Veritabanı Yönetim Sistemleri

Yapısal Sorgulama Dili (SQL) I I

Hedefler

- SQL ile kayıt ekleme işlemleri yapabilmek
- SQL ile kayıt silme işlemleri yapabilmek
- SQL ile kayıt güncelleme işlemleri yapabilmek
- SQL ile seçilen kayıtları listeleme işlemleri yapabilmek

İçindekiler

1. Kayıt Ekleme

INSERT komutu bir tabloya yeni satırlar (yani yeni kayıtlar) eklemek için kullanılır. Bu komutta yazılan değerlerin özellikleri, tablodaki alanlarının özellikleriyle aynı olmalıdır.

Kullanım şekli aşağıdaki gibidir.

```
INSERT INTO <tablo adı> (<alan adları>) VALUES (<değerler>);
```

"INSERT INTO" yazıldıktan sonra kayıt eklenecek tablonun adı yazılır. Tablonun adı yazıldıktan sonra kayıt eklenecek alanların adı, daha sonra "VALUES" yazılır. En son olarak eklenecek alanların değerleri yazılır.

Örnek olarak, aşağıdaki SQL sorgusuyla oluşturulmuş bir tablomuz olsun.

```
CREATE TABLE Personel (
   SicilNo integer NOT NULL PRIMARY KEY,
   Ad char(8) NOT NULL,
   Soyad char(10) NOT NULL,
   EvliMi smallint,
   Maas money
);
```

Personel				
SicilNo	Ad	Soyad	EvliMi	Maas

Yukarıdaki tabloya aşağıdaki SQL sorguları ile iki yeni kayıt ekleyebiliriz.

```
INSERT INTO Personel (SicilNo, Ad, Soyad, EvliMi, Maas) VALUES (477, 'Ahmet',
'Taşdemir', 0, 2450);
INSERT INTO Personel (SicilNo, Ad, Soyad, EvliMi, Maas) VALUES (124,
'Abdurrahman', 'Karacadoğan', 1, 1550);
```

Kayıtlar eklendikten sonra tablomuz aşağıdaki gibi olur.

Personel				
SicilNo	Ad	Soyad	EvliMi	Maas
477	Ahmet	Taşdemir	0	2450
124	Abdurrah	Karacadoğa	1	1550

DİKKAT

İkinci kayıtta Ad ve Soyad alanlarına kaydedilmek istenen verilerin uzunluğu tablo oluştururken belirtilen uzunluktan fazla olduğu için sonlarının kırpıldığına dikkat ediniz.

Yine örnek olarak, aşağıdaki SQL sorgusuyla oluşturulmuş bir tablomuz olsun.

```
CREATE TABLE Adres (
   Ad char(8) NOT NULL,
   Soyad char(10) NOT NULL,
   PRIMARY KEY (Ad, Soyad),
   Adres char(30),
   Telefon char(12),
   SicilNo integer
);
```

Adres				
Ad	Soyad	Adres	Telefon	SicilNo

Bu tabloya aşağıdaki SQL sorguları ile yeni kayıt eklenebilir.

```
INSERT INTO Adres (Ad, Soyad, Adres, Telefon, SicilNo) VALUES ('Ahmet',
'Taşdemir', 'İstanbul', '0532 1234567', 477);

INSERT INTO Adres (Ad, Soyad, Adres, Telefon, SicilNo) VALUES ('Bekir', 'Ferik',
'Sakarya', '0505 9876543', 111);
```

Kayıtlar eklendikten sonra tablomuz aşağıdaki gibi olur.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111

INSERT INTO komutu yazılırken mutlaka bütün alanların adları yazılması zorunlu değildir. Veri kaydedilmek istenmeyen (boş geçilmek istenen) alanlar atlanır. Atlanan alanların değeri NULL (boş) olur.

