



# SQL – Yapısal Sorgulama Dili (2)

## 2. Veri İşleme Dili (DML-Data Manipulation Language)

Veri işleme dili veritabanında kayıtlı veriler üzerinde işlem yapar.

Veri işleme dilinin temel ifadeleri :

- A. Select :** Veritabanındaki kayıtları sorgular/listeler
- B. Insert :** Yeni kayıt eklemek için kullanılır.
- C. Update :** Daha önce olan kaydı günceller.
- D. Delete :** Veritabanındaki kaydı silmek için kullanılır.

## A. SELECT

- Tablo yada tablolarda istenilen verinin seçimi için kullanılır.
- Seçilen verilerde değişim olmaz.

□ **Kullanımı:** **SELECT** sütunlar **FROM** tablo isim(leri)

SELECT ifadesinden sonra sorgu sonucunda ekranda görüntülenmesi istenen sütun isimlerinin arasına virgül konarak yan yana yazılır.

Örneğin **SELECT** ad, soyad **FROM** ogrenciler

Bu sorgu çalıştırıldığında, *ogrenciler* tablosunda yer alan öğrencilerin sadece Adı ve Soyadı kayıtları gösterilecektir.

- Eğer tabloda bulunan bütün alanlar listelenmek istenirse \* işareti kullanılır.

Örneğin **Select \* FROM MUSTERİ**

MUSTERİ

musteriNo	ad	soyad	sehir	alisveris_puani
10	Dilaver	URBAŞ	Konya	230
20	Mustafa	TEMEL	Konya	530
30	Erhan	GÜLER	Artvin	650
40	Turan	ÖZCAN	Ankara	150
50	Selçuk	YILMAZER	Artvin	10
60	Buğra	KURT	İstanbul	980

# DISTINCT Deyimi

Listelen kayıtlardan tekrarlayan kayıtlardan birini göstermek için kullanılır.

- **Kullanımı:** **SELECT DISTINCT** sütun\_isim(leri) **FROM** tablo\_isim(leri)

Örneğin *SELECT DISTINCT sehir FROM MUSTERI*

## MUSTERI

sehir
Konya
Artvin
Ankara
İstanbul

\* Bu sorgu sayesinde kaç farklı ilden müşteri kaydı olduğu öğrenilebilir.

## WHERE Deyimi

- Tablo içerisinde istenen bir şarta uygun olan kayıtların listelenmesini sağlar.

**Kullanımı:** **SELECT** sutun\_isim(leri) **FROM** tablo\_isim(leri) **WHERE** koşullar  
Koşullar yazılırken aşağıdaki operatörler kullanılır.

- ☐ Karşılaştırma
- ☐ Mantıksal
- ☐ Kümeleme

# Karşılaştırma Operatörleri

---

- Karşılaştırma operatörleri, verilen iki ifadenin birbirleri ile kıyaslanmasını sağlamaktadır. Bu ifadeleri şöyle sıralayabiliriz.
  - $X = Y$  , X ifadesi Y ifadesine eşittir.
  - $X > Y$  , X ifadesi Y ifadesinden büyüktür.
  - $X < Y$  , X ifadesi Y ifadesinden küçüktür.
  - $X \Rightarrow Y$  , X ifadesi Y ifadesine eşit ve büyüktür olabilir.
  - $X \Leftarrow Y$  , X ifadesi Y ifadesine eşit ve küçüktür olabilir.
  - $X \neq Y$  , X ifadesi Y ifadesine eşit değildir.

Şart ifadesi içerisinde kullanılacak ifade rakam ise tırnak işareti kullanılmaz. Eğer metin tabanlı bir ifade ise tek tırnaklar arasına söz konusu ifade yazılır.

