



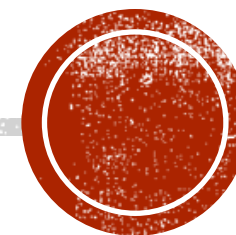
# SQL PROGRAMLAMA

personel_no	ad	soyad	cinsiyet	dogum_tarihi	dogum_yeri	baslama_tarihi	calisma_saati	maas	prim
1	Ahmet	EKİNCİ	E	1985-03-25	Elazığ	2002-03-01	35	2500,0000	350,0000
2	Burak	TURHAN	E	1980-04-15	Samsun	2002-05-03	30	2800,0000	250,0000
3	Ceyda	SAKARYA	K	1983-03-23	Rize	2005-05-12	30	3000,0000	250,0000
4	Demet	ÇOLAK	K	1982-05-05	Elazığ	2005-06-17	40	2800,0000	350,0000
5	Evrin	ÇITAK	K	1987-12-05	Ordu	2000-01-12	30	4500,0000	500,0000
6	Figen	ESMER	K	1982-10-22	Ankara	1998-05-07	35	4000,0000	250,0000

- Personel tablosunda adı s ile başlayan müşterilerin tüm bilgilerini listeyen komut satırını yazınız.
- Personel tablosunda şehri Antalya ve İstanbul olan müşterilerin tüm bilgilerini listeyen komut satırını yazınız.
- Personel tablosundaki, şehir alanında yer alan kayıtları birer kez yazdırmak istersek (yani hangi şehirden müşteriler var onu öğrenmek istiyoruz. bu şehirleri de ekrana birer kez yazdırmak istiyoruz).
- prim 200'den büyük ve çalışma saati 35 eşit olan personellerin adını ve soyadını listeyen komut satırını yazınız.



# SQL SERVER KOMUTLARI







## VERİ DÜZENLEME DİLİ

Bu alt dil bir tabloya veri ekleme (INSERT), silme (DELETE) ve güncelleme(UPDATE) yapmanın yanı sıra verileri seçmek ve raporlamak için SELECT ifadesi ve SELECT ifadesi ile birlikte kullanılan, INTO, FROM, WHERE, LIKE, GROUP BY, ORDER BY, HAVING... gibi çok yan ifade vardır. Bu ifadelere ait örnekleri bir çok yerde bulmak mümkündür.

# ARİTMETİK İFADELER

`SELECT ad, soyad, maas, maas*1.1 as 'Zamlı Maaş' from personel`

**Soru:** `SELECT Ogr_No,Ders_adi, Vize*0.4+Final*0.6 from personel where Ogr_No=3`

Ogr_No	Ders_adi	Vize	Final
1	Veri Tabanı	50	80
2	Görüntü İşleme	35	60
3	Yapay zeka	70	40

OPERATÖR	İŞLEVİ
** veya ^	Üs alma
*	Çarpma
/	Bölme
+	Toplama
-	Çıkarma

**Cevap:** Ogr\_No Ders\_adi

-----  
3 Yapay zeka 52



# GRUPLAMA (KÜMELEME) FONKSİYONLARI

- SUM (Topla) Fonksiyonu
- AVG (Ortalama) Fonksiyonu
- MAX (En Büyük) Fonksiyonu
- MIN (En Küçük) Fonksiyonu
- COUNT (Say) Fonksiyonu



# SUM FONKSİYONU

- Belirtilen sütundaki bütün değerlerin toplamını alır. Genel kullanımı aşağıdaki gibidir.

```
Select SUM(sütun_adı) FROM tablo_adı;
```

**Soru:** Personel tablosunda, cinsiyeti erkek olanların toplam maaşını listeleyen SQL komut satırını yazınız.

```
Select SUM(maas) FROM personel where cinsiyet='E'
```



# AVG FONKSİYONU

- Belirtilen sütundaki bütün değerlerin ortalamasını alır. Genel kullanımı aşağıdaki gibidir.

```
Select AVG(sütun_adı) FROM tablo_adı;
```

**Soru:** Doğum yeri Elazığ olan personellerin maaşlarının ortalamasını hesaplayan komut satırını yazınız.





# MAX FONKSİYONU

- Belirtilen sütundaki en büyük değeri verir. Genel kullanımını aşağıdaki gibidir.

```
Select MAX(sütun_adı) FROM tablo_adı;
```

**Soru-1:** SELECT MAX(maas) FROM personel

**Soru-2:** En yaşlı personel ile en genç personel arasındaki yaş farkını yıl olarak veren SQL komutunu yazınız.

```
SELECT DATEDIFF(YEAR,MIN(dogum_tarihi),MAX(dogum_tarihi))  
as 'Yaş Farkı' FROM Personel
```



# MIN FONKSİYONU

- Belirtilen sütundaki en küçük değeri verir. Genel kullanımı aşağıdaki gibidir.

