



VERİTABANI YÖNETİM SİSTEMLERİ

Dr. Öğr. üyesi
Ender Şahinaslan

BÖLÜM -6-

SQL YAPISAL SORGULAMA DİLİ



GENEL BAKIŞ...

- SQL yapısal sorgulama diline giriş,
- Temel veritabanı kavramları,
- Temel veri türleri,
- SELECT deyiminin temel yapısı,
- Aritmetik ifadelerin sorgularda kullanımı,
- Aynı değere sahip satırların denetlenmesi,
- Karşılaştırma işleçleri,
- Mantıksal işleçlerin kullanımı,
- İşleçlerin işlem sırası,
- Verinin sıralanması.

6.1. GİRİŞ

- SQL (Structured Query Language) yapısal sorgulama sorgulama dilidir.
- Veri tabanları ve tablolar oluşturmayı, oluşturulan tablolara kayıt ekleme, silme, güncelleme ve listeleme işlemlerinin yapılabilmesini sağlayan bir alt sorgulama dilidir.
- SQL'in kendine özgü deyimleri ve kuralları vardır.
 - Hemen hemen tüm ilişkisel veri tabanı yönetim sistemlerinin ortak dili olarak kabul edilmiştir.

6.1. GİRİŞ (DEVAM...)

- Verileri sorgulamak için SQL kullanan başlıca veri tabanı yönetim sistemleri şunlardır:
 - Oracle,
 - Microsoft SQL Server,
 - Sysbase,
 - DB2,
 - MySQL,
 - Microsoft Access.

6.2. TEMEL VERİTABANI KAVRAMLARI (TEKRAR)

Veritabanı, herhangi bir alanda belli bir amaca yönelik olarak düzenlenmiş verilerden oluşan yapı olarak tanımlanabilir.

- Bir firmanın ürün bilgilerinin, bir kitapevinin kitap stok bilgilerinin vb. saklandığı yapılar örnek olarak verilebilir.

Bir veritabanı tablosu, verilerin daha düzenli olarak saklanması için satırlar ve sütunlardan oluşmaktadır. **Bütün kayıtlar satırlarda yer almaktadır. Her alandaki kayıtlar birbirinden farklıdır ve verilerin birbirlerine karışmasını engellemektedir.**

- Örneğin öğrenci bilgilerinin tutulduğu bir tabloda bir varlığa ait kayıtlar satırlarda yer almaktadır. Sütunlar ise ilgili varlığa ait nitelikleri tutmaktadır. Bu nitelikler Adı, Soyadı vb. niteliklerden oluşmaktadır.

6.2. TEMEL VERİTABANI KAVRAMLARI (TEKRAR)

Bir **veritabanı tasarlanmadan önce** bir tablo içerisinde hangi niteliklerin yani sütunların bulunması gerektiği belirlenmelidir. Bu en önemli unsurların başında gelmektedir.

Daha sonraki adımlarda tablolar arasında bir ilişki bulunabildiğinden bu öğeler doğru ve dikkatli belirlenmelidir. Dolayısı ile **bir veritabanının en önemli ögesi tablolar olmaktadır**. Tablolar arasında ilişkilerin belirlenmesi varlık ilişki modelinin en önemli adımlarından biridir.

6.2. TEMEL VERİTABANI KAVRAMLARI (TEKRAR)

Aşağıdaki şekilde, bir tabloya ait satır, sütun ve kayıtlar görülmektedir.

urunKod	Barkod	bolgeKod	markaKod	urunAd	bayiFiyat	listeFiyat	Indirim	KDVoran
3505	NULL	NULL	NULL	Staj Defteri	NULL	2500	NULL	NULL
3391	3CP015098	NULL	1	HomeConnect P...	20	26,0830625999...	0	0,18
3026	KBS-720-F	NULL	2	PS/2 F TR Klavy...	4,3	5,5982598	0	0,18
3027	KBS-720	NULL	2	PS/2 Q TR Anti-...	4,5	5,8856448	0	0,18
3028	KBS-21	NULL	2	PS/2 Q Anti-Rsi ...	7,5	9,7940808	0	0,18
3029	KBS-8	NULL	2	PS/2 Q TR Multi...	7,5	9,7940808	0	0,18
3030	KBS-827	NULL	2	PS/2 Q TR Multi...	23	29,9914986	0	0,18
3031	KB-827	NULL	2	PS/2 Q TR Multi...	23	29,9914986	0	0,18
3032	KBA-527RA	NULL	2	PS/2 Q TR Klavy...	25,5	33,2676875999...	0	0,18