Örneğin MUSTERİ tablosu ile ilgili bazı sorgulamalar aşağıda verilmiştir.

musteriNo	ad	soyad	sehir	alisveris_puani
10	Dilaver	URBAŞ	Konya	230
20	Mustafa	TEMEL	Konya	530
30	Erhan	GÜLER	Artvin	650
40	Turan	ÖZCAN	Ankara	150
50	Selçuk	YILMAZER	Artvin	10
60	Buğra	KURT	İstanbul	980

❖ **SELECT \* FROM MUSTERİ WHERE  
musteriNo=10**

musteriNo	ad	soyad	sehir	alisveris_puani
10	Dilaver	URBAŞ	Konya	230

❖ **SELECT \* FROM MUSTERİ WHERE  
sehir='KONYA'**

musteriNo	ad	soyad	sehir	alisveris_puani
10	Dilaver	URBAŞ	Konya	230
20	Mustafa	TEMEL	Konya	530



- Bir başka örnek MUSTERİ tablosunda alış veriş puanı 500 ‘ün üzerinde olan müşterileri görmek için aşağıdaki sorgu yazılır.
- **SELECT \* FROM MUSTERİ Where alisveris\_puani>500**
- Bu sorgu sonunda dönen cevap aşağıdaki gibi olacaktır.

### MUSTERİ

musteriNo	ad	soyad	sehir	alisveris_puani
20	Mustafa	TEMEL	Konya	530
30	Erhan	GÜLER	Artvin	650
60	Buğra	KURT	İstanbul	980

# Mantıksal Operatörler

---

- Mantıksal operatörler, bütün dillerde olduğu gibi iki yada daha fazla şartın kıyaslanması amacıyla kullanılmaktadır. Bu operatörler şu ifadelerden oluşmaktadır.
  - AND (ve)
  - OR (yada)
  - NOT (değil)

# Mantıksal Operatörler (devam...)

---

## ❑ AND operatörü

Örneğin; Siparisler tablosunda, nakliye ücreti 10 TL den küçük ve Ankara' ya gönderilen Siparisler listelensin.

```
SELECT
    *
FROM
    Siparisler
WHERE
    (NakliyeUcreti < 10) AND (GonderilenIl='Ankara')
```

# Mantıksal Operatörler (devam...)

---

## ❑ OR operatörü

Örneğin; Siparisler tablosunda, nakliye ücreti 10 TL den küçük olanlar **ya da** Ankara' ya gönderilen Siparisler listelensin.

```
SELECT
    *
FROM
    Siparisler
WHERE
    (NakliyeUcreti < 10) OR (GonderilenIl='Ankara')
```

# Mantıksal Operatörler (devam...)

---

## ❑ NOT operatörü

Örneğin; Siparisler tablosunda, nakliye ücreti 10 TL den **küçük olmayan** Siparisler listelensin.

```
SELECT
    *
FROM
    Siparisler
WHERE
    NOT (NakliyeUcreti < 10)
```

# Kümeleme Operatörleri

---

Kümeleme operatörleri, tablo içerisindeki verileri çeşitli kriterlere göre gruplamayı sağlayan operatörlerdir.

Bu operatörler;

- ☐ BETWEEN
- ☐ IN
- ☐ LIKE

Bu komutlar yine WHERE komutundan sonra kullanılmaktadır.

## Kümeleme Operatörleri (devam...)

---

### ❑ BETWEEN :

Belli bir aralıktaki kayıtların sorgulanmasında kullanılmaktadır.

Bu komutta alt ve üst sınır tanımlamaları yapılmaktadır.

Genel kullanımı şöyledir.

`BETWEEN Alt_Sınır AND Üst_Sınır`

- Örneğin nakliye ücreti 10 ve 20 TL arasındaki Siparislerin detayları listelensin.

```
SELECT
    *
FROM
    Siparisler
WHERE
    NakliyeUcreti BETWEEN 10 AND 20
```

# Kümeleme Operatörleri (devam...)

---

## ❑ IN

Kayıtların tablodaki değerlerle karşılaştırılması söz konusu olduğunda kullanılmaktadır.

- Örneğin; Ankara, İzmir ve İstanbul'a gönderilen Siparisler listelensin.

```
SELECT
    SiparisNo,
    GonderilenIl
FROM
    Siparisler
WHERE
    GonderilenIl IN ( 'Ankara', 'İzmir', 'İstanbul' )
```



# Kümeleme Operatörleri (devam...)

---

## ❑ LIKE

WHERE komutu ile birlikte kullanılan ve sözcüğün tamamı ile değil de sadece belli bir bölümü ile arama yapmayı sağlayan bir komuttur.