Select MIN(sütun\_adı) FROM tablo\_adı;

**Soru-1:** SELECT MIN(maas) FROM personel

**Soru-2:** SELECT MIN(maas) FROM personel where cinsiyet='e'



# COUNT FONKSİYONU

Tablo içinde, her hangi bir sayma işlemi gerçekleştirmek için kullanılır. COUNT fonksiyonu sorgu sonucu dönen kayıt sayısını verir. Eğer parametre olarak “\*” girilirse tablodaki tüm kayıt sayısını verir. Parametre olarak bir sütun adı verilirse, o sütundaki içeriği NULL (boşluk) olmayan tüm kayıt sayısını verir. Genel kullanımı aşağıdaki gibidir.;

Select COUNT(sütun\_adı) FROM tablo\_adı;

**Soru-1:** SELECT COUNT(\*) FROM personel

**Soru-2:** SELECT COUNT(\*) FROM personel where calisma\_saati=35

**Soru-3:** SELECT COUNT(DISTINCT ad) FROM personel



# SQL GRUPLANDIRMA

- GROUP BY (Gruplandır) Deyimi
- HAVING (Sahip) Deyimi



# GROUP BY İFADESİ

“GROUP BY” yardımcı sözcüğü bir alana göre kayıtları gruplamak için kullanılır. Başka bir deyişle, tablodaki verileri belirli özelliklere göre gruplandırmak için kullanılır. Genel kullanımı aşağıdaki gibidir.

```
SELECT [ DISTINCT | ALL ] <sütun(lar)> FROM <tablo adı (lar)>  
[ WHERE <şart (lar)> ]  
[ GROUP BY <sütunlar>]
```

**Soru-1:** SELECT cinsiyet, COUNT(maas) FROM personel GROUP BY cinsiyet

**Soru-2:** SELECT cinsiyet, SUM(maas) FROM personel where dogum\_yeri='Samsun'  
GROUP BY cinsiyet



**Soru-3:** SELECT dogum\_yeri, count(personel\_no) FROM personel GROUP BY dogum\_yeri;

**Soru-4:** Select DISTINCT(DATEPART(Year,(dogum\_tarihi))) from personel GROUP BY dogum\_tarihi

**Soru-5:** Personellerin hangi doğum tarihinden kaç tane bulunduğunu listeleyen SQL komutunu yazınız.

Cevap: Select DATEPART(Year,(dogum\_tarihi)),COUNT(DATEPART(Year,(dogum\_tarihi))) from personel GROUP BY DATEPART(Year,(dogum\_tarihi))

**Soru-6:** SELECT cinsiyet, MIN(maas) as 'en düşük', MAX(maas) as 'en yüksek' from personel group by cinsiyet

**Soru-7:** Select DATEPART(Year,(dogum\_tarihi)) as 'Doğum Tarihi', SUM(maas) as 'Toplam Maas', AVG(maas) as 'Ortalama Maas' from personel WHERE cinsiyet='k' GROUP BY DATEPART(Year,(dogum\_tarihi))

**Soru-8:** SELECT cinsiyet, dogum\_yeri, SUM(maas) FROM personel GROUP BY cinsiyet,dogum\_yeri





# HAVING İFADESİ

Gruplandırarak kümeleme fonksiyonları uygularken, koşul da verilebilir. Bu durumda, grup üzerindeki hesaplamalarla ilişkili koşul belirtirken, HAVING sözcüğü kullanmak gerekir. “HAVING” yardımcı sözcüğü “GROUP BY” yardımcı sözcüğü ile gruplanan kayıtlar üzerinde kısıtlama yapma işine yarar. Genel kullanımını aşağıdaki gibidir.

```
SELECT [ DISTINCT | ALL ] <sütun(lar)> FROM <tablo adı (lar)>  
[ WHERE <şart (lar)> ]  
[ GROUP BY <sütunlar> ]  
[ HAVING < grup kısıtlaması> ]
```



HAVING sözcüğü SELECT komutunda GROUP BY bulunmadığı zaman geçersizdir. HAVING sözcüğünü izleyen ifade içinde SUM, COUNT(\*), AVG, MAX yada MIN fonksiyonlarından en az biri bulunmalıdır. HAVING sözcüğü sadece gruplanmış veriler üzerindeki işlemlerde geçerlidir. WHERE sözcüğü bir tablonun tek satırları üzerinde işlem yapan koşullar içinde geçerlidir. Bazı durumlarda HAVING ve WHERE sözcükleri ile birlikte SELECT komutu içinde kullanılabilir.