Satır


Sütun

6.3. TEMEL VERİ TÜRLERİ

MS SQL Server' a ait veri türleri şöyledir.

char (uzunluk): Sayısal olmayan yani alfabetik verileri tanımlamak için kullanılmaktadır. Bir sütunda saklanacak olan verinin boyutu uzunluk parametresi ile belirtilir.


Örnek: 'products' tablosunda, «productName» alanı aşağıdaki şekilde tanımlanabilir.

	Column Name	Data Type	Allow Nulls
	productID	int	<input type="checkbox"/>
	productName	char(30)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

nchar (uzunluk): Yine bu veri türü de alfabetik karakterlerin tanımlanması için kullanılmaktadır. Bu veri türü bilgileri ANSI formatta değil de Unicode formatında saklamaktadır. ANSI formatı 256 farklı karakter barındırabilir fakat Unicode formatı 65.536 farklı karakteri barındırabilmektedir.


Örnek: ‘staff’ tablosunda, «staffName» alanı aşağıdaki şekilde tanımlanabilir.

	Column Name	Data Type	Allow Nulls
	staffID	int	<input type="checkbox"/>
	staffName	nchar(20)	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

varchar (uzunluk): Yine bu veri türü de alfabetik olan alanlar tanımlamak için kullanılmaktadır. Diğer iki veri türünden farklı olarak değişen tek şey uzunluk bilgisidir. Burada parametre olarak belirtilen uzunluk, aynı zamanda maksimum uzunluk olmaktadır.


Örnek: ‘staff’ tablosunda, «staffName» alanı aşağıdaki şekilde tanımlanabilir.

	Column Name	Data Type	Allow Nulls
	staffID	int	<input type="checkbox"/>
	staffName	varchar(20)	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

nvarchar (uzunluk): ‘varchar’ veri türü için geçerli olan her şey bu veri türü içinde aynen geçerlidir. Farklı olarak veriler ANSI formatta değil de Unicode formatta tutulmaktadır.

Örnek: ‘staff’ tablosunda, «staffName» alanı aşağıdaki şekilde tanımlanabilir.

	Column Name	Data Type	Allow Nulls
	staffID	int	<input type="checkbox"/>
	staffName	nvarchar(50)	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

text: Çok büyük açıklama verilerini saklamak için kullanılmaktadır. Örneğin bir tabloda Açıklama olarak tanımlanan bir nitelik alanına çok sayıda karakteri içeren bir bilgi girilmek istenildiğinde kullanılabilir. 2 GB ta kadar veri girilebilen bir alandır.

Örnek: ‘products’ tablosunda, «description» alanı aşağıdaki şekilde tanımlanabilir.

	Column Name	Data Type	Allow Nulls
?	productID	int	<input type="checkbox"/>
	productName	char(30)	<input checked="" type="checkbox"/>
	description	text	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

ntext: ‘text’ veri türü için bahsetmiş olduğumuz bütün özellikleri barındıran bir veri türüdür. Farklı olarak verileri ANSI formatta değil de Unicode formatta tutmaktadır.

Örnek: ‘products’ tablosunda, «description» alanı aşağıdaki şekilde tanımlanabilir.

	Column Name	Data Type	Allow Nulls
🔑	productID	int	<input type="checkbox"/>
	productName	char(30)	<input checked="" type="checkbox"/>
	description	ntext	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

image: Veritabanında resim tutmak için kullanılan bir veri türüdür.

Örnek: ‘staff’ tablosunda kişinin adı, soyadı bilgilerinin yanı sıra resim bilgisi de tutulmak isteniyorsa bu veri türü kullanılabilir.