- Örneğin, Siparislerin gönderildiği kişilerden adları N ile başlayan kişiler listelensin.

```
SELECT
    GonderilenAd
FROM
    Siparisler
WHERE
    GonderilenAd LIKE 'N%'
```

# Kümeleme Operatörleri (devam...)

---

- Örneğin, Siparislerin gönderildiği kişilerden adları N ile biten kişiler listelensin.

SELECT

GonderilenAd

FROM

Siparisler

WHERE GonderilenAd LIKE '%N';

## Kümeleme Operatörleri (devam...)

---

Bir başka örnek verilecek olursa, Siparislerin gönderildiği kişilerden adlarının içinde N harfi bulunan kişiler listelensin.

```
SELECT
    GonderilenAd
FROM
    Siparisler
WHERE
    GonderilenAd LIKE '%N%'
```

## Kümeleme Operatörleri (devam...)

---

- LIKE deyimi ile birlikte kullanılan karakterlerden bir diğeri \_ işaretidir. Bu işaret bir harf veya rakamın yerini tutar.
- Örneğin, Siparislerin gönderildiği kişilerden adlarının içinde sondan ikinci harfi N olan kişileri listeleyen sorgu aşağıda verilmiştir.

```
SELECT
    GonderilenAd
FROM
    Siparisler
WHERE
    GonderilenAd LIKE '%N_'
```

# ORDER BY İfadesi

---

- **ORDER BY** komutu, önceden belirlenmiş olan sütuna göre sıralama işlemlerini gerçekleştirmektedir.
- Sıralama işleminde sıralama yapılacak olan sütun, rakamsal ifadelerden oluşuyorsa sıralama küçük rakamdan büyüğe yada büyük rakamdan küçüğe doğru sıralanabilmektedir.
- Eğer sütun alfabetik değerlerden oluşuyorsa sıralama işlemi A'dan Z' ye doğru yada Z' den A' ya doğru yapılabilir.

# ORDER BY İfadesi (devam...)

---

- ORDER BY komutu ile istenilen sıralama biçiminin yapılabilmesi için ardından bir komut almaktadır. Bu komutlar;
  - **ASC**: Sütundaki verileri küçükten büyüğe yada A' dan Z' ye doğru sıralama işlemini gerçekleştirir.
  - **DESC**: Sütundaki verileri büyükten küçüğe yada Z' den A' ya doğru sıralama işlemini gerçekleştirmektedir.

**Not:** Eğer ORDER BY komutundan sonra herhangi bir ifade seçilmez ise, varsayılan ifade olarak küçükten büyüğe doğru sıralama işlemi yapılmaktadır. Yani varsayılan arama tipi ASC' dir.

# ORDER BY İfadesi (devam...)

- Örnek: Bahçe İşleri veritabanında Siparisler tablosunda, *nakliye ücreti* 10 TL ile 25 TL arasında olan Siparislerin Siparisno, nakliyecisi ve gönderilen kişi bilgilerini nakliye ücretine göre büyükten küçüğe doğru sıralayan SQL sorgusu aşağıdaki gibi olacaktır. olacaktır.

```
SELECT
    SiparisNo,
    Nakliyecisi,
    GönderilenAd,
    NakliyeUcreti
FROM
    Siparisler
WHERE
    NakliyeUcreti BETWEEN 10 AND 25
ORDER BY
    NakliyeUcreti DESC
```

# ORDER BY İfadesi (devam...)

- Sorgu çalıştırıldıktan sonra yandaki ekran karşınıza çıkacaktır.

