] tablosunda her bir BolumID'ye ait derslerin toplam kredi miktarını hesaplayınız.

```
1 SELECT BolumID, SUM(Kredi) AS ToplamKredi
2 FROM Dersler GROUP BY BolumID;
```

149 % No issues found

Results Messages

	BolumID	ToplamKredi
1	1	22
2	2	10
3	3	10
4	4	7
5	5	3
6	6	5
7	7	4

GROUP BY Komutu

[Ogretmenler] tablosunda bölümlere göre uzman dağılımını görmek için her bir BoluID'deki öğretmen sayısını listeleyin.

```
1 SELECT BoluID, COUNT(*) AS OgretmenSayisi  
2 FROM Ogretmenler GROUP BY BoluID;
```

149 % No issues found

Results Messages

	BoluID	OgretmenSayisi
1	1	3
2	2	3
3	3	2
4	4	2
5	5	1
6	6	1
7	7	1

GROUP BY Komutu

[Notlar] tablosunda her bir SinavTuru (Vize, Final, Ödev vb.) için okuldaki genel puan ortalamasını listeleyin.

```
1 SELECT SinavTuru, AVG(Puan) AS OrtalamaPuan  
2 FROM Notlar GROUP BY SinavTuru;
```

149 % No issues found

Results Messages

	SinavTuru	OrtalamaPuan
1	Final	86.5731707317073
2	Odev	89
3	Vize	83.030612244898

GROUP BY Komutu

[Notlar] tablosunda her bir DersID bazında öğrencilerin aldığı en yüksek (MAX) puanı raporlayın.

```
1 SELECT DersID, MAX(Puan) AS EnYuksekPuan  
2 FROM Notlar GROUP BY DersID;
```

149 % No issues found

Results Messages

	DersID	EnYuksekPuan
1	1	96
2	2	92
3	3	90
4	4	93
5	5	91
6	6	95
7	7	95
8	8	87
9	9	86
10	10	86
11	11	99
12	12	91
13	13	88

GROUP BY Komutu

[Ogrenciler] tablosunda öğrencileri **Cinsiyetlerine** göre gruplandırarak toplam sayılarını bulunuz

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.

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.

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.

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.

GROUP BY Komutu

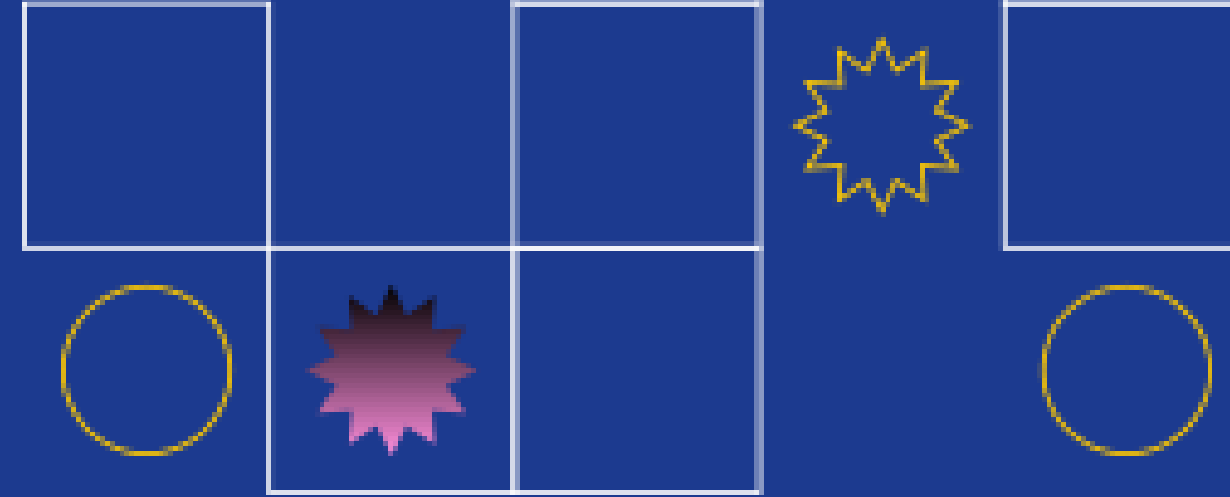
[Dersler] tablosunda bölümlere göre ders çeşitliliğini görmek için her **BolumID**'deki ders sayısını bulun.

