



# SQL Komutları (2)

Uzm. Murat YAZICI



# Sıralama

Sıralama işlemi için **SELECT** ifadesinde **ORDER BY** kullanılır.

Bu ifadede **ASC** kelimesi kullanılırsa sıralama küçükten büyüğe doğru (A-Z) , **DESC** kullanılırsa büyükten küçüğe doğru (Z-A) yapılır.

**Not:** Her ikisi de kullanılmazsa sıralama küçükten büyüğe doğru yapılır.

# ORDER BY

## Örneğin;

Öğrenciyi numarasına göre sıralayan SQL ifadesi :

**Select \* from ogrenci Order By ogrno ASC**

**Select \* from ogrenci Order By ogrno**

Öğrencileri ortalamasına göre (büyükten küçüğe) sıralamak için :

**Select ogrno, ad, soyad from ogrenci  
Order By ortalama DESC**

# ORDER BY

## Örneğin;

Öğrencileri ad ve soyada göre sıralamak için [A-Z]:

```
Select * from ogrenci  
Order By ad, soyad ASC
```

Sadece ada göre sıralamak isteseydik [A-Z]:

```
Select * from ogrenci  
Order By ad
```

# Fonksiyonlar

## AVG Fonksiyonu

Herhangi bir sütunun sayısal değerlerinin aritmetik ortalamasını bulmak için kullanılır.

Örnek, Öğrencilerin yaş ortalamasını bulmak için,

```
Select AVG(yas) from ogrenci
```

- Personel tablosunda görevi 'Memur' olanların ortalama maaşını hesaplamak için,

```
Select AVG(maas) from personel where gorev='Memur'
```

# Fonksiyonlar

## SUM Fonksiyonu

Herhangi bir sütunun sayısal değerlerinin toplamını bulur.

Örnek, Kitap fiyatlarının toplamını veren sorgu,

```
Select SUM(fiyat) from kitap
```

- Fiyatları farklı olan kitapların fiyatları toplamını bulmak için,

```
Select SUM(distinct fiyat) from kitap
```

# Fonksiyonlar

## MAX ve MIN Fonksiyonları

Herhangi bir sütunun içerdiği değerlerin en büyük (MAX) ve en küçük (MIN) olanlarını bulmak için kullanılır.

Örnek, En yüksek ortalama değerini bulmak için,

```
Select MAX(ortalama) from ogrenci
```

- Fizik bölümünde okuyan öğrencilerden en düşük olanının ortalaması için,

```
Select MIN(ortalama) from ogrenci where bolum='Fizik'
```



# Fonksiyonlar

## COUNT Fonksiyonu

Bir tablodaki kayıtların sayılması amacı ile bu fonksiyon kullanılır.

**COUNT(\*)** : Bu fonksiyon NULL değerleri de göz önüne alarak tüm kayıtların sayılmasını sağlar.

**COUNT(sütunAdı)** : Fonksiyon bu tarzda kullanılırsa belirtilen sütunda NULL değerler içermeyen tüm kayıtların sayılmasını sağlar.



# Fonksiyonlar

## COUNT Fonksiyonu

Örnek, Öğrenci tablosunda kaç kayıt bulunduğunu bulan SQL ifadesi,

```
Select COUNT(*) from ogrenci
```

- Bilgisayar bölümünde okuyup 3.0' ın üzerinde ortalamaya sahip öğrenci sayısını bulmak için,

```
Select COUNT(*) from ogrenci  
Where bolum='Bilgisayar' and ortalama>=3.0
```



# Veri Ekleme, Güncelleme ve Silme

**SELECT** deyimi ile yapılan işlemler sorgulama işlemleridir. Bu sorgulamalar tablo üzerinde herhangi bir değişikliğe yol açmazlar.

Eğer bir tabloya yeni bir kayıt eklenmesi, var olan satırların güncellenmesi ya da silinmesi isteniyorsa **SELECT** deyimi ile bu gerçekleştirilmez.

Bunlar için **INSERT**, **UPDATE** ve **DELETE** deyimleri kullanılır.



# Tablodaki Verileri Güncelleme

**UPDATE** SQL deyimiyle gerçekleştirilir.

**UPDATE** tablo

**SET** sütun1=değer1, sütun2=değer2,...

**WHERE** şart

**Örneğin;** Personel tablosunda maaşları 1000 ile 1500 arasında olan çalışanların maaşlarını 2000 yapan SQL ifadesi,

**UPDATE** personel

**SET** maas=2000

**WHERE** maas **BETWEEN** 1000 **AND** 1500



# Tablodaki Verileri Güncelleme

**Örneğin;** Tüm personelin maaşlarına %10 zam yapan SQL ifadesi,

```
UPDATE personel SET maas=maas*1.1
```

**Örneğin;** Görevi 'Şef' olan personellerin görevini 'Müdür' olarak değiştiren SQL ifadesi,

```
UPDATE personel SET gorev='Müdür'  
WHERE gorev='Şef'
```

# Tablodan Satır Silmek

**DELETE** SQL deyimiyle gerçekleştirilir.

**DELETE FROM** tablo  
**WHERE** şart

**Örneğin;** Öğrenci tablosunda eposta adresi boş (NULL) olan kayıtları silmek için,

```
DELETE FROM ogrenci  
WHERE eposta IS NULL
```



# Veri Tanımlama Dili (DDL)

Veritabanında nesneler oluşturmak için gerekli olan ifadeleri sağlamaktadır. Bu ifadeler, veritabanı veya tablo gibi yeni bir nesnenin oluşturulması, var olan nesne üzerinde değişiklikler yapılması veya nesnenin yok edilmesi için kullanılır.

SQL dilinde 3 adet veri tanımlama ifadesi bulunmaktadır.

Bunlar: **CREATE**, **ALTER** ve **DROP** ifadeleridir.



# Veri Tanımlama Dili (DDL)

**CREATE** : Veritabanı ya da tablo oluşturmak için kullanılır.

```
CREATE DATABASE universite
```

**ALTER** : Var olan bir nesne üzerinde değişiklikler yapmak için kullanılır.

```
ALTER TABLE personel  
ADD eposta VARCHAR(40) NULL
```

**DROP** : Veritabanındaki herhangi bir nesneyi kaldırmak için kullanılır.

```
DROP TABLE dersler
```



# SQL' de Gruplama

Bir tablonun satırları gruplara ayrılarak fonksiyonların bunlara uygulanması mümkündür. Gruplara ayırmak için **SELECT** deyimi içerisinde **GROUP BY** deyimi kullanılır.

**Örneğin:** Personel tablosunda her bölümde çalışanların ortalama maaşını hesaplamak için :

```
SELECT bolum, AVG(maas) FROM personel  
GROUP BY bolum
```