Siparişler Sorgu: Seçme Sorgusu

SiparişNo	Nakliyecisi	GönderilenAd	NakliyeÜcreti
11111	Yollar Kargo	Begüm Şafak	25,00 YTL
11091	Gök Nakliye	Özkan Nizam	24,50 YTL
11145	Yollar Kargo	Zehra Solmaz	23,00 YTL
11134	Nederci Taşımacılık	Durali Mumcu	22,00 YTL
11113	Nederci Taşımacılık	Aynur Şener	21,95 YTL
11103	Kuşkonmaz Nakliye	Rıfki Araştır	21,50 YTL
11132	Gök Nakliye	Necmi Devran	21,50 YTL
11087	Yollar Kargo	Can Temelli	20,00 YTL
11137	Uzay Kargo	Barış Timur	19,95 YTL
11095	Yollar Kargo	Feride Karaoğlu	18,95 YTL
11119	Gök Nakliye	Erkan Kocabıyık	18,50 YTL
11098		Musa Löküoğlu	18,50 YTL
11079	Yollar Kargo	Tayfun Süssler	18,00 YTL
11092	Uzay Kargo	Ertuğrul Ağır	17,00 YTL
11100	Nederci Taşımacılık	Halil Kaydu	16,50 YTL
11135	Uzay Kargo	Gökçe Özel	14,95 YTL
11090	Yollar Kargo	Barkın Nişancı	14,50 YTL
11106	Uzay Kargo	Ahmet Turgut	14,50 YTL
11110	Gök Nakliye	Erdem Aktan	14,50 YTL
11096		Feyzullah Deler	14,40 YTL
11080	Gök Nakliye	Fazıla Yücel	13,25 YTL
11161	Uzay Kargo	Fazilet Fahri	12,95 YTL
11122	Nederci Taşımacılık	Adnan Nohutçu	12,95 YTL
11125	Uzay Kargo	Yalın Ayaz	12,95 YTL
11138	Kuşkonmaz Nakliye	Perihan Okyay	12,95 YTL
11112	Nederci Taşımacılık	Lütfi Tahir	12,95 YTL
11102	Uzay Kargo	Mert Peker	12,45 YTL
11140	Uzay Kargo	Kasım Acil	12,25 YTL
11115	Uzay Kargo	Nadir Töre	11,95 YTL
11114	Nederci Taşımacılık	Ömer Kor	10,95 YTL
* (OtomatikSayı)			

Kayıt: 1 / 30



# ORDER BY İfadesi (devam...)

- ORDER BY içerisinde birden fazla ifade ile sıralama yaptırmak mümkündür.

Örneğin Siparisler tablosunda, nakliye ücreti 10 TL ile 25 TL arasında olan Siparislerin *Siparisno*, *nakliyecisi*, *gönderilen kişi* ve *Siparis tarihi* bilgilerini göstererek, önce nakliye ücretini büyükten küçüğe sonra da Siparis tarihini küçükten büyüğe doğru sıralayan SQL sorgusu aşağıdaki gibi olacaktır.

```
SELECT
    SiparisNo,
    Nakliyecisi,
    GönderilenAd,
    NakliyeUcreti,
    SiparisTarihi
FROM
    Siparisler
WHERE
    NakliyeUcreti BETWEEN 10 AND 25
ORDER BY
    NakliyeUcreti DESC, SiparisTarihi ASC
```

# ORDER BY İfadesi (devam...)

- Sorgu çalıştırıldıktan sonra yandaki ekran karşınıza çıkacaktır.