Örneğin Adres tablosunda Adres ve Telefon alanları boş geçilerek bir kayıt eklemek için aşağıdaki komut kullanılır.

```
INSERT INTO Adres (Ad, Soyad, SicilNo) VALUES ('Hasan', 'Sert', 814);
```

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Hasan	Sert	*NULL*	*NULL*	814

Eğer tablo oluşturulurken NOT NULL şeklinde tanımlanmış (boş geçilemeyecek) bir alan boş geçilmeye çalışılırsa komut çalıştırılmak istendiğinde hata verecektir.

Örneğin, Adres tablosundaki Ad ve Soyad alanları zorunlu olduğu için aşağıdaki SQL komutu hata döndürecektir.

```
INSERT INTO Adres (Adres, Telefon, SicilNo) VALUES ('Trabzon', '0542 5678912',
161); -- HATA!
```

INSERT INTO komutundaki alanların tablodaki sıralama ile yazılması da zorunlu değildir. Burada önemli olan alan listesindeki sıralama ile değerler listesindeki sıralamanın aynı olmasıdır.

Örneğin, Adres tablosuna alan adlarının sırasını karışık yazarak yeni bir kayıt eklemek için aşağıdaki SQL komutu kullanılır.

```
INSERT INTO Adres (Soyad, SicilNo, Ad) VALUES ('Sevecen', 394, 'Tarık');
```

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Hasan	Sert	*NULL*	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

2. Kayıt Silme

DELETE FROM komutu bir tablonun bir veya birden fazla kaydının silinmesi amacıyla kullanılır.

Kullanım şekli aşağıdaki gibidir.

```
DELETE FROM <tablo adı> WHERE <koşul(lar)>;
```

"DELETE FROM" yazıldıktan sonra silinecek kaydın bulunduğu tablonun adı yazılır.

"WHERE" yazıldıktan sonra ise silme işlemi için kullanılacak koşullar yazılır.

Tablodaki tüm kayıtlar silinmek isteniyorsa koşul yazılmaz. Eğer koşul yazılmayacaksa WHERE anahtar kelimesi de yazılmaz.

ÖRNEK

Siparis tablosundaki ***tüm kayıtları silmek için*** aşağıdaki SQL komutu kullanılır.

```
DELETE FROM Siparis;
```

Eğer WHERE anahtar kelimesinden sonra bir koşul yazılmışsa, bu koşulu sağlayan bütün kayıtlar silinir.

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Personel				
SicilNo	Ad	Soyad	EvliMi	Maas
477	Ahmet	Taşdemir	0	2450
104	Aysu	Halin	0	1400
124	Abdurrah	Karacadoğa	1	1550

Yukarıdaki tablodan evli olmayan herkesi silen SQL sorgusu aşağıdaki gibidir.

```
DELETE FROM Personel WHERE EvliMi = 0;
```

Bu sorgu çalıştırıldıktan sonra tablonun yeni hali aşağıdaki gibi olacaktır.

Personel				
SicilNo	Ad	Soyad	EvliMi	Maas
124	Abdurrah	Karacadoğa	1	1550

ÖRNEK

Yine örnek olarak, aşağıdaki gibi bir tablomuz olsun.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Hasan	Sert	*NULL*	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

Bu tablodan sicil numarası 477 olan kaydı silmek için aşağıdaki SQL sorgusu kullanılır.

```
DELETE FROM Adres WHERE SicilNo = 477;
```

3. Kayıt Güncelleme

UPDATE komutu bir tablodaki bir veya birden fazla kaydın güncellenmesi için kullanılır.

Kullanım şekli aşağıdaki gibidir.

```
UPDATE <tablo adı> SET <alan adı = değer, ...> WHERE <koşul(lar)>;
```

Tıpkı DELETE FROM komutunda olduğu gibi, tablodaki tüm kayıtlar güncellenmek isteniyorsa koşul yazılmaz. Eğer koşul yazılmayacaksa WHERE anahtar kelimesi de yazılmaz.

