

# SORGU OLUŞTURMAK VE ÇEŞİTLERİNİ KULLANMAK-1



- Tabloyu güncellemek
- Select deyiminin yazım kuralları
- Sütunların sınırlandırılması
- Satırların sınırlandırılması
- Sıralama işlemleri

## İÇİNDEKİLER



- Bu üniteyi çalıştıktan sonra;
  - Bir tabloya sorgu ile sütun ekleyip güncelleyebilecek,
  - Basit bir select cümlesinin yazımını ve bir tablodaki verilerin select deyimi kullanılarak listelenmesini kavrayabilecek,
  - Tablodan istenilen sütunların ve satırların görüntülenmesini sağlayabilecek,
  - Karşılaştırmada kullanılan bazı operatörleri ve fonksiyonları öğrenebilecek,
  - Null değerler üzerinde sorgulama yapabilecek,
  - Tablodaki verilerin sıralanarak listelenmesi hakkında bilgi sahibi

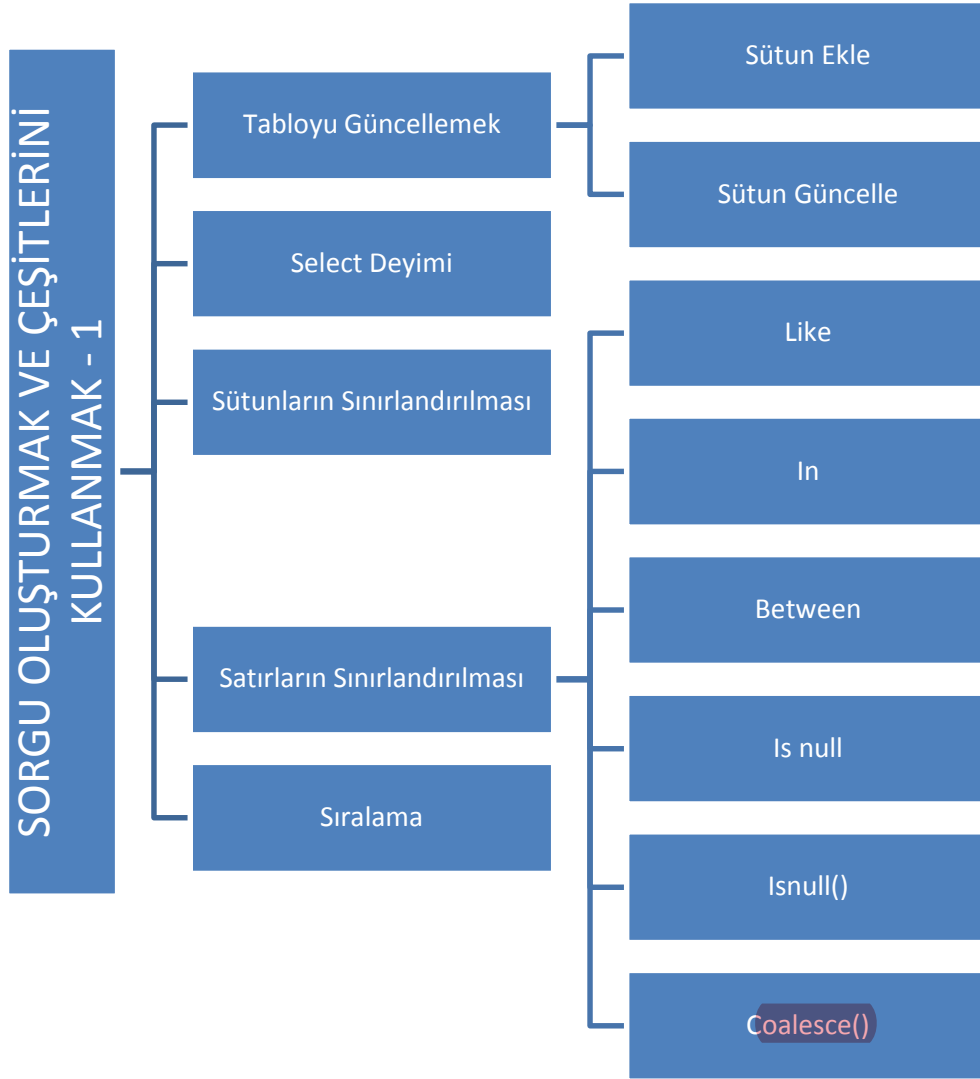
## HEDEFLER



**Atatürk Üniversitesi**  
Açıköğretim Fakültesi

**VERİ TABANI  
YÖNETİM SİSTEMLERİ**  
Dr. Öğr. Üyesi Sinan KUL

**ÜNİTE**  
**6**



## GİRİŞ

Önceki ünite de SQL dilinin genel yapısından bahsedilmişti, T-SQL sorguları ile veri tabanının ve tabloların nasıl oluşturulacağı ve mevcut tabloların nasıl ortadan kaldırılacağı gösterilmişti.

Bu ünite de ise tablodaki sütun özelliklerinin değiştirilmesi ve tabloya yeni sütun eklenmesi için gerekli TSQL kodları gösterilecektir. Bu üniteyle birlikte ayrıca en sık kullanılan SQL deyimi olan “Select” deyiminin yazım kuralları gösterilerek SQL ile sorgu oluşturmanın eğlenceli dünyasına giriş yapılacaktır.

“Select” deyimi kullanılarak tablolardan satır ve sütunlar halinde verilerin çekilerek nasıl esnek bir biçimde görüntülenebileceği öğretilecektir. Bu bağlamda tablodaki bazı sütunların istenen sırada görüntülenebileceği sütunların sınırlandırılması konu başlığında incelenirken tablodaki belirli koşula veya koşullara uyan bazı kayıtların filtrelenmesi ise satırların sınırlandırılması konu başlığında incelenecektir. Satırların sınırlandırılmasında kullanılan koşullar, ilgili sütunlar ve değerlerin karşılaştırılmasını sağlayan operatörler ile oluşturulurken birden fazla koşul ise mantıksal operatörler ile birbirine bağlanmaktadır. Dolayısıyla bu ünite de karşılaştırma operatörleri ve mantıksal operatörler (And, Or ve Not) tanıtıldıktan sonra kullanımlarına örnekler verilecektir.

SQL’de diğer dillerde pek karşılığı bulunmayan karşılaştırma operatörleri ise ayrı alt başlıklar ile incelenecektir. Metinsel ifadeler içinde joker karakterler kullanarak arama yapılmasını sağlayan “like” operatörü, aralık sorgulamada kullanılan “between” operatörü, liste içinden arama yapmayı sağlayan “in” Operatörü ve herhangi bir hücreye değer girilip girilmediğini kontrol eden “is null” operatörü bu kapsamda incelenecektir. Null değerlerle çalışmayı kolaylaştıran Isnull ve Coalesce fonksiyonları ise kapsamın biraz dışında olmasına rağmen konuyla irtibatlı olması dolayısıyla üniteye dâhil edildi.

Bu ünite de son olarak verilerin belirli alanlara göre artan veya azalan olarak sıralanması ve sıralama sonrası en üstte kalan birkaç kaydın görüntülenebilmesi için sorgu kodlarının nasıl yazılacağı gösterilecektir.

Kısacası veri tabanındaki tablolardan değer okurken aranan değer in koordinatını tam olarak verebilmek ve verileri sıralı olarak listelemeyi öğrenmek için bu üniteyi dikkatlice okumanız gerekmektedir. Yeri gelmişken belirtmekte fayda görüyorum, SQL sorguları yazabilmek için SQL yazım kurallarını bilmek yetmez, sorgu yazabilme mantığının geliştirilebilmesi için bol uygulama yapılması gerekmektedir.