Siparişler Sorgu: Seçme Sorgusu

SiparişNo	Nakliyecisi	GönderilenAd	NakliyeÜcreti	SiparişTarihi
11111	Yollar Kargo	Begüm Şafak	25,00 YTL	24.01.2001
11091	Gök Nakliye	Özkan Nizam	24,50 YTL	15.01.2001
11145	Yollar Kargo	Zehra Solmaz	23,00 YTL	01.02.2001
11134	Nederci Taşımacılık	Durali Mumcu	22,00 YTL	30.01.2001
11113	Nederci Taşımacılık	Aynur Şener	21,95 YTL	25.01.2001
11103	Kuşkonmaz Nakliye	Rıfki Araştır	21,50 YTL	23.01.2001
11132	Gök Nakliye	Necmi Devran	21,50 YTL	30.01.2001
11087	Yollar Kargo	Can Temelli	20,00 YTL	12.01.2001
11137	Uzay Kargo	Başı Temur	19,95 YTL	31.01.2001
11095	Yollar Kargo	Feride Karaoğlu	18,95 YTL	22.01.2001
11098		Musa Löküoğlu	18,50 YTL	22.01.2001
11119	Gök Nakliye	Erkan Kocabıyık	18,50 YTL	26.01.2001
11079	Yollar Kargo	Tayfun Süsler	18,00 YTL	05.01.2001
11092	Uzay Kargo	Ertuğrul Ağır	17,00 YTL	16.01.2001
11100	Nederci Taşımacılık	Halil Kaydu	16,50 YTL	23.01.2001
11135	Uzay Kargo	Gökçe Özel	14,95 YTL	30.01.2001
11090	Yollar Kargo	Barkın Nişancı	14,50 YTL	14.01.2001
11106	Uzay Kargo	Ahmet Turgut	14,50 YTL	24.01.2001
11110	Gök Nakliye	Erdem Aktan	14,50 YTL	24.01.2001
11096		Feyzullah Deler	14,40 YTL	22.01.2001
11080	Gök Nakliye	Fazıla Yücel	13,25 YTL	05.01.2001
11112	Nederci Taşımacılık	Lütfi Tahir	12,95 YTL	25.01.2001
11122	Nederci Taşımacılık	Adnan Nohutçu	12,95 YTL	29.01.2001
11125	Uzay Kargo	Yalın Ayaz	12,95 YTL	29.01.2001
11138	Kuşkonmaz Nakliye	Perihan Okyay	12,95 YTL	31.01.2001
11161	Uzay Kargo	Fazilet Fahri	12,95 YTL	07.02.2001
11102	Uzay Kargo	Mert Peker	12,45 YTL	23.01.2001
11140	Uzay Kargo	Kasım Acil	12,25 YTL	01.02.2001
11115	Uzay Kargo	Nadir Töre	11,95 YTL	25.01.2001
11114	Nederci Taşımacılık	Ömer Kor	10,95 YTL	25.01.2001

\* (OtomatikSayı)

Kayıt: 1 / 30

## B. INSERT

- Tabloya yeni bir veri girişi yapmak için kullanılır.

### □ Kullanım Şekilleri :

- i. **INSERT INTO** tablo\_ismi (sutunAdları) **VALUES** (değerler)

**INSERT INTO** *uyeler* (*uye\_no*, *uye\_adi*, *soyad*, *cinsiyet*, *telefon*, *e\_posta*, *dogum\_tarih*)  
**VALUES** (1, 'Çoşkun', 'YILMAZ', 'E', '789456123', 'COSKUN@ABC.COM', '1978-08-09');

Sayısal değerlerin tek tırnak içerisinde yazılmasına gerek yoktur.

uye_no	uye_adi	soyad	cinsiyet	telefon	e_posta	dogum_tarih
1	Çoşkun	YILMAZ	E	789456123	COSKUN@ABC.COM	2004-05-04

## B. INSERT

### ii. INSERT INTO tablo\_ismi VALUES (değerler)

Tüm tabloya değerler ekleyeceğimiz zamanlar tablodaki alanları yazmamıza gerek yoktur.

INSERT INTO uyeler VALUES (2, 'Ayşe', 'KUL', 'K', '969456123', 'ayse@ABC.COM', '1980-04-03');

uye_no	uye_adi	soyad	cinsiyet	telefon	e_posta	dogum_tarih
1	Çoşkun	YILMAZ	E	789456123	COSKUN@ABC.COM	2004-05-04
2	Ayşe	KUL	K	969456123	ayse@ABC.COM	1980-04-03

Yukarıdaki sorguda tablodaki tüm alanların alacağı değerleri sırasıyla belirtilerek eklendi.

Örneğin aşağıdaki kodu yazmış olsaydık program hata verecekti.

INSERT INTO uyeler VALUES (3, 'Ferhat', 'GÖKMEN', 'ferhat@def.com');

**Hata:** 4. Sıradaki alan cinsiyet dolayısıyla cinsiyet alanına E veya K karakteri yerine e-posta verisi kaydedilemez. Farklı tipteki (ENUM yerine Varchar) bir veri girilmeye çalışıldı.