ÖRNEK

Örnek olarak, aşağıdaki gibi bir tablomuz olsun.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Hasan	Sert	*NULL*	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

Yukarıdaki tabloda sicil numarası 477 olan personelin adresini “Sakarya” olarak değiştiren SQL sorgusu aşağıdaki gibi yazılır.

```
UPDATE Adres SET Adres = 'Sakarya' WHERE SicilNo = 477;
```

Yukarıdaki sorgu çalıştırıldığında tablodaki kayıtların yeni hali aşağıdaki gibi olacaktır.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	Sakarya	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Hasan	Sert	*NULL*	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

Aynı tablo üzerinden örneğimize devam edecek olursak, aşağıdaki sorguda koşul yazılmadığı için bütün kayıtlar güncellenecektir.

```
UPDATE Adres SET Adres = 'Trabzon';
```

Yukarıdaki sorgu çalıştırıldığında tablodaki kayıtların yeni hali aşağıdaki gibi olacaktır.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	Trabzon	0532 1234567	477
Bekir	Ferik	Trabzon	0505 9876543	111
Hasan	Sert	Trabzon	*NULL*	814
Tarık	Sevecen	Trabzon	*NULL*	394

Aynı anda birden fazla alandaki verinin güncellenmesi de mümkündür.

ÖRNEK

Örnek olarak, aşağıdaki gibi bir tablomuz olsun

Personel				
SicilNo	Ad	Soyad	EvliMi	Maas
477	Ahmet	Taşdemir	0	2450
104	Aysu	Halin	0	1400

Bu tabloda, sicil numarası 104 olan personelin hem medeni durumunu hem de maaşını güncellemek için aşağıdaki gibi bir sorgu kullanılır.

```
UPDATE Personel SET EvliMi = 1, Maas = 1800 WHERE SicilNo = 104;
```

Yukarıdaki sorgu çalıştırıldıktan sonra tablodaki kayıtların yeni hali aşağıdaki gibi olacaktır.

Personel				
SicilNo	Ad	Soyad	EvliMi	Maas
477	Ahmet	Taşdemir	0	2450
104	Aysu	Halin	1	1800

4. Kayıt Listeleme

SELECT komutu SQL dilinde en çok kullanılan komutlardan biridir. Bir veri tabanındaki bir veya birden fazla sayıdaki tablodan belli bir koşula uyan veya bütün kayıtları alarak ve istenilen formatta listelemek için kullanılır.

SELECT komutu tablonun yapısını veya verileri asla değiştirmez!

En basit hâliyle kullanım şekli aşağıdaki gibidir.

```
SELECT <alan adları> FROM <tablo adı> WHERE <koşul(lar)>;
```

"SELECT" yazıldıktan sonra listelenecek alanların adı yazılır.

Listelenecek alanların adı yazıldıktan sonra "FROM" yazılır ve sonra tablonun adı yazılır.

En son olarak "WHERE" ifadesinden sonra koşul veya koşullar yazılır.

Alan adları listesinde birden fazla alan adı yazılacaksa aralarına virgül konulur. Eğer tüm alanların seçilmesini istiyorsak * sembolünü kullanabiliriz.

ÖRNEK

Örnek olarak, aşağıdaki gibi bir tablomuz olsun.

Siparis		
SiparisNo	Urun	Musteri
1201	Masa	Seren
1245	Sandalye	Seren
1254	Masa	Melina

Yukarıdaki Siparis tablosundaki tüm alanlardaki verilerin listesini görmek için SQL sorgusu aşağıdaki gibi olmalıdır.

```
SELECT SiparisNo, Urun, Musteri FROM Siparis;
```

Yukarıdaki sorguda tablodaki bütün alanlar aynı sırayla yazılmıştır. Bunun yerine alan listesinde * işareti de kullanılabilir. Bu tablo için, aşağıdaki sorgu ile yukarıdaki sorgu birebir aynı işlevi görecektir.

```
SELECT * FROM Siparis;
```

Sadece ürün ve müşteriler listelenmek isteniyorsa aşağıdaki SQL sorgusu kullanılır.

```
SELECT Urun, Musteri FROM Siparis;
```

Yukarıdaki sorgu çalıştırıldığında ekranda aşağıdaki gibi bir liste görüntülenecektir.