## TABLOYU GÜNCELLEMEK

Tablo oluşturulduktan sonra tablonun yapısında (şemasında) değişiklik yapılabilir. Böylelikle tabloya yeni sütunlar eklenebileceği gibi tablodaki mevcut sütunların isimleri, veri türleri ve kısıtları (constraint) güncellenebilmektedir. Bu bağlamda tablo yapısının güncellenebilmesi için SQL’de kullanılan komut “Alter Table” komutudur. Herhangi bir tabloya sütun eklemek için

mesela “Alter Table” (tabloyu değiştir) deyimi ile birlikte “Add” (ekle) deyimi kullanılmaktadır (Adar, 2012).

### Sütun Ekleme

Herhangi bir tabloya sütun eklemek için “Alter Table” (tabloyu değiştir) deyimi ile birlikte “Add” (ekle) deyimi kullanılmaktadır. “Alter Table” deyiminden sonra yapısı güncellenecek tablo adı yazılır; “Add” deyiminden sonra ise eklenecek sütun adı ve veri türü belirtilir (Adar, 2012). Cümlelerin sonunda ayrıca zorunlu olmamakla birlikte ilgili sütunun “Null” geçilebileceği veya geçilemeyeceğini belirten bir ifade yani bir kısıt (constraint) bulunabilir. Bu ifade “Null” ise sütun “Null” değer içerebilir demektir. “Not Null” ise ilgili sütunun “Null” geçilemeyeceğini ve mutlaka değer girilmesi gerekliliğini belirtmektedir. Bu ifadenin kullanılmadığı durumda ise kısıtlamanın bulunmadığı yani ilgili sütunun “Null” geçilebileceği anlaşılır.

#### Söz Dizimi:

```
ALTER TABLE Tablo_Adı ADD Alan_Adı [Alan_Türü] [NULL|NOT NULL]
```

Örnek olarak “Birey” tablosuna bir alan ekleyelim:



Örnek

```
ALTER TABLE Birey ADD DogumYeri VARCHAR(50) NULL
```

Yukarıdaki SQL kodu ile “Birey” tablosuna “DogumYeri” adında ve “varchar” türünde ve 50 karakter uzunluğunda bir sütun eklenmektedir ve “null” (boş) değer içerebileceği belirtilmektedir. “Birey” tablo şemasının son hâli aşağıdaki gibidir:

	Column Name	Data Type	Allow Nulls
🔑	ID	int	<input type="checkbox"/>
	TcNo	varchar(11)	<input checked="" type="checkbox"/>
	Adi	varchar(50)	<input checked="" type="checkbox"/>
	Soyadi	varchar(50)	<input checked="" type="checkbox"/>
	AnneAdi	varchar(50)	<input checked="" type="checkbox"/>
	BabaAdi	varchar(50)	<input checked="" type="checkbox"/>
	Cinsiyeti	varchar(5)	<input checked="" type="checkbox"/>
	DogumYeri	varchar(50)	<input checked="" type="checkbox"/>

Şekil 6.1. Birey Tablosunun “Dogumyeri” Alanı Eklenmiş Hali

Şimdi de bir tabloya bir SQL cümlesi ile birden fazla sütun eklenmesini görelim. Örnek olması için “Birey” tablosuna “Varchar” türünde ve 50 karakter uzunluğunda “DoğumYeri” alanı ve “Date” türünde doğum tarihi sütunlarını ekleyelim.



Null olup olmayacağı belirtilmezse, null kabul eder.

Örnek

- ALTER TABLE Birey ADD ( DogumYeri VARCHAR(50) NULL,
- DogumTarihi DATE )

Yukarıdaki SQL kodunda da görüleceği üzere “Add” deyiminden sonra virgüller ile ayrılmış kolon tanımlamaları parantezler içine alınmaktadır. Örnekte ayrıca “DogumTarihi” sütunu tanımlanırken “Null” olup olmayacağı belirtilmediği halde Şekil 6.2.de de görüleceği üzere varsayılan olarak “Null” değer içerebilir olarak yorumlanmıştır.

	Column Name	Data Type	Allow Nulls
🔑	ID	int	<input type="checkbox"/>
	TcNo	varchar(11)	<input checked="" type="checkbox"/>
	Adi	varchar(50)	<input checked="" type="checkbox"/>
	Soyadi	varchar(50)	<input checked="" type="checkbox"/>
	AnneAdi	varchar(50)	<input checked="" type="checkbox"/>
	BabaAdi	varchar(50)	<input checked="" type="checkbox"/>
	Cinsiyeti	varchar(5)	<input checked="" type="checkbox"/>
	DogumYeri	varchar(50)	<input checked="" type="checkbox"/>
	DogumTarihi	date	<input checked="" type="checkbox"/>

Şekil 6.2. Birey Tablosunun “Dogumyeri” Ve “Dogumtarihi” Alanları Eklenmiş Hâli

## Sütun Güncellemek

Veri tabanı tablosu oluşturulduktan sonra tablodaki sütunların veri tiplerinin ve/veya kısıtların daha sonra güncellenmesi gerekebilmektedir. Tablodaki herhangi bir sütunun veri tipi ve constraint özellikleri değiştirilirken “Alter Column” deyimini kullanılmaktadır. Alter Column deyiminden sonra özelliği değiştirilecek olan sütun adı belirtilir ve sonrasında yeni veri türü ve “Null” olup olmama durumu belirtilir.

### Söz Dizimi:

```
ALTER TABLE Tablo_Adi ALTER COLUMN Alan_Adi [Alan_Türü] [NULL|NOT NULL]
```

Örnek olarak “Birey” tablosundaki “TcNo” alanının veri tipini varchar (11) olarak değiştirelim ve “Null” değer içermeyeceğini belirtelim.



Tablonun sütunları değiştirilirken “Alter Column” deyimini kullanılır.

Örnek

- ALTER TABLE Birey ALTER COLUMN TcNo VARCHAR(11) NOT NULL

Birey tablosunun son hali Şekil 6.3.teki gibi olmaktadır.

	Column Name	Data Type	Allow Nulls
🔑	ID	int	<input type="checkbox"/>
	TcNo	varchar(11)	<input type="checkbox"/>
	Adi	varchar(50)	<input checked="" type="checkbox"/>
	Soyadi	varchar(50)	<input checked="" type="checkbox"/>
	Cinsiyet	int	<input checked="" type="checkbox"/>
	AnneAdi	varchar(50)	<input checked="" type="checkbox"/>
	BabaAdi	varchar(50)	<input checked="" type="checkbox"/>
	DogumYeri	varchar(50)	<input checked="" type="checkbox"/>
	DogumTarihi	date	<input checked="" type="checkbox"/>

Şekil 6.3. Birey Tablosunun Son Hâli

## SELECT DEYİMİNİN YAZIM KURALLARI

“Select” deyimi, veri tabanında bulunan veri kaynaklarından (tablo ve görünüm gibi) verileri sorgulayarak seçmek ve listelemek için kullanılmaktadır. “Select” deyimi ile ilgili tablodaki veriler sadece okunmaktadır; veriler üzerinde herhangi bir değişiklik ( ekleme ve silme gibi ) yapılmamaktadır.

En basit kullanımıyla “Select” deyimi şu şekilde yazılabilmektedir:

*Söz Dizimi:*

```
SELECT Select_Listesi FROM Tablo_Listesi
```

Select deyiminden hemen sonra kullanılan select listesi ile görüntülenmek istenen sütun adları yazılırken “From” ifadesinden sonra kullanılan tablo listesinde verilerin bulunduğu tablo adları yer almaktadır. Birden fazla sütun ve tablo adı birlikte kullanıldığında ise sütun adları ve tablo adları arasında virgül konulmaktadır.