## B. INSERT

Kayıt eklemede hatalar oluşmaması için tablodaki alanlar belirtilir ve sonra bu alanların alacağı değerler yazılır.

Örneğin önceki örnekte karşılaşılan hatayı gidermek için aşağıdaki sorgu yazılır.

**INSERT INTO** uyeler (uye\_no, uye\_adi, soyad, e\_posta) **VALUES** (3, 'Ferhat', 'GÖKMEN', 'ferhat@def.com');

uye_no	uye_adi	soyad	cinsiyet	telefon	e_posta	dogum_tarih
1	Çoşkun	YILMAZ	E	789456123	COSKUN@ABC.COM	2004-05-04
2	Ayşe	KUL	K	969456123	ayse@ABC.COM	1980-04-03
3	Ferhat	GÖKMEN	E	NULL	ferhat@def.com	NULL

## C. UPDATE

- Tabloya daha önce girilmiş verilerin güncellenmesi için kullanılır.

### □ Kullanımı :

– **UPDATE tablo\_ismi SET sütun1=deger1, sütun2=deger2... WHERE koşul**

UYELER

uye_no	uye_adi	soyad	cinsiyet	telefon	e_posta	dogum_tarih
1	Çoşkun	YILMAZ	E	789456123	COSKUN@ABC.COM	2004-05-04
2	Ayşe	KUL	K	969456123	ayse@ABC.COM	1980-04-03
3	Ferhat	GÖKMEN	E	NULL	ferhat@def.com	NULL

Örneğin 2 numaralı müşterinin soyadını KUTLU ve eposta bilgisini [aysekutlu@abc.com](mailto:aysekutlu@abc.com) olarak değiştiren SQL komutu aşağıda verilmiştir.

**UPDATE uyeler SET soyad='KUTLU', e\_posta='aysekutlu@abc.com' WHERE uye\_no='2'**

## C. UPDATE

uye_no	uye_adi	soyad	cinsiyet	telefon	e_posta	dogum_tarih
1	Çoşkun	YILMAZ	E	789456123	COSKUN@ABC.COM	2004-05-04
2	Ayşe	KUTLU	K	969456123	aysekutlu@abc.com	1980-04-03
3	Ferhat	GÖKMEN	E	NULL	ferhat@def.com	NULL

Yukarıdaki tablodaki görüldüğü gibi üye numarası 2 olan kişinin soyad ve eposta bilgileri güncellendi.

- Bir başka örnek ise adı FERHAT ve soyadı GÖKMEN olan kullanıcının e\_posta ve doğum tarihi bilgileri güncelleyen SQL komut aşağıda verilmiştir.
- **UPDATE uyeler SET dogum\_tarih='1982-08-09', telefon='12345678901' WHERE uye\_adi='Ferhat' AND soyad='GÖKMEN'**

uye_no	uye_adi	soyad	cinsiyet	telefon	e_posta	dogum_tarih
1	Çoşkun	YILMAZ	E	789456123	COSKUN@ABC.COM	2004-05-04
2	Ayşe	KUTLU	K	969456123	aysekutlu@abc.com	1980-04-03
3	Ferhat	GÖKMEN	E	12345678901	ferhat@def.com	1982-08-09

## D. DELETE

Tabloya daha önce girilmiş verilerin silinmesi için kullanılır.

### ❑ Kullanımı

**DELETE FROM** tablo\_ismi **WHERE** koşul

Örneğin adı Çoşkun ve soyadı YILMAZ olan üyenin silinmesi için aşağıdaki SQL sorgusu yazılır.

**DELETE FROM uyeler WHERE uye\_adi='ÇOŞKUN' AND soyad='YILMAZ'**

uye_no	uye_adi	soyad	cinsiyet	telefon	e_posta	dogum_tarih
2	Ayşe	KUTLU	K	969456123	aysekutlu@abc.com	1980-04-03
3	Ferhat	GÖKMEN	E	12345678901	ferhat@def.com	1982-08-09



**Not:** WHERE den sonraki metin tabanlı karşılaştırmalarda büyük-küçük harf fark etmemektedir.



### 3. Veri Kontrol Dili (DCL-Data Control Language)

Veri tabanı üzerindeki verilere erişim yetkilerini düzenler.