	Column Name	Data Type	Allow Nulls
🔑	staffID	int	<input type="checkbox"/>
	staffName	nvarchar(50)	<input checked="" type="checkbox"/>
	staffSurname	nvarchar(50)	<input checked="" type="checkbox"/>
	picture	image	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

int (integer): Tam sayı verilerini saklamak için kullanılmaktadır. Bu alana pozitif ve negatif tamsayılar girilebilir. Bu alana girilebilen negatif ve pozitif tamsayılar

-2.147.483.648 ile 2.147.483.647 arasındaki değerler girilebilir.


Örnek: ‘staff’ tablosunda kişinin aldığı ücret bilgisi tutulmak isteniyorsa bu veri türü kullanılabilir.

	Column Name	Data Type	Allow Nulls
🔑	staffID	int	<input type="checkbox"/>
	staffName	nvarchar(50)	<input checked="" type="checkbox"/>
	staffSurname	nvarchar(50)	<input checked="" type="checkbox"/>
	picture	image	<input checked="" type="checkbox"/>
	salary	int	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

bigint (big integer): Çok büyük sayısal verileri saklamak için kullanılan bir veri türüdür. Bu alana -9.223.372.036.854.775.808 ile 9.223.372.036.854.775.807 arasındaki değerler girilebilir.

Örnek: staff tablosunda kişinin aldığı ücret bilgisi tutulmak isteniyorsa bu veri türü kullanılabilir.

	staffID	int	<input type="checkbox"/>
	staffName	nvarchar(50)	<input checked="" type="checkbox"/>
	staffSurname	nvarchar(50)	<input checked="" type="checkbox"/>
	picture	image	<input checked="" type="checkbox"/>
	salary	bigint	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

smallint (small integer): Küçük tamsayı değerlerini tanımlamak için kullanılan bir veri türüdür. Bu alana -32.768 ile 32.765 arasındaki değerler yazılabilir.


Örnek: ‘staff’ tablosunda kişinin aldığı ücret bilgisi tutulmak isteniyorsa bu veri türü kullanılabilir.

	Column Name	Data Type	Allow Nulls
🔑	staffID	int	<input type="checkbox"/>
	staffName	nvarchar(50)	<input checked="" type="checkbox"/>
	staffSurname	nvarchar(50)	<input checked="" type="checkbox"/>
	picture	image	<input checked="" type="checkbox"/>
	salary	smallint	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

tinyint: Bu alan 0 ile 255 arasındaki sayısal değerlerin eklenebilmesi için kullanılan bir veri türüdür.


Örnek: ‘products’ tablosunda ürünlerin adet bilgileri tutulmak isteniyorsa bu veri türü kullanılabilir.

	Column Name	Data Type	Allow Nulls
	productID	int	<input type="checkbox"/>
	productName	char(30)	<input checked="" type="checkbox"/>
	description	ntext	<input checked="" type="checkbox"/>
	productNumber	tinyint	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

decimal: Ondalıklı yani virgüllü sayıların saklanması için tanımlanan bir veri türüdür.


Örnek: ‘products’ tablosunda ürünlerin fiyat bilgileri tutulmak isteniyorsa bu veri türü kullanılabilir.

	Column Name	Data Type	Allow Nulls
	productID	int	<input type="checkbox"/>
	productName	char(30)	<input checked="" type="checkbox"/>
	description	ntext	<input checked="" type="checkbox"/>
	productNumber	tinyint	<input checked="" type="checkbox"/>
	price	decimal(18, 0)	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

float: Çok büyük kesirli sayıları saklamak için kullanılan bir veri türüdür.


Örnek: ‘staff’ tablosunda çalışanın aylık prim bilgileri tutulmak isteniyorsa bu veri türü kullanılabilir.

	Column Name	Data Type	Allow Nulls
	staffID	int	<input type="checkbox"/>
	staffName	nvarchar(50)	<input checked="" type="checkbox"/>
	staffSurname	nvarchar(50)	<input checked="" type="checkbox"/>
	picture	image	<input checked="" type="checkbox"/>
	salary	smallint	<input checked="" type="checkbox"/>
	bonus	float	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

real: ‘float’ a göre daha küçük ondalıklı sayıları saklamak için kullanılan bir veri türüdür.