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

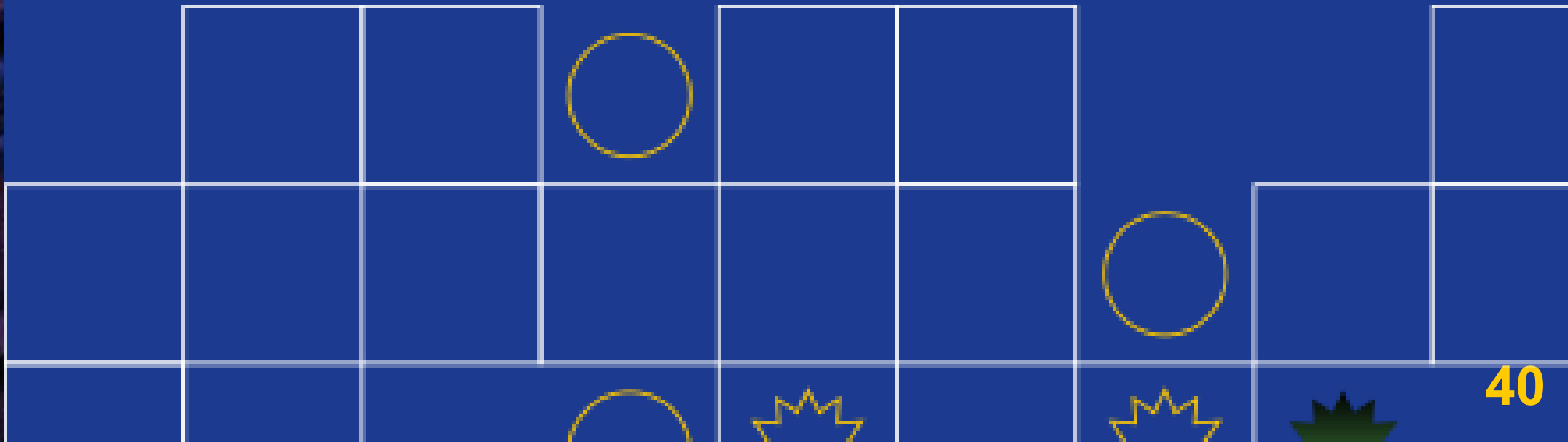
149 % No issues found

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



HAVING Komutu

HAVING, GROUP BY ile oluşturulmuş gruplar üzerinde filtreleme yapmak için kullanılır. WHERE ifadesi tekil satırları (kayıtları) filtrelerken, **HAVING** ifadesi toplamsal fonksiyonların (SUM, AVG, COUNT vb.) sonuçlarına göre grupları filtreler.

Örneğin; "Sınıfındaki öğrencileri say" dediğimizde GROUP BY kullanılır. Ancak "Öğrenci sayısı 5'ten fazla olan sınıfları getir" dediğimizde, hesaplanan bu sayı (COUNT) üzerinden bir filtreleme yapıldığı için **HAVING** devreye girer.

HAVING Komutu

Dikkat Edilmesi Gereken Hususlar

WHERE vs HAVING: WHERE ifadesi toplamsal fonksiyonlarla (örneğin `WHERE SUM(Puan) > 100`) kullanılamaz. Toplamsal fonksiyon sonuçları üzerinden filtreleme yapılacaksa tek seçenek HAVING'dir.

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 her bir DersID için ortalamayı hesaplayın, sadece ortalaması 90'dan büyük olan dersleri listeleyin.

```
1 SELECT DersID, AVG(Puan) AS Ortalama FROM Notlar
2 GROUP BY DersID
```

149 % No issues found

Results Messages

	DersID	Ortalama
1	1	87.777777777778
2	2	73.90625
3	3	83.3125
4	4	90.5
5	5	85.875
6	6	91
7	7	87.333333333333
8	8	85
9	9	83
10	10	82
11	11	92.625
12	12	86.5
13	13	86.5

HAVING Komutu

[Notlar] tablosunda her bir DersID için ortalamayı hesaplayın, sadece ortalaması 90'dan büyük olan dersleri listeleyin.

```
1 SELECT DersID, AVG(Puan) AS Ortalama FROM Notlar
2 GROUP BY DersID
```

DersID	Ortalama
1	87.7777777777778
2	73.90625
3	83.3125
4	90.5
5	85.875
6	91
7	87.3333333333333
8	85
9	83
10	82
11	92.625
12	86.5
13	86.5

```
1 SELECT DersID, AVG(Puan) AS Ortalama FROM Notlar
2 GROUP BY DersID HAVING AVG(Puan) > 90;
3
```

	DersID	Ortalama
1	4	90.5
2	6	91
3	11	92.625

HAVING Komutu

[Ogretmenler] tablosunda, bünyesinde 2'den fazla öğretmen barındıran BolumID'leri bulun.

```
1  SELECT BolumID, COUNT(*) AS OgretmenSayisi  
2  FROM Ogretmenler GROUP BY BolumID HAVING COUNT(*) > 2;
```