NOT: Kümeleme fonksiyonları ile WHERE ifadesi birlikte kullanılamadığından HAVING ifadesine ihtiyaç duyulmuştur.

```
SELECT cinsiyet, AVG(maas) FROM personel where maas<3000 GROUP BY cinsiyet
```

```
SELECT cinsiyet, AVG(maas) FROM personel GROUP BY cinsiyet HAVING  
AVG(maas)>3500
```



**Soru-1:** SELECT dogum\_yeri, AVG(maas) FROM personel where cinsiyet='e'  
GROUP BY dogum\_yeri HAVING SUM(maas)<5000

**Soru-2:** SELECT dogum\_yeri, SUM(maas) FROM personel GROUP BY  
dogum\_yeri HAVING AVG(DATEDIFF(YEAR,dogum\_tarihi,GETDATE()))<40

**Soru-3:** SELECT dogum\_yeri, AVG(maas) FROM personel GROUP BY dogum\_yeri  
HAVING COUNT(\*)>2

**Soru-4:** SELECT dogum\_yeri, AVG(maas) FROM personel GROUP BY dogum\_yeri  
HAVING COUNT(\*)>1 and (SUM(maas)/SUM(prim)<10)



# INSERT İFADESİ

Bir tabloya bilgi eklemek için kullanılan komuttur. Genel kullanım şekli aşağıdaki gibidir ;

**INSERT INTO table\_name VALUES (deger1, deger2, deger3,...)**

**ÖRN-1:** INSERT INTO personel VALUES ('Muammer','Türkoğlu','E','1990-06-28','Elazığ','2002-01-12',30,3000,250)

**ÖRN-2:** INSERT INTO personel(ad, soyad, cinsiyet, calisma\_saati, maas)  
VALUES ('Mehmet','Türkoğlu','E',40,3500)



```
use sirket
```

```
INSERT INTO ogrenci(ogr_no,ad, soyad)
VALUES
(1, 'Ahmet', 'BÜYÜK'),
(2, 'Mehmet', 'TEK'),
(3, 'Ayşe', 'YILDIZ'),
(4, 'HATİCE', 'Polat'),
(5, 'YAVUZ', 'ADA');
```

	Column Name	Data Type	Allow Nulls
▶🔑	ogr_no	int	<input type="checkbox"/>
	ad	nvarchar(50)	<input checked="" type="checkbox"/>
	soyad	nvarchar(50)	<input checked="" type="checkbox"/>
	sehir	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>



# UPDATE İFADESİ

Tablo satırlarında güncelleme yapmak için SQL'de UPDATE komutu kullanılır. Koşul belirtilmemişse, belirtilen değişiklik tüm tablo satırları üzerinde gerçekleştirilir. Koşul belirtildiği takdirde, sadece koşulu sağlayan satırlar üzerinde değişiklik gerçekleştirilir. UPDATE komutunun yazılış biçimi aşağıdaki gibidir.

```
UPDATE tablo_adı  
SET sütun_adı_1=deger1, sütun_adı_2=deger2,...N  
WHERE kosul;
```

**ÖRN-1:** UPDATE personel SET calisma\_saati=30 WHERE dogum\_yeri='Malatya'

**ÖRN-2:** UPDATE personel SET ad='Mehmet', soyad='Tek' WHERE personel\_no=15

**ÖRN-3:** UPDATE personel SET maas=maas\*1.2

**ÖRN-4:** UPDATE personel SET maas=maas\*1.2 WHERE dogum\_yeri='Elazığ'





# DELETE İFADESİ

- Tablodaki bir kayıt veya kayıtlar silinmek istenirse DELETE komutu kullanılır. Genel kullanım şekli aşağıdaki gibidir ;

DELETE FROM tablo\_adı WHERE koşul;

ÖRN-1: DELETE FROM personel

(Personel tablosundaki tüm bilgileri siler)

ÖRN-2: DELETE FROM personel WHERE ad='Ahmet'

ÖRN-3: DELETE FROM personel WHERE maas IS NULL

(Personel tablosundan maas alanı boş olan personelleri siler. Not: IS NOT NULL ifadesi boş olmayan alanlar için kullanılır.)