Örnek: ‘staff’ tablosunda çalışanın aylık prim bilgileri tutulmak isteniyorsa bu veri türü kullanılabilir.

	Column Name	Data Type	Allow Nulls
	staffID	int	<input type="checkbox"/>
	staffName	nvarchar(50)	<input checked="" type="checkbox"/>
	staffSurname	nvarchar(50)	<input checked="" type="checkbox"/>
	picture	image	<input checked="" type="checkbox"/>
	salary	smallint	<input checked="" type="checkbox"/>
	bonus	real	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

money ve smallmoney: Parasal değerlerin saklanması için kullanılan veri türleridir.


Örnek: ‘staff’ tablosunda çalışanın aylık ücret bilgileri tutulmak isteniyorsa bu veri türü kullanılabilir.

	Column Name	Data Type	Allow Nulls
	staffID	int	<input type="checkbox"/>
	staffName	nvarchar(50)	<input checked="" type="checkbox"/>
	staffSurname	nvarchar(50)	<input checked="" type="checkbox"/>
	picture	image	<input checked="" type="checkbox"/>
	salary	money	<input checked="" type="checkbox"/>
	bonus	real	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

datetime ve smalldatetime: Tarih ve saat bilgilerinin saklanması için kullanılan veri türleridir.

Örnek: ‘staff’ tablosunda çalışanın doğum tarihi bilgisi tutulmak isteniyorsa bu veri türü kullanılabilir.

	Column Name	Data Type	Allow Nulls
	staffID	int	<input type="checkbox"/>
	staffName	nvarchar(50)	<input checked="" type="checkbox"/>
	staffSurname	nvarchar(50)	<input checked="" type="checkbox"/>
	picture	image	<input checked="" type="checkbox"/>
	salary	money	<input checked="" type="checkbox"/>
	bonus	real	<input checked="" type="checkbox"/>
	birthdate	datetime	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

binary: İkili (binary) verilerin saklanması için kullanılan bir veri türüdür.


binary(n) şeklinde, n değeri 1 ile 8000 arasında değer alır.

n bayt kadar yer kaplar.

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

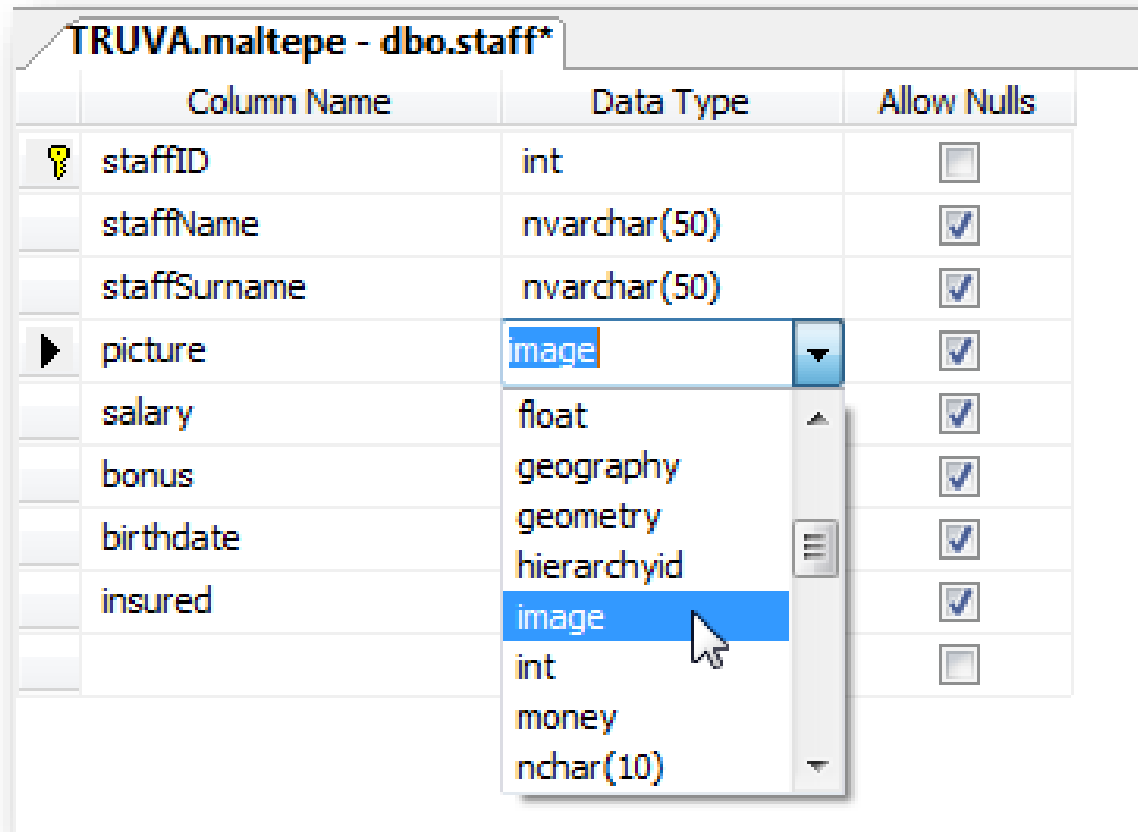
bit: 0 ve 1 gibi iki değerin yani iki seçenekli verilerin saklanması sağlayan veri türüdür.