149 % No issues found

Results Messages

	BolumID	OgretmenSayisi
1	1	3
2	2	3

HAVING Komutu

[Dersler] tablosunda, toplam kredi miktarı 10'dan fazla olan bölümleri (BolumID) listeleyin.

```
1 SELECT BolumID, SUM(Kredi) AS ToplamKredi
2 FROM Dersler GROUP BY BolumID HAVING SUM(Kredi) > 10;
```

149 % No issues found

Results Messages

	BolumID	ToplamKredi
1	1	22

↓	↓	55
	BolumID	ToplamKredi

HAVING Komutu

[Notlar] tablosunda, en az 5 adet sınav kaydı girilmiş olan öğrencileri (OgrenciNo) bulun.

```
1 SELECT OgrenciNo, COUNT(*) AS SinavAdedi  
2 FROM Notlar GROUP BY OgrenciNo HAVING COUNT(*) >= 5;
```

149 % No issues found

Results Messages

	OgrenciNo	SinavAdedi
1	20231001	6

HAVING Komutu

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

HAVING Komutu

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

```
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.

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 listeleyin.

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.

HAVING Komutu

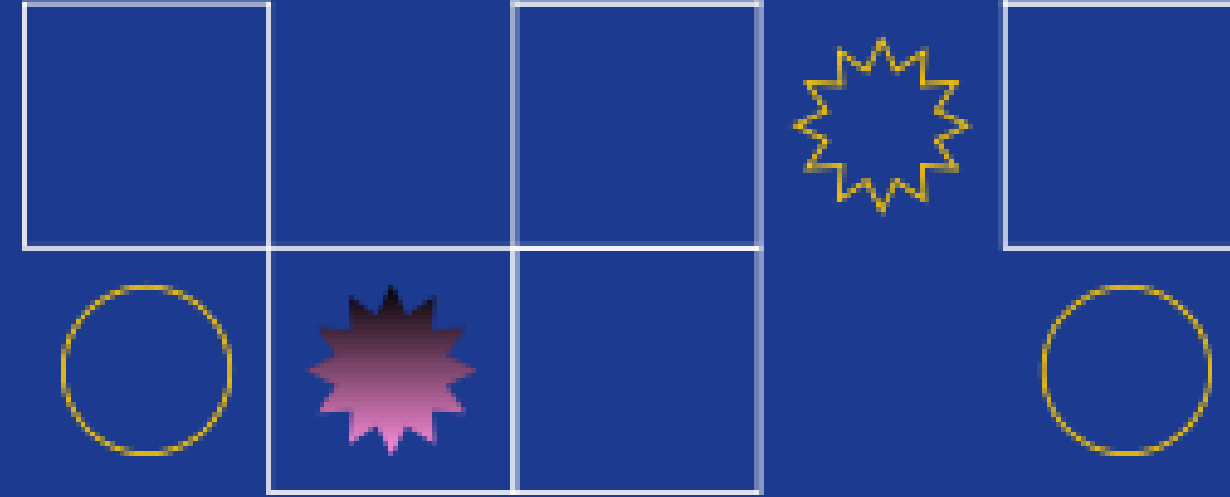
[Dersler] tablosunda, bir öğretmene (**OgretmenID**) atanmış toplam ders kredisi **5'ten fazla** ise o öğretmenleri listeleyin.

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

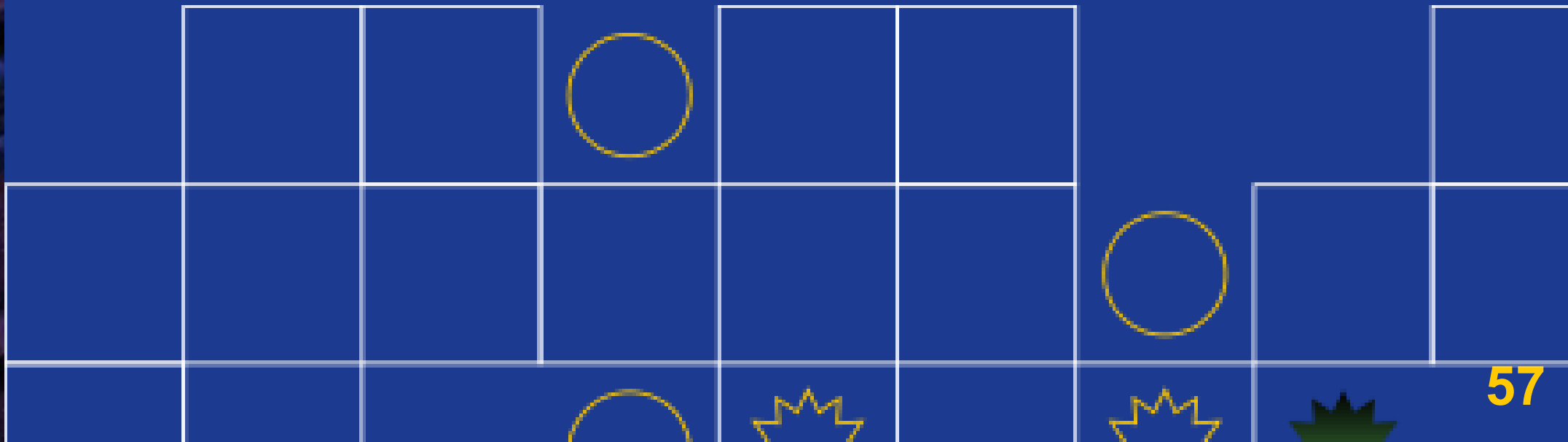
149 % ✓ No issues found

Results Messages

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



INNER JOIN İfadesi



INNER JOIN

İfadesi

INNER JOIN, iki veya daha fazla tablo arasındaki ortak olan (eşleşen) kayıtları yan yana getirmek için kullanılır. Eğer iki veya daha fazla tabloyu, ortak bir sütun (column) üzerinden eşleştirip sadece eşleşen kayıtları almak için kullanılır. Yani, iki tablodaki verilerden her iki tabloda da ortak olanlar sonucunda gösterilir; ortak olmayan veriler görünmez.

Örnek: Öğrenciler tablosunda sadece BolumID yazar. Bolumler tablosunda ise BolumAdi yazar. INNER JOIN sayesinde, öğrencinin adının yanına bölümün gerçek adını getirebiliriz.

INNER JOIN

İfadesi

Dikkat Edilmesi Gereken Hususlar

Ortak Sütun Şartı: Birleştirme yapabilmek için tablolar arasında ortak bir sütun (genellikle birinde Primary Key, diğerinde Foreign Key) bulunmalıdır.

ON İfadesi: Eşleşmenin hangi sütunlar üzerinden yapılacağını ON anahtar kelimesiyle belirtmek zorunludur.

Alias (Takma Ad) Kullanımı: Sorguyu kısaltmak ve okunabilirliği artırmak için tablo adlarına takma ad vermek (Örn: Öğrenciler AS O) çok yaygın ve faydalı bir pratiktir.

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 İfadesi

[Ogrenciler] ve [Bolumler] tablolarını birleştirerek, her öğrencinin Ad, Soyad ve okuduğu BolumAdi bilgilerini listeleyin.