Siparis	
Urun	Musteri
Masa	Seren
Sandalye	Seren
Masa	Melina

Tıpkı INSERT INTO komutunda olduğu gibi, alan adlarının tablodaki sırayla yazılması gerekmez. SELECT sorgusunda alan adları hangi sırayla yazılırsa, listede o sırayla görüntülenecektir.

ÖRNEK

Alanları tablodaki sıradan farklı bir sıralama listelemek için aşağıdaki gibi bir sorgu yazılabilir.

```
SELECT Urun, SiparisNo, Musteri FROM Siparis;
```

Yukarıdaki sorgu çalıştırıldığında ekranda aşağıdaki gibi bir liste görüntülenecektir.

Siparis		
Urun	SiparisNo	Musteri
Masa	1201	Seren
Sandalye	1245	Seren
Masa	1254	Melina

4.1. AS anahtar kelimesi

AS anahtar kelimesi SELECT komutuyla birlikte kullanılan ve kayıtların listelenmesi sırasında alan adlarının farklı bir başlıkla görüntülenmesini sağlayan bir komuttur.

AS anahtar kelimesi tablodaki alan adlarını asla değiştirmez. Sadece farklı bir başlık altında görüntülenmesini sağlar.

Kullanımı aşağıdaki gibidir.

```
SELECT <alan adi> AS <başlık> FROM <tablo adı>;
```

"SELECT" yazıldıktan sonra kayıtları listelenecek alanın adı yazılır. Listelenecek alanın adı yazıldıktan sonra "AS" yazılır ve sonra istenen başlık yazılır. En son olarak "FROM" yazılır ve en son olarak eklenecek alanların olduğu tablo adı yazılır.

ÖRNEK

Örnek olarak, aşağıdaki gibi bir tablomuz olsun.

Siparis		
SiparisNo	Urun	Musteri
1201	Masa	Seren
1245	Sandalye	Seren
1254	Masa	Melina

Bu tabloda Urun ve Musteri alanlarının başlıklarını sırayla UrunAdi ve MusteriAdi şeklinde görüntülemek istersek, aşağıdaki gibi bir SQL sorgusu yazmalıyız.

```
SELECT SiparisNo, Urun AS UrunAdi, Musteri AS MusteriAdi FROM Siparis;
```

Bu sorgu çalıştırıldığında tablo aşağıdaki gibi görüntülenir.

Siparis		
SiparisNo	UrunAdi	MusteriAdi
1201	Masa	Seren
1245	Sandalye	Seren
1254	Masa	Melina

Fark edeceğiniz üzere, bütün alan adları için AS anahtar kelimesinin yazılması zorunlu değildir. Sadece başlığını değiştirmek istediğimiz alan adlarından sonra "AS başlık" yazılır. Yukarıdaki örnekte SiparisNo alanının adı yazıldıktan sonra AS anahtar kelimesi kullanılmadığı için bu alanın başlığı değiştirilmeden görüntülenmiştir.

4.2. Ön ekler

Belli bir sayıda kayda ulaşmak için, tekrar eden kayıtların alınmaması için, alan adları yazılmadan önce bazı ön ekler yazılabilir.

Ön eklerin kullanım şekli aşağıdaki gibidir.

```
SELECT <ön ek> <alan adları> FROM <tablo adı> WHERE <koşul(lar)>;
```

Bu ön eklerden bazıları aşağıda listelenmiştir.

TOP

Belli sayıda kayıtın listelenmesini sağlar.

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Hasan	Sert	Trabzon	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

Bu tablonun ilk 2 kaydını listelemek için aşağıdaki komut kullanılır.

```
SELECT TOP 2 * FROM Adres;
```

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111

DISTINCT

Seçilen alanlardaki tekrarlı kayıtlardan sadece birini listeler.

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Siparis		
SiparisNo	Urun	Musteri
1201	Masa	Seren
1245	Sandalye	Seren
1254	Masa	Melina

Bu tablodan sadece ürün alanını listelemek istersek aşağıdaki sorgu yazılır.

```
SELECT Urun FROM Siparis;
```

Ancak bu durumda aynı ürün adı birden fazla kayıtta birden görüntülenecektir.