Örnek: ‘staff’ tablosunda çalışanın sigortalı olup olmadığı bilgisi tutulmak isteniyorsa bu veri türü kullanılabilir.

	Column Name	Data Type	Allow Nulls
	staffID	int	<input type="checkbox"/>
	staffName	nvarchar(50)	<input checked="" type="checkbox"/>
	staffSurname	nvarchar(50)	<input checked="" type="checkbox"/>
	picture	image	<input checked="" type="checkbox"/>
	salary	money	<input checked="" type="checkbox"/>
	bonus	real	<input checked="" type="checkbox"/>
	birthdate	datetime	<input checked="" type="checkbox"/>
	insured	bit	<input checked="" type="checkbox"/>

6.3. TEMEL VERİ TÜRLERİ (DEVAM...)

MSSQL resim(image) veri türlerinin tanımlandığı ekran aşağıdaki gibidir.



The screenshot shows the 'dbo.staff' table structure in SQL Server Enterprise Manager. The table has the following columns and data types:

Column Name	Data Type	Allow Nulls
staffID	int	<input type="checkbox"/>
staffName	nvarchar(50)	<input checked="" type="checkbox"/>
staffSurname	nvarchar(50)	<input checked="" type="checkbox"/>
picture	image	<input checked="" type="checkbox"/>
salary	float	<input checked="" type="checkbox"/>
bonus	geography	<input checked="" type="checkbox"/>
birthdate	geometry	<input checked="" type="checkbox"/>
insured	hierarchyid	<input checked="" type="checkbox"/>
	image	<input checked="" type="checkbox"/>
	int	<input type="checkbox"/>
	money	
	nchar(10)	

6.4. SORGULAMA İŞLEMLERİ

SQL'de sorgulama işlemleri, SELECT deyimi yardımıyla yerine getirilir. SELECT deyimi temel olarak üç farklı işlemi yerine getirmek için kullanılır:

- **Seçme İşlemi (Select):** Bir tablodaki satırları seçme işlemidir. Bunun için çeşitli seçme kriterleri kullanılabilir.
- **Atma İşlemi (Projection):** Bir tablodaki belirli kolonların seçilmesi işlemidir.
- **Birleştirme İşlemi (Join):** Birden fazla tablonun belirlenen sütunları arasındaki ilişkilerin kullanılarak birleştirilmesi işlemidir.

6.4.1. SELECT DEYİMİNİN YAPISI

SELECT deyimi en basit biçimde şu şekilde ifade edilmektedir;

```
SELECT [DISTINCT] * / <sutun>  
FROM <Tablo_Adi/Tablolar>
```

Tanım içinde bazı SQL anahtar kelimelerine yer verilmektedir. Bu anahtar kelimeler, SQL 'in kendi özel kelimeleridir ve aynen bu şekilde ifade edilmelidir.

- **SELECT** → SQL'in sorgulama deyimidir.
- **FROM** → Hangi tablonun sorgulanacağını ifade eder.
- **DISTINCT** → Çift kayıtları önleyen anahtar kelimedir.