```

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

```

149 % No issues found

Results Messages

	Ad	Soyad	BolumAdi
1	Ali	Koç	Arka Yüz Yazılım Geliştirme
2	Zeynep	Arslan	Ön Yüz Yazılım Geliştirme
3	Mert	Yıldız	Arka Yüz Yazılım Geliştirme
4	Ali	AKGÜN	Arka Yüz Yazılım Geliştirme
5	Ayşe	ATAKUL	Arka Yüz Yazılım Geliştirme
6	Can	BAYRAM	Ön Yüz Yazılım Geliştirme
7	Elif	ÖZTÜRK	Ön Yüz Yazılım Geliştirme
8	Murat	KAYA	Sivil Havacılık Kabin Hizmetleri
9	Zeynep	ÇETİN	Makine Resim ve Konstrüksiyonu
10	Emre	ŞAHİN	Arka Yüz Yazılım Geliştirme
11	Selin	YILMAZ	Arka Yüz Yazılım Geliştirme
12	Burak	DEMİR	Ön Yüz Yazılım Geliştirme
13	Merve	ARSLAN	Ön Yüz Yazılım Geliştirme
14	Ahmet	KARA	Arka Yüz Yazılım Geliştirme
15	Fatma	ŞEN	Ön Yüz Yazılım Geliştirme
16	Mehmet	YILDIZ	Arka Yüz Yazılım Geliştirme
17	Esra	AKIN	Ön Yüz Yazılım Geliştirme
18	Hakan	ÇOBAN	Sivil Havacılık Kabin Hizmetleri
19	Gizem	KURT	Sivil Havacılık Kabin Hizmetleri

INNER JOIN İfadesi

[Ogretmenler] ve [Bolumler] tablolarını birleştirerek, öğretmenlerin Ad, Soyad ve hangi BolumAdi'nda görev yaptığını listeleyin.

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

149 % No issues found

Results Messages

	Ad	Soyad	BolumAdi
1	Furkan	DURMUŞ	Arka Yüz Yazılım Geliştirme
2	Kemal	KARGA	Arka Yüz Yazılım Geliştirme
3	Sümeyye	YANILMAZ	Ön Yüz Yazılım Geliştirme
4	Ferhat	PERÇİN	Ön Yüz Yazılım Geliştirme
5	Serdar	ÜNVER	Sivil Havacılık Kabin Hizmetleri
6	Sümeyye	ÖZER	Sivil Havacılık Kabin Hizmetleri
7	Tuba	YILDIZ KAPLAN	Makine Resim ve Konstrüksiyonu
8	Ali Samet	AYVAZ	Makine Resim ve Konstrüksiyonu
9	Mehmet	YILMAZ	Arka Yüz Yazılım Geliştirme
10	Ayşe	KAYA	Ön Yüz Yazılım Geliştirme
11	Ahmet	YILDIRIM	Havacılık
12	Tuncay	ALTUN	İlahiyat
13	Ahmet Faruk	DURSUN	Yazılım Mühendisliği

INNER JOIN İfadesi

[Dersler] ve [Ogretmenler] tablolarını birleştirerek, her DersAdi'ni ve o dersi veren öğretmenin Ad, Soyad bilgilerini listeleyin.