Tabloda yer alan bütün sütun adlarının listelenmesi istenirse sütun adlarını tek tek yazmak yerine, “\*” karakteri kullanılabilir. Bu kullanım ile tablo isimlerinin yazılış sırasına ve sütun adlarının tablolardaki yer alış sırasına göre listelendiğini belirtmek gerekir. Ayrıca ilgili tablodaki satırlar tablodaki kaydediliş sırasına göre listelenmektedir. Örnek olarak “Birey” tablosundaki kayıtları listeleyelim.



Bütün sütun adları yerine sadece “\*” karakteri kullanılır.

Örnek

```
•SELECT * FROM Birey
```

Sorgu cümlesi çalıştırıldığında “Results” (sonuçlar) sekmesinde “Birey” tablosundaki kayıtlar listelenmektedir (Şekil 6.4). Ekranın sağ alt köşesinde sorgu ile kaç satır döndüğü bilgisi, onun hemen solunda sorgu süresi (mili saniye cinsinden) görüntülenmektedir.

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti	
1	2	12312312312	Hidayet	ÇÖLKUSU	Ayşe	Yusuf	Erkek
2	3	35462349764	Taha	BAYRAM	Fatma	Ali	Erkek
3	4	79854631320	Ayşe	YILDIZ	Hayriye	Cihan	Kız

Şekil 6.4. Birey Tablosundaki Kayıtların Listelenmesi

## SÜTUNLARIN SINIRLANDIRILMASI

Tablodaki kayıtlar listelenirken sadece belirli sütun adlarının görüntülenebilmesi için “Select” ifadesinden sonra ilgili sütun adlarının yazılması gerektiğini söylemiştik. Örnek olarak “Birey” tablosunda yer alan bireylerin adlarını listeleyelim. Bunun için “Select” deyiminden sonra ve “From” deyiminden önce sadece “Adi” yazılmalıdır.

Örnek

•SELECT Adi FROM Birey

Şekil 6.5.te de görüleceği üzere “Results” sekmesinde sadece “Adi” sütunu görüntülenmektedir.

Adi	
1	Hidayet
2	Taha
3	Ayşe
4	Nazli
5	Erkan
6	Ahmet
7	Mustafa
8	Selmani
9	Samet
10	Nazife

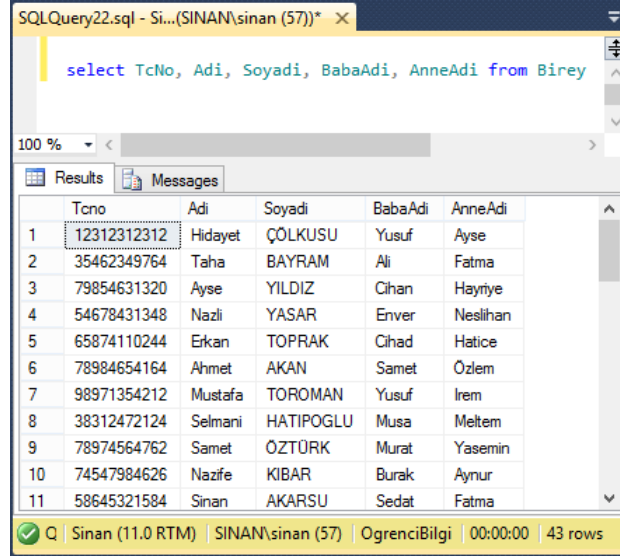
Şekil 6.5. Birey Adlarının Listelenmesi

Sütun adları yazıldığı sırada listelenmektedir.

Şimdi de “Birey” tablosundaki kayıtları listelerken TC kimlik numarası, adı, soyadı, baba adı ve anne adı alanları ile sorguyu sınırlayalım. “Select” deyiminden sonra ilgili sütun adlarının aralarına virgül konarak kullanılmaktadır.

Örnek

- SELECT TcNo, Adi, Soyadi, BabaAdi, AnneAdi FROM Birey



	Tcno	Adi	Soyadi	BabaAdi	AnneAdi
1	12312312312	Hidayet	ÇÖLKUSU	Yusuf	Ayşe
2	35462349764	Taha	BAYRAM	Ali	Fatma
3	79854631320	Ayşe	YILDIZ	Cihan	Hayriye
4	54678431348	Nazli	YASAR	Enver	Neslihan
5	65874110244	Erkan	TOPRAK	Cihad	Hatice
6	78984654164	Ahmet	AKAN	Samet	Özlem
7	98971354212	Mustafa	TOROMAN	Yusuf	Irem
8	38312472124	Selmani	HATIPOGLU	Musa	Meltem
9	78974564762	Samet	ÖZTÜRK	Murat	Yasemin
10	74547984626	Nazife	KIBAR	Burak	Aynur
11	58645321584	Sinan	AKARSU	Sedat	Fatma

Şekil 6.6. Birey Tablosundan Bazı Bilgilerin Listelenmesi

Select ifadesindeki sütun isimlerinin yazıldığı sırada ve kayıtların tablodaki fiziksel konumlarına göre listendiğini görebilirsiniz.

## SATIRLARIN SINIRLANDIRILMASI

SQL sorgu sonuçlarının sınırlandırılarak ilgili bazı kayıtların listelenmesi için özel bir koşul belirtilebilmektedir. Böylece sadece koşula uyan kayıtlar listelenecektir. Bu bağlamda, satırları filtreleyen koşulları belirtmek için “WHERE” operatöründen faydalanılmaktadır (Gözüdeli, 2010).

*Söz Dizimi:*

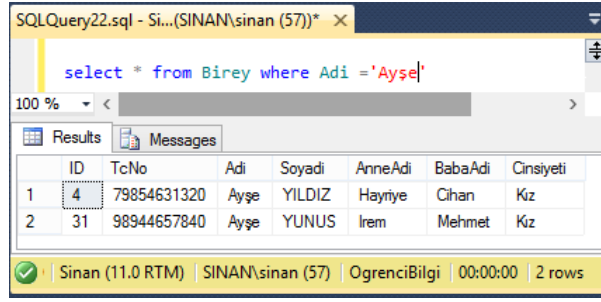
SELECT *Select\_Listesi* FROM *Tablo\_Listesi* WHERE *Koşullar*

Örnek

- SELECT \* FROM Birey WHERE Adi ='Ayşe'

Veriler satır bazında filtrelenirken “Where” deyimini kullanılır.





SQLQuery22.sql - Si...(SINAN\sinan (57))\*

```
select * from Biyey where Adi = 'Ayşe'
```

100 %

Results Messages

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti	
1	4	79854631320	Ayşe	YILDIZ	Hayriye	Cihan	Kız
2	31	98944657840	Ayşe	YUNUS	Irem	Mehmet	Kız

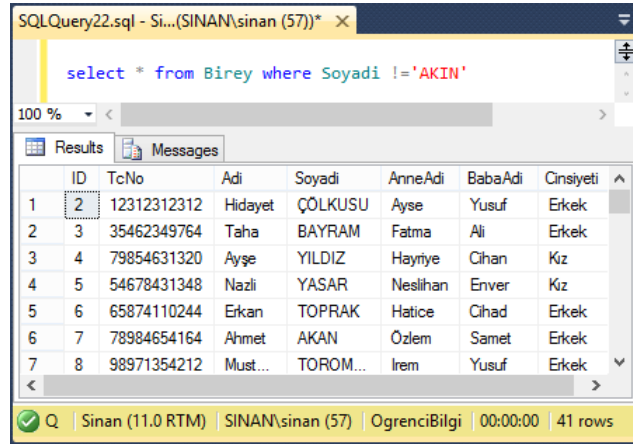
Sinan (11.0 RTM) SINAN\sinan (57) OgrenciBilgi 00:00:00 2 rows

Şekil 6.7. Adı “Ayşe” Olanların Listelenmesi (2 Kayıt)