6.4.1. SELECT DEYİMİNİN YAPISI (DEVAM...)

SELECT ifadesinden sonra sorgu sonucunda ekranda görüntülenmesi istenen sütun isimlerinin arasına virgül konarak yan yana yazılır. Eğer tabloda bulunan bütün alanlar listelenmek istenirse * işareti kullanılır.

Örneğin;

- **SELECT * FROM students**

- Bu sorgu çalıştırıldığında, öğrenciler tablosundaki bütün alanları göstermektedir.

- **SELECT studentName, studentSurname FROM students**

- Bu sorgu çalıştırıldığında, öğrenciler tablosunda yer alan öğrencilerin sadece Adı ve Soyadı bilgilerini göstermektedir.

6.4.1. SELECT DEYİMİNİN YAPISI (DEVAM...)

FROM ifadesinden sonra listeleme işleminin hangi tablo üzerinde yapılacağı belirtilir. Birden fazla tablo kullanıldığında aralarına virgül işareti konur.

Örneğin;

- **SELECT * FROM students, departments**
 - Bu sorgu çalıştırıldığında, öğrenciler ve bölümler tablosundaki bütün alanları göstermektedir.

6.4.1. SELECT DEYİMİNİN YAPISI (DEVAM...)

SELECT * **FROM** **students, departments** sorgusunda her iki tablodan da belli alanlar görüntülenmek istenirse, **SELECT** ve **FROM** ifadeleri arasına tablo ismi belirtilerek yazılır.

Örneğin;

- **SELECT** **students.studentName,**
students.studentSurname,
departments.departmentName
FROM **students, departments**

6.4.2. WHERE DEYİMİ

WHERE deyimi, tablo içerisinde istenen bir şarta uygun olan kayıtların listelenmesi amacıyla kullanılmaktadır.

İstenen şartlar WHERE ifadesinden sonra yazılmaktadır.

WHERE koşul ifadesine bağlı olarak listelenmesi istenen alanlarda yazılabilir.

- Şart ifadesi içerisinde kullanılacak ifade **rakam ise tırnak işareti kullanılmaz.** Eğer **alfabetik bir ifade ise** aranacak ifade **tek tırnaklar arasına** yazılır.

6.4.2. WHERE DEYİMİ (DEVAM...)

Örnek:

```
SELECT * FROM Musteriler WHERE MusteriNo = 10
```

- Müşteriler tablosunda Müşteri numarası 10 olan kişinin tüm bilgilerini listeler.

```
SELECT Adi, Soyadi  
FROM Musteriler  
WHERE MusteriNo = 10
```

- Müşteriler tablosunda Müşteri numarası 10 olan kişinin sadece Adı ve Soyadı bilgilerini listelenir.

6.5. SQL OPERATÖRLERİ

Her programlama dilinde olduğu gibi **SQL** de de bazı **operatörler** bulunmaktadır. Bu operatörler, listeleme yapılırken istenilen koşulların yerine getirilmesine yardımcı olmaktadır. Bu operatörleri 4 ana başlık altında toplamak mümkündür.

- *Karşılaştırma operatörleri,*
- *Mantıksal operatörler,*
- *Aritmetiksel ifadeler,*
- *Kümeleme operatörleri.*

6.5.1. KARŞILAŞTIRMA OPERATÖRLERİ

Karşılaştırma operatörleri, verilen iki ifadenin birbirleri ile kıyaslanmasını sağlamaktadır. Bu ifadeleri şöyle sıralayabiliriz.

- $X > Y$, X ifadesi Y ifadesinden **büyüktür**.
- $X < Y$, X ifadesi Y ifadesinden **küçüktür**.
- $X = Y$, X ifadesi Y ifadesine **eşittir**.
- $X >= Y$, X ifadesi Y ifadesine **eşit ve büyüktür** olabilir.
- $X <= Y$, X ifadesi Y ifadesine **eşit ve küçüktür** olabilir.
- $X <> Y$, X ifadesi Y ifadesine **eşit değildir**.