Siparis
Urun
Masa
Sandalye
Masa

Aynı ürün sadece bir kez listelensin isteniyorsa DISTINCT ön eki kullanılmalıdır.

```
SELECT DISTINCT Urun FROM Siparis;
```

Siparis
Urun
Masa
Sandalye

4.3. Sıralama

SELECT sorgularında kayıtları istediğimiz sırada listelemek için ORDER BY anahtar kelimesi kullanılır.

Kullanımı aşağıdaki gibidir.

```
SELECT <ön ek> <alan adları> FROM <tablo adı> WHERE <koşul(lar)> ORDER BY <alan
ad(lar)ı>;
```

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Ahmet	Sert	Trabzon	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

Bu tablodaki kayıtları sicil numarasına göre sıralı görüntülemek için aşağıdaki sorgu yazılır.

```
SELECT * FROM Adres ORDER BY SicilNo;
```

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Bekir	Ferik	Sakarya	0505 9876543	111
Tarık	Sevecen	*NULL*	*NULL*	394
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Ahmet	Sert	Trabzon	*NULL*	814

Eğer tersten sıralama yapılması isteniyorsa alan adından sonra DESC anahtar kelimesi yazılır.

```
SELECT * FROM Adres ORDER BY SicilNo DESC;
```

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Sert	Trabzon	*NULL*	814
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Tarık	Sevecen	*NULL*	*NULL*	394
Bekir	Ferik	Sakarya	0505 9876543	111

Sıralama için birden fazla alanın adı yazılabilir. Bu durumda ilk yazılan alandaki değerler aynı ise, bu kayıtlar kendi içinde ikinci alana göre sıralanır.

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Ahmet	Sert	Trabzon	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

Bu tabloda ada göre sıralama yapılması, adı aynı olan kişilerin de kendi içinde soyada göre sıralanması isteniyor. Bunun için aşağıdaki sorgu yazılır.

```
SELECT * FROM Adres ORDER BY Ad, Soyad;
```

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Sert	Trabzon	*NULL*	814
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Tarık	Sevecen	*NULL*	*NULL*	394

5. Koşullu Sorgular

Yukarıda da bahsedildiği gibi koşullar SELECT, DELETE FROM ve UPDATE komutlarında kullanılabilmektedir.

SQL sorgularında birden fazla koşul aynı anda kullanılabilir.

Eğer *koşulların tümünü birden sağlayan* kayıtların sorgudan etkilenmesi isteniyorsa, kısıtlamaların arasına **AND** anahtar kelimesi yazılır.

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Hasan	Sert	*NULL*	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

Hasan Sert isimli kişinin adresini Trabzon olarak güncellemek için aşağıdaki sorgu yazılır.

```
UPDATE Adres SET Adres = 'Trabzon' WHERE Ad = 'Hasan' AND Soyad = 'Sert';
```

Yukarıdaki sorgu çalıştırıldığında hem adının hasan olması hem de soyadının sert olması koşulunu sağlayan bütün kayıtlar (bu durumda sadece 1 kayıt) güncellenecektir.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Hasan	Sert	Trabzon	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

Eğer *koşullardan en az bir tanesini sağlayan* kayıtların sorgudan etkilenmesi isteniyorsa, kısıtlamaların arasına **OR** anahtar kelimesi yazılır.

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Siparis		
SiparisNo	Urun	Musteri
1201	Masa	Seren
1245	Sandalye	Seren
1254	Masa	Melina
1270	Koltuk	Can

Sadece Masa veya Sandalye alınmış olan siparişleri listelemek istersek aşağıdaki SQL sorgusu yazılmalıdır.

```
SELECT * FROM Siparis WHERE Urun = 'Masa' OR Urun = 'Sandalye';
```

Yukarıdaki sorgu çalıştırıldığında aşağıdaki kayıtlar listelenecektir.

Siparis		
SiparisNo	Urun	Musteri
1201	Masa	Seren
1245	Sandalye	Seren
1254	Masa	Melina

DİKKAT

AND ve OR anahtar kelimelerinin hem sağında hem de solunda birer koşul yazıyor olması zorunludur.