Şekil 6.7.deki örnek ile adı “Ayşe” olanlar listelendi. Sütun adı ile değer arasında “=” (eşittir) operatörünün kullanıldığını gördünüz. Şimdi de soyadı “AKIN” olmayan bireyleri listeleyelim:

Örnek

•SELECT \* FROM Biyey WHERE Soyadi !='AKIN'



SQLQuery22.sql - Si...(SINAN\sinan (57))\*

```
select * from Biyey where Soyadi != 'AKIN'
```

100 %

Results Messages

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti	
1	2	12312312312	Hidayet	ÇÖLKUSU	Ayşe	Yusuf	Erkek
2	3	35462349764	Taha	BAYRAM	Fatma	Ali	Erkek
3	4	79854631320	Ayşe	YILDIZ	Hayriye	Cihan	Kız
4	5	54678431348	Nazli	YASAR	Neslihan	Enver	Kız
5	6	65874110244	Erkan	TOPRAK	Hatice	Cihad	Erkek
6	7	78984654164	Ahmet	AKAN	Özlem	Samet	Erkek
7	8	98971354212	Must...	TOROM...	Irem	Yusuf	Erkek

Sinan (11.0 RTM) SINAN\sinan (57) OgrenciBilgi 00:00:00 41 rows

Şekil 6.8. Soyadı “AKIN” Olmayanların Listelenmesi. (41 Kayıt)

Şekil 6.8.de eşit olmama durumunu ifade etmek için sütun adı ile değer arasında “!=" operatörü kullanıldı. Koşul tanımında kullanılan diğer tüm karşılaştırma operatörleri aşağıdaki tabloda (Tablo 6.1) sunulmuştur. Tabloda eşit olmama durumu için iki farklı operatör kullanılmaktadır. Uygulamada bunlar arasında bir fark yoktur. Benzer bir biçimde “>=” (büyük eşit) operatörü, “!<” (küçük değil) operatörüyle eşdeğer; “<=” (küçük eşit) operatörü ise “!>” (büyük değil) operatörüyle eşdeğer kullanıma sahiptir.

**Tablo 6.1.** Karşılaştırma operatörleri

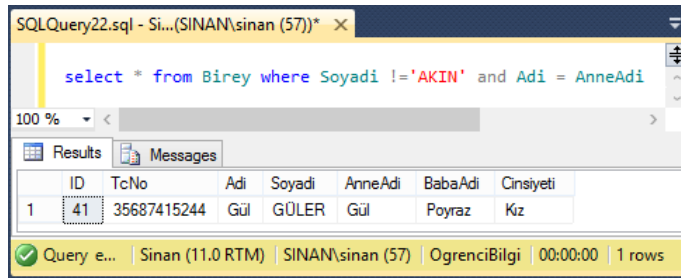
Operatör	Anlamı
=	Eşit
!=	Eşit değil
<>	Eşit değil
>	Büyük
!>	Büyük değil
<	Küçük
!<	Küçük değil
>=	Büyük eşit
<=	Küçük eşit
LIKE	Karakter türde karşılaştırma yapar

Operatörler performansları bakımından kıyaslandığında en performanslı kullanıma sahip olan operatör “=” (eşittir) operatörüdür. Kullanıldığı sorguyu en yavaş sonuçlandıran operatör ise “!=”, “<>” (eşit değil) operatörleridir. Karşılaştırma operatörlerinin performanslarıyla ilgili değinilecek son husus bu operatörlerin sayısal değerlerle daha hızlı çalışmasıdır.

Satırları sınırlandırırken birden fazla koşul ifadesi kullanılacaksa bunlar arasına ya “And” (ve) veya “Or” (veya) operatörünün kullanılması beklenmektedir. “And” operatörü kullanıldığında iki koşulun aynı anda doğru olması durumu aranırken, “Or” operatöründe koşullardan sadece birinin doğru olması sonucun doğru olması için yeterlidir.

Örnek

- SELECT \* FROM Birey WHERE Soyadi !='AKIN' AND Adi=AnneAdi

**Şekil 6.9.** Soyadı “AKIN” Olmayan Ve Adı Annesinin Adı İle Aynı Olanların Listelenmesi.

Şekil 6.9.daki örnekte iki şart ifadesi “AND” (ve) operatörüyle birleştirildi. Böylece “birey” tablosundaki kayıtlar içinde soyadı “AKIN” olmayan ve adı annesinin adına eşit olan kayıtlar listlenmektedir. “Birey” tablosunda bu koşullara uyan bir kayıt bulunmaktadır.



“Like” operatörü karakter türündeki değerler için kullanılır. Sayısal veya tarihsel veriler için kullanılamaz.

## Like Operatörü

Karşılaştırma operatörlerinden biri olan “Like” operatörü, diğer

operatörlerden farklı olarak sadece karakter türdeki değerler içinde arama yapmak için geliştirilmiştir. Özellikle aranan karakter ifadenin ne olduğu tam olarak bilinmediği durumlarda “=” ifadesi yerine “Like” operatörü kullanmak gerekmektedir.

**Tablo 6.2.** Joker Karakterleri

Joker Karakter	Anlamı
%	Birden fazla harf ya da rakamın yerini tutar.
_	Bir tek harf veya rakamın yerini tutar.
[Harf]	Herhangi bir harf yerine gelebilecek harfleri belirtir.
[^Harf]	Herhangi bir harf yerine gelemeyecek harfleri belirtir.
[A-Z]	A ile Z arasındaki harfleri belirtir.

Örnek olarak ismi “K” harfi ile başlayan bireyleri listeleyelim:



% karakteri birden fazla karakterin yerini tutar.



Örnek

```
•SELECT * FROM Birey WHERE Adi LIKE 'K%'
```



Örnek

```
•SELECT * FROM Birey WHERE Adi NOT LIKE '_e%'
```



“\_” karakteri sadece bir karakterin yerini tutar.

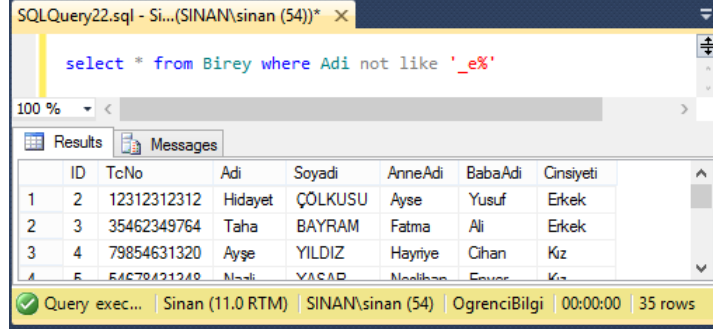
The screenshot shows a SQL query window with the following query: `select * from Birey where Adi like 'K%'`. The results table displays the following data:

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
1	29	Kutlu	AKIN	Özlem	Nihat	Erkek
2	35	Kemal	KUTAY	Zehra	Azrail	Erkek
3	45	Kemal	ORM...	Ümit	Necati	Erkek

**Şekil 6.10.** İsmi “K” Harfi İle Başlayan Bireyler (3 Kayıt)

Şekil 6.10’da da görüldüğü gibi ismi “K” ile başlayanları ifade etmek için tek tırnaklar arasında “K” harfinden sonra “%” karakteri kullanılmıştır. Yüzde karakteri çok sayıda harfi veya rakamı temsil edebildiği gibi boşluğu da temsil edebilmektedir. Yani ilgili konumda herhangi bir karakter bulunmaya da bilir.