6.5.1. KARŞILAŞTIRMA OPERATÖRLERİ (DEVAM...)

Karşılaştırma operatörlerine, «Siparişler» tablosunu kullanarak bir örnek verelim. Nakliye ücreti 20 TL den büyük olan siparişler hangileridir? Bu sorgunun SQL cümlesi aşağıdaki gibi olacaktır:

- **SELECT ***
FROM Siparisler
WHERE NakliyeUcreti > 20

6.5.2. MANTIKSAL 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)

Önemli Nokta!

- **SQL'de** bu operatörlerin hepsi **WHERE** deyiminden sonra kullanılmaktadır.

6.5.2. MANTIKSAL OPERATÖRLER (DEVAM...)

AND operatörüne örnek verecek olursak; siparişler tablosunda, nakliye ücreti 10 TL den küçük **ve** Ankara'ya gönderilen siparişler hangileridir?

SELECT *

FROM Siparisler

WHERE (NakliyeUcreti<10)AND (GonderilenIL='Ankara')

6.5.2. MANTIKSAL OPERATÖRLER (DEVAM...)

OR operatörüne örnek verecek olursak; siparişler tablosunda, nakliye ücreti 10 TL den küçük yada Ankara' ya gönderilen siparişler hangileridir?

SELECT *

FROM Siparisler

WHERE (NakliyeUcreti <10) OR (Gönderilenİl='Ankara')

6.5.2. MANTIKSAL OPERATÖRLER (DEVAM...)

NOT operatörüne örnek verecek olursak; siparişler tablosunda, nakliye ücreti 10 TL den küçük olmayan siparişler hangileridir?

SELECT *

FROM Siparisler

WHERE NOT (NakliyeUcreti <10)

6.5.3. ARİTMETİKSEL İFADELER

Aritmetiksel ifadeler, tablo içerisindeki herhangi bir sütun üzerinde işlem yapmayı sağlamaktadır.

Bu operatörler;

- Toplama (+),
- Çıkarma (-),
- Çarpma (*),
- Bölme (/).

işlemlerden oluşmaktadır.

6.5.3. ARİTMETİKSEL İFADELER (DEVAM...)

Aritmetiksel işlevler yapılırken **işlem sırası önemlidir.**

İşlem sırası şu şekildedir:

■ * / + -

Örneğin bir hücre ile işlem yapılırken şu şekilde kullanılmaktadır:

■ NakliyeUcreti * 12

6.5.3. ARİTMETİKSEL İFADELER (DEVAM...)

Örneğin, “Siparisler” tablosunda nakliye ücretlerinin sabit olduğunu düşünelim. Bu satırların her biri için yıllık nakliye ücreti ne olur?

- **SELECT NakliyeUcreti * 12**
FROM Siparisler

6.5.3. ARİTMETİKSEL İFADELER (DEVAM...)

Aritmetiksel ifadeler kullanılırken parantezlerin kullanımı son derece önemlidir. Parantezlerin, işlem öncelikleri göz önüne alınarak doğru yerde kullanılması gerekmektedir. Aksi durumda yanlış işlem sonuçları ortaya çıkabilmektedir.

Örneğin; ücretin 10 TL olduğunu varsayarsak;

- $12 * (\text{ucret} + 500) = 12 * 510 = 6120$
- $12 * \text{ucret} + 500 = 120 + 500 = 620$

Burada işlem önceliği çarpmanın olduğundan sonuç farklı olacaktır.

6.5.4. KÜMELEME OPERATÖRLERİ

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

- **BETWEEN ... AND ...**
- **IN**
- **LIKE**

Bu komutlar yine WHERE komutundan sonra kullanılmaktadır.

6.5.4. KÜMELEME OPERATÖRLERİ (DEVAM...)

BETWEEN komutu belli bir aralıktaki kayıtların sorgulanmasında kullanılmaktadır. Bu komutta alt ve üst sınır tanımlamaları yapılmaktadır. Genel kullanımı şöyledir.

BETWEEN *Alt_Sınır* **AND** *Ust_Sınır*

Örnek: Nakliye ücreti 10 ve 20 TL arasındaki siparişlerin detayları nelerdir?

■ **SELECT ***

FROM Siparisler

WHERE NakliyeUcreti **BETWEEN 10 AND 20**

6.5.4. KÜMELEME OPERATÖRLERİ (DEVAM...)

IN komutu, kayıtların tablodaki değerlerle karşılaştırılması söz konusu olduğunda kullanılmaktadır.