Günlük konuşma dilinde doğru gibi algılansa da, aşağıdaki sorgu hata verecektir.

```
SELECT * FROM Siparis WHERE Urun = 'Masa' OR 'Sandalye'; -- HATA!
```

Burada hata vermesinin sebebi OR anahtar kelimesinin sağına yazılan ifadenin bir koşul olmamasıdır.

Yazılan bir *koşulu sağlamayan* kayıtların etkilenmesi isteniyorsa, koşulun başına **NOT** anahtar kelimesi yazılır.

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Siparis		
SiparisNo	Urun	Musteri
1201	Masa	Seren
1245	Sandalye	Seren
1254	Masa	Melina
1270	Koltuk	Can

Masa haricindeki bütün siparişleri listelemek için aşağıdaki sorgu yazılır.

```
SELECT * FROM Siparis WHERE NOT Urun = 'MASA';
```

Siparis		
SiparisNo	Urun	Musteri
1245	Sandalye	Seren
1270	Koltuk	Can

Kısıtlamalar yazılırken <, >, =, <>, <=, >= karşılaştırma operatörleri kullanılabilir.

ÖRNEK

Sipariş numarası 1250'den büyük olan bütün kayıtları listelemek için aşağıdaki komut kullanılır.

```
SELECT * FROM Siparis WHERE SiparisNo > 1250;
```

ÖRNEK

Masa haricindeki bütün siparişleri listelemek için (NOT operatörüne alternatif olarak) aşağıdaki gibi bir sorgu da yazılabilir.

```
SELECT * FROM Siparis WHERE Urun <> 'MASA';
```

Koşullarda karşılaştırma operatörleri yerine SQL dilinin operatörleri de kullanılabilir.

IS NULL

Eğer belirtilen alan için herhangi bir değer girilmemişse kayıt etkilenir.

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Ahmet	Taşdemir	İstanbul	0532 1234567	477
Bekir	Ferik	Sakarya	0505 9876543	111
Hasan	Sert	Trabzon	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

Telefon numarası kaydedilmemiş olan tüm kayıtları listelemek için aşağıdaki komut kullanılır.

```
SELECT * FROM Adres WHERE Telefon IS NULL;
```

Adres				
Ad	Soyad	Adres	Telefon	SicilNo
Hasan	Sert	Trabzon	*NULL*	814
Tarık	Sevecen	*NULL*	*NULL*	394

IN (liste)

Eğer kaydın belirtilen alanındaki değer parantez içindeki listede varsa kayıt etkilenir.

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Siparis		
SiparisNo	Urun	Musteri
1201	Masa	Seren
1245	Sandalye	Seren
1254	Masa	Melina
1270	Koltuk	Can

Sandalye ve koltuk siparişlerini görüntülemek için aşağıdaki sorgu kullanılabilir.

```
SELECT * FROM Siparis WHERE Urun IN ('Sandalye', 'Koltuk');
```

Siparis		
SiparisNo	Urun	Musteri
1245	Sandalye	Seren
1270	Koltuk	Can

BETWEEN min AND max

Eğer tanımlı alanın değeri bu değerler arasındaysa kayıt etkilenir. Yazılan değerler (min ve max) aralığa dâhildir.

ÖRNEK

Aşağıdaki gibi bir tablomuz olsun.

Personel				
SicilNo	Ad	Soyad	EvliMi	Maas
477	Ahmet	Taşdemir	0	2450
104	Aysu	Halin	0	1400
124	Abdurrah	Karacadoğa	1	1550

Maaşları 1500 ile 3000 arasında olan tüm personelleri listeleyen SQL sorgusu aşağıdaki gibidir.

```
SELECT * FROM Personel WHERE Maas BETWEEN 1500 AND 3000;
```

Personel				
SicilNo	Ad	Soyad	EvliMi	Maas
477	Ahmet	Taşdemir	0	2450
124	Abdurrah	Karacadoğa	1	1550