Verilen değer ile eşleşme durumunu kontrol etmek için de “Like” operatörüyle birlikte “Not” operatöründen de faydalanılmaktadır. Örnek olarak adının ikinci harfi “e” olmayan bireyleri listeleyelim:



The screenshot shows a SQL query window with the following query: `select * from Birey where Adi not like '_e%'`. The results table is as follows:

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti	
1	2	12312312312	Hidayet	ÇÖLKUSU	Ayşe	Yusuf	Erkek
2	3	35462349764	Taha	BAYRAM	Fatma	Ali	Erkek
3	4	79854631320	Ayşe	YILDIZ	Hayriye	Cihan	Kız
4	5	54678421248	Nesli	YASAR	Melike	Evren	Kız

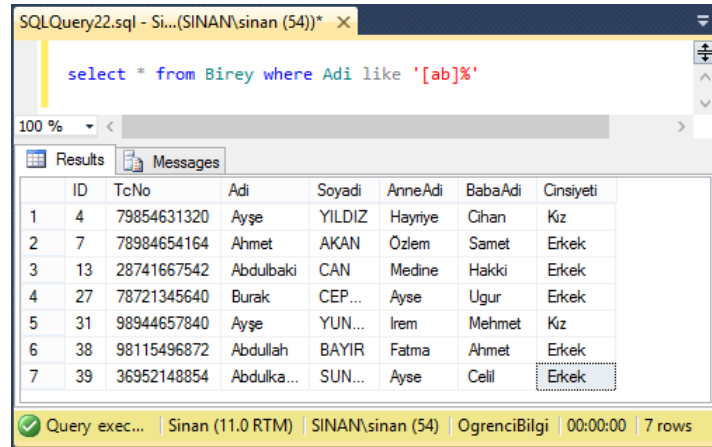
Şekil 6.11. İsmi'nin ikinci harfi “e” olmayan bireyler (35 kayıt)

Şekil 6.11.deki örnekte de birinci harfi temsil için “\_” (alttire) karakteri kullanıldı. Alt tirenin, yüzde karakterinden ilk farkı, tek karakteri temsil etmesidir. İkinci farkı ise boşluk yerine kullanılmamasıdır. Yani alttire kullanılan konumda kesinlikle bir karakterin (harf, sayı vs.) bulunması beklenmektedir.

Şimdi de adının baş harfi “a” veya “b” olan bireyleri listeleyelim. Bunun için köşeli parantezler arasında “a” ve “b” harflerinin birlikte kullanıldığına dikkat edin.

Örnek

•SELECT \* FROM Birey WHERE Adi LIKE '[ab]%'



The screenshot shows a SQL query window with the following query: `select * from Birey where Adi like '[ab]%'`. The results table is as follows:

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti	
1	4	79854631320	Ayşe	YILDIZ	Hayriye	Cihan	Kız
2	7	78984654164	Ahmet	AKAN	Özlem	Samet	Erkek
3	13	28741667542	Abdubaki	CAN	Medine	Hakki	Erkek
4	27	78721345640	Burak	CEP...	Ayşe	Ugur	Erkek
5	31	98944657840	Ayşe	YUN...	Irem	Mehmet	Kız
6	38	98115496872	Abdullah	BAYIR	Fatma	Ahmet	Erkek
7	39	36952148854	Abdulka...	SUN...	Ayşe	Celil	Erkek

Şekil 6.12. Adının Baş Harfi “A” Veya “B” Olan Bireyler (7 Kayıt)

Şimdi de adının üçüncü harfi “k”, “l”, “m” veya “n” olan bireyleri listeleyelim. Bunun için de köşeli parantezler içerisinde “k-n” kullanıldığına dikkat edin (şekil 6.13). Aralığın ilk harfi ile son harfi arasında “-” (tire) kullanıldı. Adının ilk iki harfini

temsil için iki adet “\_” (alt tire ) kullanıldı. Metin içindeki % karakteri ise ismin 3’ten fazla uzunlukta olabilmesi durumunu ifade etmektedir.

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti	
1	5	54678431348	Nazli	YASAR	Neslihan	Enver	Kız
2	8	98971354212	Mustafa	TOROMAN	Irem	Yusuf	Erkek
3	11	74547984626	Nazife	KIBAR	Aynur	Burak	Kız
4	17	89846465430	Muhammed	TURKOGLU	Zehra	Saban	Erkek
5	18	56547978410	Leyla	KUTAY	Elif	Yusuf	Kız
6	19	88797466544	Mülayim	AKAR	Seda	Halil	Erkek
7	29	25617885696	Kıralı	AKIN	Özlem	Nihat	Erkek

Şekil 6.13. Adının Üçüncü Harfi “K”, “L”, “M” Veya “N” Olan Bireyler (13 Kayıt)

Şimdi de çift isimli olan yani adı içinde boşluk karakteri bulunan bireyleri listeleyelim. Bunun için yüzde karakterleri arasında boşluk karakteri kullanıldı (Şekil 6.14).

Örnek

•SELECT \* FROM Birey WHERE Adi LIKE '% %'

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti	
1	36	97879864116	Emir Hattab	KUL	Seda	Mustafa	Erkek

Şekil 6.14. Adı İçinde Boşluk Karakteri Bulunan Bireyler (1 Kayıt)

## Between Operatörü

“Between” operatörü, bir aralık içinde yer alan kayıtları sorgulamak için kullanılır. Koşulun “true” (doğru) olabilmesi için kıyaslanan değer, “Between” operatöründen sonra gelen birinci değerden büyük, ikinci değerden küçük veya bu değerlere eşit olması gerekmektedir.

*Söz Dizimi:*

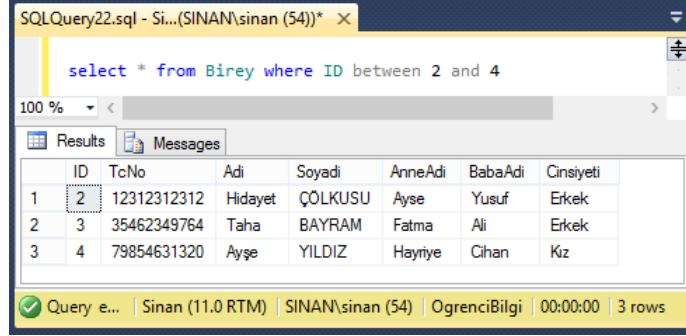
*Sütun\_Adı BETWEEN Alt\_Sınır AND Üst\_Sınır*



Between operatöründen sonra önce küçük değer kullanılmaktadır.

Örnek

- SELECT \* FROM Birey WHERE ID BETWEEN 2 AND 4



ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
2	12312312312	Hidayet	ÇÖLKUSU	Ayşe	Yusuř	Erkek
3	35462349764	Taha	BAYRAM	Fatma	Ali	Erkek
4	79854631320	Ayşe	YILDIZ	Hayriye	Cihan	Kız

Şekil 6.15. ID Deęeri 2 İle 4 Arasında Olan Bireyler (3 Kayıt)

Şekil 6.15.te “Between” operatörüne örnek verilirken “Birey” tablosundaki Id alanı 2 ve 4 arasında olan kayıtlar listelenmektedir.

“Between ... and ...” operatörünün eşlenięi “And” operatörü ile yukarıdaki örnek sorgu ařaęıdaki gibi de yazılabilir:

```
SELECT * FROM Birey WHERE ID >= 2 AND ID <= 4
```

## In Operatörü

In operatörü, bir sütunun birden fazla deęer ile aynı anda karşılaştırılabilmesi için kullanılmaktadır. Yani bir küme içinde arama yapılmaktadır. Karşılařtırmada kullanılan deęerler parantez içinde virgüllerle ayrılmaktadır. “In”, operatörü “Not” operatörü ile kullanıldığında ise sütun deęerinin liste içindeki herhangi bir deęere eřit olmama durumu sorgulanmaktadır (Elbahadır, 2012).

*Söz Dizimi:*

*Sütun\_Adı [NOT] IN (Parametre\_Listesi)*

Örnek

- SELECT \* FROM Birey WHERE ID IN (6,9)



“In” operatörü liste içinde arama yapmak için kullanılır.