Örneğin; Ankara, İzmir ve İstanbul'a gönderilen siparişler hangileridir? sorgusunu yazalım.

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


6.5.4. KÜMELEME OPERATÖRLERİ (DEVAM...)

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

Örneğin, siparişlerin gönderildiği kişilerden adları **N ile başlayan** kişiler kimlerdir? sorgusunu yapalım.

■ **SELECT** GonderilenAd

FROM Siparisler

WHERE GonderilenAd **LIKE** 'N%'

6.5.4. KÜMELEME OPERATÖRLERİ (DEVAM...)

Örneğin, siparişlerin gönderildiği kişilerden adlarının **içinde N harfi bulunan** kişiler kimlerdir? sorgusunu yapalım.

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

6.5.4. KÜMELEME OPERATÖRLERİ (DEVAM...)

LIKE deyimi ile birlikte kullanılan karakterlerden bir diğeri `_` işaretidir. Bu işaret bir harf veya rakamın yerini tutar.

Örneğin, siparişlerin gönderildiği kişilerden adlarının **içinde sondan ikinci harfi N olan** kişiler kimlerdir? sorgusunu yapalım.

■ **SELECT** GonderilenAd

FROM Siparisler

WHERE GonderilenAd **LIKE** '%N_'

6.6. ORDER BY İFADESİ

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.

6.6. ORDER BY İFADESİ (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.

6.6. ORDER BY İFADESİ (DEVAM...)

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.

6.6. ORDER BY İFADESİ (DEVAM...)

Örnek: Bahçe İşleri veritabanında siparişler tablosunda, nakliye ücreti 10 TL ile 25 TL arasında olan siparişlerin siparişno, nakliyecisi ve gönderilen kişi bilgilerini büyükten küçüğe doğru sıralayan SQL sorgusu şöyle olacaktır.

- **SELECT** SiparisNo, Nakliyecisi, GonderilenAd, NakliyeUcreti
FROM Siparisler
WHERE NakliyeUcreti **BETWEEN** 10 **AND** 25
ORDER BY NakliyeUcreti **DESC**

6.6. ORDER BY İFADESİ (DEVAM...)

Sorgu çalıştırıldıktan sonra yandaki ekran karşınıza çıkacaktır.

Siparişler Sorgu: Seçme Sorgusu

SiparişNo	Nakliyeci	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	Başı Temur	19,95 YTL
11095	Yollar Kargo	Feride Karaoğlu	18,95 YTL
11119	Gök Nakliye	Erkan Kocabıyık	18,50 YTL
11098		Musa Löklüoğlu	18,50 YTL
11079	Yollar Kargo	Tayfun Süsler	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	Barkin 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

6.6. ORDER BY İFADESİ (DEVAM...)

Şimdi aynı sorguyu **küçükten büyüğe** doğru yapalım.

- **SELECT** SiparisNo, Nakliyecisi, GonderilenAd, NakliyeUcreti
FROM Siparisler
WHERE NakliyeUcreti **BETWEEN 10 AND 25**
ORDER BY NakliyeUcreti

veya

- **SELECT** SiparisNo, Nakliyecisi, GonderilenAd, NakliyeUcreti
FROM Siparisler
WHERE NakliyeUcreti **BETWEEN 10 AND 25**
ORDER BY NakliyeUcreti **ASC**

6.6. ORDER BY İFADESİ (DEVAM...)

ORDER BY içerisinde birden fazla ifade ile sıralama yaptırmak mümkündür.

- Bahçe İşleri veritabanında siparişler tablosunda, nakliye ücreti 10 TL ile 25 TL arasında olan siparişlerin siparişno, nakliyecisi, gönderilen kişi ve sipariş tarihi bilgilerini göstererek, önce nakliye ücretini büyükten küçüğe sonra da sipariş tarihini küçükten büyüğe doğru sıralayan SQL sorgusu şöyle olacaktır.

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

6.6. ORDER BY İFADESİ (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

TEŞEKKÜRLER...

SORULARINIZ ?