- Veri kontrol dilinin temel ifadeleri
  - A. Grant:** Kullanıcıların kayıtlar üzerinde işlem yapmasına izin verir.
  - B. Revoke:** Daha önce yapılan tüm kısıtlama ve izinleri iptal eder.

## A. GRANT

**Grant**, kullanıcıya veritabanı veya nesneleri üzerinde çeşitli izinler vermek için kullanılır.

❑ **Kullanımı:** **GRANT** izinler **ON** nesne\_adi **TO** kullanıcılar

Örneğin; **GRANT DELETE,INSERT, UPDATE ON** SIPARISLER **TO** mus\_hiz\_kullanici

Müşteri hizmetleri kullanıcılarında ekleme, silme ve güncelleme yetkisi verilmiştir.

**Not:** Tüm yetkilerin verilmesi için ALL parametresi kullanılır.

Grant, yönetici yetkisine sahip kullanıcılar tarafından kullanılabilir.

İZİNLER		
Veritabanı için	Tablo İçin	Görünüm (View) için
CREATE DATABASE, BACKUP DATABASE, CREATE TABLE, CREATE VIEW	DELETE, INSERT, SELECT, UPDATE	DELETE, INSERT, SELECT, UPDATE

**Görünümler (View)** birer sanal tablodur. Veritabanı, Görünümler ile oluşan veriyi saklamaz.

Görünüm kullanmanın başlıca avantajları aşağıda verilmiştir.

- ❖ Güvenliği arttırmak.

Örneğin birden fazla ile çalışılıyorsa kullanıcılara verilerin bir kısmının gözükmemesini sağlar.

- ❖ Karmaşık sorguları basitleştirmek,

- ❖ Sorgu süresini kısaltmak,

- ❖ Raporlar oluşturmak

## C. REVOKE

**Revoke;** kullanıcıya veritabanı veya nesneleri üzerinde verilen tüm yetkileri **kaldırmak** için kullanılır.

❑ **Kullanımı:** **REVOKE** *izinler* **ON** *nesne adi* **FROM** *kullanıcı(lar)*

Örneğin **REVOKE DELETE,INSERT ON** SIPARISLER **FROM** mus\_hiz\_kullanici, satis\_temsilcisi

Her iki kullanıcı için Sipariş tablosunda silme ve ekleme yetkisi kaldırılmıştır.

**Not:** Revoke, yönetici yetkisine sahip kullanıcılar tarafından kullanılabilir.

Tüm yetkilerin kaldırılması için **ALL** parametres kullanılır.

Örneğin Satış temsilcisinin tüm yetkilerini kaldırmak için **REVOKE ALL FROM satis\_temsilcisi** komutu yazılır.

İZİNLER		
Veritabanı için	Tablo İçin	Görünüm (View) İçin
CREATE DATABASE, BACKUP DATABASE, CREATE TABLE, CREATE VIEW	DELETE, INSERT SELECT, UPDATE	DELETE, INSERT, SELECT , UPDATE

# Kaynakça

- ▶ Turgut Özseven, Veritabanı Yönetim Sistemleri-1 , Ekin Basın Yayın Dağıtım 6. Baskı
- ▶ <http://www.pau.edu.tr/ali/tr>
- ▶ <http://tr.wikipedia.org/wiki/Veritaban%C4%B1>
- ▶ <https://harunkolyigit.wordpress.com/2013/03/14/veritabani/>
- ▶ [http://akademik.maltepe.edu.tr/~senolerdogan/YZM\\_307/](http://akademik.maltepe.edu.tr/~senolerdogan/YZM_307/)
- ▶ <http://www.ortogon.com.tr/video/db/>
- ▶ [altanmesut.trakya.edu.tr/vt](http://altanmesut.trakya.edu.tr/vt)
- ▶ <http://akademik.maltepe.edu.tr/~eminborandag/VeriTabani/>