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
6	65874110244	Erkan	TOPRAK	Hatice	Cihad	Erkek
9	38312472124	Selmani	HATIPOGLU	Meltem	Musa	Erkek

Şekil 6.16 ID Değeri 6 Olanların Ve ID Değeri 9 Olanların Listelenmesi (2 Kayıt)

Şekil 6.16.daki örnekte “Birey” tablosunda Id’si 6 veya 9 olan kayıtlar listelenmektedir. Aynı sorgu “Or” operatörü marifetiyle aşağıdaki gibi de yazılabilir:

```
SELECT * FROM Birey WHERE ID =6 OR ID=9
```

Aşağıdaki örnekte ise “Not” operatörünün “In” operatörüyle kullanımı gösterilmektedir. “Adi” alanı “Hidayet” ve “Taha” olmayan bireyler listelenmektedir.

**Örnek**

- SELECT \* FROM Birey WHERE Adı NOT IN ('Hidayet','Taha')

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
4	79854631320	Ayşe	YILDIZ	Hayriye	Cihan	Kız
5	54678431348	Nazlı	YASAR	Neslihan	Enver	Kız
6	65874110244	Erkan	TOPRAK	Hatice	Cihad	Erkek

Şekil6.17. Adı “Hidayet” Ve “Taha” Olmayan Kayıtların Listelenmesi (41 Kayıt)

## Is Null Operatörü

Bu operatör, herhangi bir alana veri girilip girilmediğinin kontrolünü sağlamak amacıyla kullanılmaktadır. Veri girilmemiş olan alanın değeri bildiğiniz gibi “Null”dur. Kısacası “is null” operatörüyle “Null” olan değer veya hücre tespit edilebilmektedir. “Null” olmayan hücrenin tespit edilebilmesi için de “Is Not Null” kullanılmaktadır.

*Söz Dizimi:*



“Null” değerler için karşılaştırma operatörü kullanılamaz. Yani “adi=null” denemez.

Sütun\_Adı IS [NOT] NULL

Örnek

•SELECT \* FROM Birey WHERE Cinsiyeti IS NULL

The screenshot shows a SQL query window with the following SQL statement: `select * from Birey where Cinsiyeti is null`. The results grid displays the following data:

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
43	87984156564	Semra	YILDIZ	Serife	Alper	NULL
44	25497812158	Züal	AYAR	Hayriye	Lütfü	NULL

Şekil 6.18. Cinsiyet Bilgisi Girilmeyen Bireyler (2 Kayıt)

## Isnull() Fonksiyonu

Isnull fonksiyonu, herhangi bir değerın null olması durumunda, sorgu sonucundan başka bir değer döndürmek için kullanılır. Bu fonksiyon, iki parametre almaktadır. Birinci parametre "null" olma durumu kontrol edilecek değeri, ikinci parametre ise "null" olma durumunda geri döndürülecek değeri temsil etmektedir.

*Söz Dizimi:*

ISNULL( Sütun\_Adı, Alternatif\_Değer )

Örnek

•SELECT \* FROM Birey WHERE ISNULL(Cinsiyeti,'Kız')='Erkek'

The screenshot shows a SQL query window with the following SQL statement: `select * from Birey where isnull(Cinsiyeti , 'Kız')='Erkek'`. The results grid displays the following data:

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
2	12312312312	Hidayet	ÇOLKUSU	Ayşe	Yusuf	Erkek
3	35462349764	Taha	BAYRAM	Fatma	Ali	Erkek
6	65874110244	Erkan	TOPRAK	Hatice	Cihad	Erkek

Şekil 6.19. Erkek Bireylerin Listelenmesi (29 Kayıt)



Bu örnekte “Birey” tablosunda “Cinsiyeti” alanı “null” olan kayıtlar için değer, “Kız” olarak kabul ediliyor.

### Coalesce() Fonksiyonu

Bu fonksiyon, “Isnull” fonksiyonuna benzemektedir. Farkı, bu fonksiyonun ikiden fazla parametre alabilmesidir. Fonksiyonun aldığı parametreler içinden, soldan sağa doğru “null” olmayan ilk parametre geriye döndürülmektedir. Bütün parametreler “null” değeri içeriyorsa da geriye “null” döndürülmektedir.

*Söz Dizimi:*

COALESCE( Sütun\_Adi, Alternatif\_Değer1, Alternatif\_Değer2 ... )



“Coalesce” fonksiyonu, “Isnull” fonksiyonunun geliştirilmiştir.

Örnek

•SELECT \* FROM Birey WHERE COALESCE(Cinsiyeti,'Kız')='Erkek'

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
1	2	Hidayet	ÇOLKUSU	Ayse	Yusuf	Erkek
2	3	Taha	BAYRAM	Fatma	Ali	Erkek
3	6	Erkan	TOPRAK	Hatice	Cihad	Erkek

Şekil 6.20. Erkek Bireylerin Listelenmesi (29 Kayıt)

### SIRALAMA İŞLEMLERİ

Veri tabanı tablolarından kayıtlar listelenirken tabloda kayıtlı olduğu sıraya göre görüntülendiğini belirtmiştik. Kayıtların bir veya birden fazla alana göre sıralı olarak listelenmesi için de “ORDER BY” deyimi kullanılabilir. Kayıtların ilgili alana göre A’den Z’ye veya küçükten büyüğe doğru sıralanmasında “ASC” (Ascending) deyimi kullanılırken, Z’den A’ya veya büyükten küçüğe sıralamada ise “DESC” (Descending) deyimi kullanılır. Sıralama tipi belirtilmediğinde ise varsayılan olarak “ASC” kabul edilmektedir.

*Söz Dizimi:*

SELECT Select\_Listesi FROM Tablo\_Listesi WHERE Koşullar  
ORDER BY sütun1 [DESC|ASC] [,sütun2 [DESC|ASC], ... ]

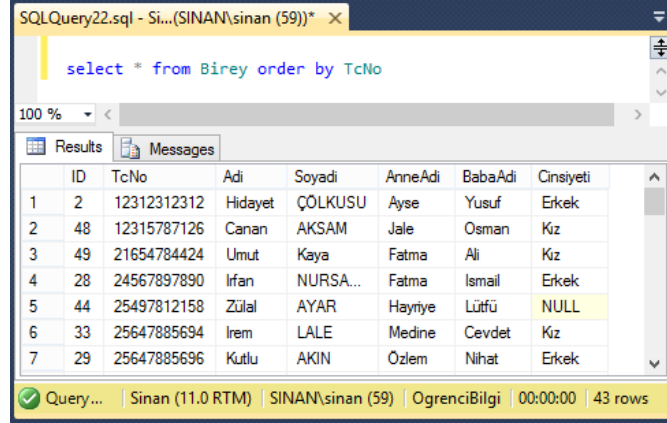


Select ifadesinin yazımında en sonra sıralama işlemi gelmektedir.

Örnek olması için “Birey” tablosundaki kayıtları “TcNo” alanına göre sıralayarak listeleyen sorguyu yazalım. Sıralama tipi belirtilmediği için sıralamanın küçükten büyüğe doğru yapıldığını görebilirsiniz (Şekil 6.21).