```

1  SELECT D.DersAdi, T.Ad, T.Soyad FROM Dersler AS D
2  INNER JOIN Ogretmenler AS T ON D.OgretmenID = T.OgretmenID;

```

149 % No issues found

Results Messages

	DersAdi	Ad	Soyad
1	Veritabanı Yönetim Sistemleri	Furkan	DURMUŞ
2	Web Programlama	Sümeyye	YANILMAZ
3	Algoritma ve Programlama	Kemal	KARGA
4	Bilgisayar Ağları	Furkan	DURMUŞ
5	Kullanıcı Arayüzü Tasarımı	Ferhat	PERÇİN
6	İşletme Yönetimi	Serdar	ÜNVER
7	Temel Programlama 1	Kemal	KARGA
8	Bulut Bilişim	Mehmet	YILMAZ
9	Web Tasarım	Sümeyye	YANILMAZ
10	İngilizce	Sümeyye	ÖZER
11	Makine Elemanları	Tuba	YILDIZ KAPLAN
12	Teknik Resim	Ali Samet	AYVAZ
13	Havacılık Güvenliği	Serdar	ÜNVER
14	Müşteri İlişkileri	Sümeyye	ÖZER
15	Nesne Yönelimli Programlama	Mehmet	YILMAZ
16	Mobil Uygulama Geliştirme	Ayşe	KAYA
17	Nesne Tabanlı Programlama	Ahmet Faruk	DURSUN
18	Güzel Ahlak	Tuncay	ALTUN
19	Dinamik	Ahmet	YILDIRIM

INNER JOIN İfadesi

[Notlar] ve [Dersler] tablolarını birleştirerek, alınan her notun hangi DersAdi'na ait olduğunu ve Puan bilgisini listeleyin.

```
1 SELECT D.DersAdi, N.Puan FROM Notlar AS N
2 INNER JOIN Dersler AS D ON N.DersID = D.DersID;
```

149 % No issues found

Results Messages

	DersAdi	Puan
1	Veritabanı Yönetim Sistemleri	85.5
2	Veritabanı Yönetim Sistemleri	90
3	Algoritma ve Programlama	78
4	Algoritma ve Programlama	82.5
5	Temel Programlama 1	88
6	Temel Programlama 1	95
7	Veritabanı Yönetim Sistemleri	92
8	Veritabanı Yönetim Sistemleri	88.5
9	Algoritma ve Programlama	75
10	Algoritma ve Programlama	80
11	Web Programlama	70
12	Web Programlama	75.5

HAVING Komutu

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

HAVING 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
14	Can	BAYRAM	80
13	Can	BAYRAM	82
15	Can	BAYRAM	72.5
11	Can	BAYRAM	70

HAVING Komutu

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

HAVING 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

HAVING Komutu

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

HAVING 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

HAVING Komutu

Sınava Hazırlık :

[Notlar] tablosunda **Vize** sınavı olan öğrencilerin adlarını, ders adlarını ve puanlarını listeleyin.

	Ad	DersAdi	Puan
1	Ali	Veritabanı Yönetim Sistemleri	85.5
2			
3			
4			
5			
6			
7			
8			
9			
10			