Örnek

- SELECT \* FROM Birey ORER BY TcNo



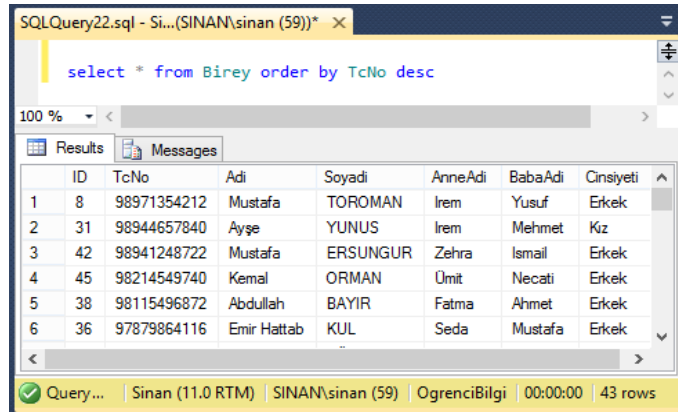
ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
1	2	Hidayet	ÇÖLKUSU	Ayşe	Yusuf	Erkek
2	48	Canan	AKSAM	Jale	Osman	Kız
3	49	Umut	Kaya	Fatma	Ali	Kız
4	28	İrfan	NURSA...	Fatma	İsmail	Erkek
5	44	Zühal	AYAR	Hayriye	Lütfü	NULL
6	33	Irem	LALE	Medine	Cevdet	Kız
7	29	Kutlu	AKIN	Özlem	Nihat	Erkek

Şekil 6.21. Bireylerin TC Kimlik Numaralarına Göre Artan Şekilde Sıralanması (43kayıt)

Şimdi de “Birey” tablosundaki kayıtları “TcNo” alanına göre azalan şekilde sıralayalım.

Örnek

- SELECT \* FROM Birey ORDER BY TcNo DESC



ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
1	8	Mustafa	TOROMAN	Irem	Yusuf	Erkek
2	31	Ayşe	YUNUS	Irem	Mehmet	Kız
3	42	Mustafa	ERSUNGUR	Zehra	İsmail	Erkek
4	45	Kemal	ORMAN	Ümit	Necati	Erkek
5	38	Abdullah	BAYIR	Fatma	Ahmet	Erkek
6	36	Emir Hattab	KUL	Seda	Mustafa	Erkek

Şekil 6.22. Bireylerin TC Kimlik Numaralarına Göre Azalan Şekilde Sıralanması (43kayıt)

Bu defa da birden fazla alana göre sıralamaya örnek verelim. Bireyleri önce adlarına göre artan sonra soyadlarına göre azalan şekilde sıralayalım. “Order by” operatörü ile kayıtlar birden fazla sütuna göre sıralanırken sütun adlarının yazılış sırasına göre sıralanmaktadır. Yani örnekte (Şekil 6.23) “Soyadi, adi” şeklindeki kullanımda, öncelikle “Soyadi” alanına göre sıralamakta daha sonra soyadı aynı olanlar içinde “Adi” alanına göre kayıtları sıralayarak listelemektedir. Dikkat edin bu kod sadece listelemektedir; kayıtların tablodaki sırasını değiştirmemektedir.



“Order by” ve “Top” operatörleri birlikte kullanıldığında en yüksek veya en düşük değerlere sahip kayıtlar listelenebilir.

Örnek

•SELECT \* FROM Birey ORDER BY Adi, Soyadi DESC

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
1	13	28741667542	Abdülbaki	CAN	Medine Hakkı	Erkek
2	39	36952148854	Abdulkadir	SUNAY	Ayşe Celil	Erkek
3	38	98115496872	Abdullah	BAYIR	Fatma Ahmet	Erkek
4	7	78984654164	Ahmet	AKAN	Özlem Samet	Erkek
5	31	98944657840	Ayşe	YUNUS	Irem Mehmet	Kız
6	4	79854631320	Ayşe	YILDIZ	Hayriye Cihan	Kız
7	27	78721345640	Burak	CEPKEN	Ayşe Ugur	Erkek

Şekil 6.23. Bireylerin Adlarına Göre Artan, Soyadlarına Göre Azalan Şekilde Sıralanması

*Top Operatörü:*

“Order by” operatörü, “Top” operatörü ile birlikte kullanıldığında, sorgu sonucu dönen ilk N adet kaydın listelenmesi sağlanmaktadır.

*Söz Dizimi:*

SELECT TOP(n) [PERCENT] select\_listesi

FROM tablo\_listesi

WHERE koşullar

...

ORDER BY sütun1 [DESC|ASC] [,sütun2 [DESC|ASC], ... ]

Örnek

•SELECT TOP 10 PERCENT ID, Adi, Soyadi FROM Birey ORDER BY ID

## Örnek



•SELECT TOP 1 \* FROM Birey ORDER BY id DESC

ID	TcNo	Adi	Soyadi	AnneAdi	BabaAdi	Cinsiyeti
49	21654784424	Umut	Kaya	Fatma	Ali	Kız

Şekil 6.24. ID Değeri En Büyük Olan Kaydın Gösterilmesi(1kayıt)

“Percent” (yüzde) deyiminin de bu iki deyimle eklenmesiyle kayıtların yüzde kaçının listeleneceği belirtilebilmektedir.

ID	Adi	Soyadi
2	Hidayet	ÇÖLKUSU
3	Taha	BAYRAM
4	Ayşe	YILDIZ
5	Nazlı	YASAR
6	Erkan	TOPRAK

Şekil 6.25. ID Değeri En Küçük Olanların %10'unun Listelenmesi (5 Kayıt)

Yukarıdaki örnekte “Birey” tablosundaki kayıtlar “ID” alanına göre sıralanarak kayıtların %10'u listelenmektedir.



## Bireysel Etkinlik

- Bir tablodaki kayıtların sıralanmasında, tarihsel veri içeren kolona göre sıralamanın, tam sayı türünde kolona göre sıralamaya göre performans farkını test ediniz.
- Select deyiminin kullanımında bütün alanların getirilmesi ile yalnızca bir alanın getirilmesinin performansa etkisini test ediniz.
- Koşul tanımlarken "in" operatörü ile "=" operatörünün performans farkını test ediniz.
- "=" operatörünün "Like" operatöründen hızlı çalıştığını test ediniz.

Kayıtların %10'unu getirmek için “Top 10 Percent” ifadesi kullanılabilir.



## Özet

- Herhangi bir tabloya SQL sorguları kullanılarak sütun eklemek için, "Alter Table" (tabloyu değiştir) deyimi ile birlikte "Add" (ekle) deyimi kullanılmaktadır. Tabloya birden fazla sütun birlikte eklenirken sütun tanımları virgüller ile ayrılmış olarak kullanılır. "Null" olup olmayacağı belirtilmeyen bir sütun ise varsayılan olarak "null" değer içerebilir olarak yorumlanmaktadır.
- Tablodaki bir sütunu güncellemek için "Alter Table" deyimi ile birlikte "Alter Column" deyimi kullanılmaktadır.
- "Select" deyimi, veri tabanında bulunan veri kaynaklarından verileri sorgulayarak seçmek için ve listelemek için kullanılmaktadır. Select deyimi ile tablodaki veriler üzerinde herhangi bir değişiklik yapılmamaktadır.
- "Select ... from..." cümleciğinde "Select" deyiminden sonra listelenmesi istenen sütun adları ve "From" deyiminden sonra ise tablo isimleri gelmektedir. Birden fazla sütun ve tablo adı birlikte kullanıldığında ise aralarına virgül konulmaktadır.
- Tablodaki bütün sütun adları listelenmek istendiğinde sütun adlarını tek tek yazmak yerine "\*" karakteri kullanılmaktadır. Bu kullanım ile tablo isimlerinin yazılış sırasına ve sütun adlarının tablolardaki yer alış sırasına göre listelendiğini belirtmek gerekir.
- Select ifadesinden sonra yazılan sütun adları hangi sırada yazıldıysa o sırada ve sadece adı yazılı olan sütunlar listelenmektedir.
- SQL sorgu sonuçlarının sınırlandırılarak ilgili bazı kayıtların listelenmesi için özel bir koşul belirtilebilmektedir. Böylece sadece koşula uyan kayıtlar listelenecektir. Bu bağlamda, satırları filtreleyen koşulları belirtmek için "WHERE" operatöründen faydalanılmaktadır.
- İlgili kayıtların filtrelenebilmesi için kullanılan karşılaştırma operatörleri: eşit, eşit değil, büyük, büyük değil, küçük, küçük değil, büyük eşit, küçük eşit ve like operatörüdür. Birden fazla şartın birleştirilebilmesi için "And" ve "Or" mantıksal operatörlerinden faydalanılmaktadır. "AND" operatörü ile her iki şartın da sağlandığı kayıtlar listelenirken her iki şarttan birinin sağlanmasının yeterli olduğu durumlarda "veya" anlamına gelen "OR" operatörü kullanılmaktadır. "Like" operatörü, karakter türdeki değerler için özelleştirilmiş, karşılaştırma operatörüdür.
- "Like" operatörü ile karşılaştırma yaparken "\_" ve "%" joker karakterleri kullanılabilir.
- "Between .. and.." karşılaştırma operatörü iki değer aralığını tanımlamak için kullanılır.
- "In" operatörü bir liste içindeki değerler ile topluca karşılaştırma yapmak için kullanılır.
- Sütun içeriğinin "Null" olup olmadığının kontrolü için "Is null" ve "Is not null" operatörleri kullanılmaktadır.
- "Isnull" ve "Coalesce" fonksiyonları bir sütun değerinin "null" olması durumunda yerine geçecek değeri tanımlamak için kullanılmaktadır.
- Sorgu sonucunda kayıtları sıralamak için "Order by" operatörü kullanılmaktadır.

## DEĞERLENDİRME SORULARI

1. Aşağıdakilerden hangisi “Birey” tablosuna, tarihsel veri içeren ve “null” değer içermeyen “DogumTarihi” alanını eklemektedir?
  - a) ALTER TABLE Birey ADD DogumTarihi DATETIME NOT NULL
  - b) ALTER TABLE Birey ALTER COLUMN DogumTarihi NOT NULL
  - c) ALTER TABLE Birey ADD DogumTarihi NULL
  - d) ALTER TABLE Birey ALTER COLUMN DogumTarihi DATETIME NULL
  - e) ALTER TABLE Birey ADD DogumTarihi DATETIME NULL
  
2. Aşağıdakilerden hangisi “Kitap” tablosundaki “YazarAdi” alanının uzunluğunu 50 karakter olarak güncellemektedir?
  - a) ALTER TABLE Birey ALTER COLUMN TcNo VARCHAR(11)
  - b) ALTER TABLE Kitap ALTER COLUMN YazarAdi VARCHAR(50)
  - c) SELECT TOP 50 YazarAdi FROM Kitap
  - d) ALTER TABLE Kitap ADD YazarAdi INT(50)
  - e) ALTER TABLE YazarAdi ALTER COLUMN Kitap VARCHAR(11)
  
3. Aşağıdakilerden hangisi adının ikinci harfi “.” olan bireyleri listeler?
  - a) SELECT \* FROM Birey WHERE Adi LIKE ‘\_.%’
  - b) SELECT \* FROM Birey WHERE Adi = ‘.2’
  - c) SELECT \* FROM Birey WHERE Adi LIKE ‘%.%’
  - d) SELECT \* FROM Birey WHERE LIKE Adi ‘.%’
  - e) SELECT \* FROM Birey WHERE Adi LIKE ‘%\_.%’
  
4. Aşağıdakilerden hangisi adı içinde boşluk karakteri olan bireyleri listeler?
  - a) SELECT \* FROM Birey WHERE Adi LIKE ‘ ’
  - b) SELECT \* FROM Birey WHERE Adi = ‘% %’
  - c) SELECT \* FROM Birey WHERE Adi LIKE ‘% %’
  - d) SELECT \* FROM Birey WHERE LIKE Adi ‘ ’
  - e) SELECT \* FROM Birey WHERE Adi LIKE ‘\_ \_’
  
5. Aşağıdakilerden hangisi “SicilNo” bilgisi girilmemiş olan “Personel” kayıtlarını listeler?
  - a) SELECT \* FROM Personel WHERE SicilNo = ‘NULL’
  - b) SELECT \* FROM Personel WHERE ISNULL(SicilNo)
  - c) SELECT \* FROM Personel WHERE SicilNo LIKE NULL
  - d) SELECT \* FROM Personel WHERE SicilNo IS NULL
  - e) SELECT \* FROM Personel WHERE COALESCE (SicilNo, ‘ ’)

6. Aşağıdaki sorgulardan hangisi "Birey" tablosundaki kayıtları önce "Adi" sonra "Soyadi" alanına göre "a" dan "z" ye doğru sıralamamaktadır?
- SELECT \* FROM Birey ORDER BY Adi, Soyadi
  - SELECT \* FROM Birey ORDER BY Adi ASC, Soyadi
  - SELECT \* FROM Birey ORDER BY Adi, Soyadi ASC
  - SELECT \* FROM Birey ORDER BY Adi ASC, Soyadi ASC
  - SELECT \* FROM Birey ORDER BY Adi DESC, Soyadi DESC
7. Aşağıdaki sorgulardan hangisi maaşı en yüksek olan 3 personeli listeler?
- SELECT \* FROM Personel WHERE Maas = MAX(3)
  - SELECT \* FROM Personel GROUP BY MaaS, 3
  - SELECT TOP 3 \* FROM Personel ORDER BY Maas DESC
  - SELECT TOP 3 PERCENT \* FROM Personel ORDER BY Maas
  - SELECT TOP 3 \* FROM Personel ORDER BY Maas
8. Aşağıdaki sorgulardan hangisi "Birey" tablosundaki kayıtları listelerken sadece "Adi" ve "Soyadi" alanlarını görüntülemektedir?
- SELECT \* FROM Birey WHERE Adi=Soyadi
  - SELECT Adi, Soyadi FROM Personel
  - SELECT \* FROM Birey ORDER BY Adi, Soyadi
  - SELECT Adi, Soyadi, \* FROM Birey
  - SELECT Adi, Soyadi FROM Birey
9. Aşağıdaki sorgulardan hangisi sayfa sayısı 400 ile 500 arasında olan kitapları yazar adlarına göre "z" den "a" ya doğru sıralamaktadır?
- SELECT \* FROM Kitap ORDER BY YazarAdi DESC
  - SELECT \* FROM Kitap WHERE SayfaSayisi > 400 ORDER BY YazarAdi
  - SELECT \* FROM Kitap WHERE SayfaSayisi BETWEEN 400 AND 500
  - SELECT \* FROM Kitap WHERE SayfaSayisi BETWEEN 400 AND 500 ORDER BY YazarAdi DESC
  - SELECT \* FROM Kitap WHERE SayfaSayisi >400 AND SayfaSayisi<500
10. Aşağıdaki sorgulardan hangisi yanlış yazılmıştır?
- SELECT \* FROM Birey ORDER BY Adi
  - SELECT \* FROM Birey ORDER BY Adi ASC, Soyadi DESC
  - SELECT \* FROM Birey ORDER BY Adi WHERE Adi='Ali'
  - SELECT \* FROM Birey WHERE Adi= 'Ali' ORDER BY Adi
  - SELECT \* FROM Birey WHERE Adi LIKE '[az]%'

**Cevap Anahtarı**

1.a, 2.b, 3.a, 4.c, 5.d, 6.e, 7.c, 8.e, 9.d, 10.c

## **YARARLANILAN KAYNAKLAR**

- Adar, İ., 2012. SQL Server 2012 Programlama ve Yönetim. İstanbul, 720  
Gözüdeli, Y., 2010. Yazılımcılar için SQL SERVER 2008 R2 ve  
Veritabanı  
Elbahadır, H., 2012. T-SQL SQL SERVER 2012. İstanbul, 294  
Programlama. Ankara, 671