



Veri Tabanı Yönetim Sistemleri

Bölüm - 4



İçerik

- SQL'e Giriş.
- SQL Yapısal Sorgulama Dili.
- Temel SQL Komutları:
 - Sorgulama İşlemleri
 - SELECT deyiminin temel yapısı
 - Seçme İşlemi
 - Atma İşlemi
 - Aritmetik İfadelerin Kullanımı
 - Birleştirme İşlemi (CONCAT)
 - Takma İsim (AS)
 - Aynı Değere Sahip Satırlar (DISTINCT)



SQL

Structured Query Language - Yapısal Sorgulama Dili

Tarihi Gelişim:

- Veri tabanı kavramının olgunlaşması ile bir veri sorgulama diline ihtiyaç duyulmaya başlanmıştır.
- İlk başta matematiksel bir sözdizimine sahip olan SQUARE adlı bir dil geliştirilmiştir.
- Kolay kullanım hedeflenerek İngilizceye benzer bir sözdizimi kullanılması planlanmıştır.
- SEQUEL – Structured English Query Language olarak adlandırılan bu yeni dil SQUARE'in yerini almıştır.
- Günümüzde SEQUEL'in kısaltması olarak SQL kullanılmaya başlanmıştır.



SQL – Yapısal Sorgulama Dili

- SQL bir sorgulama dilidir.
- SQL'in kendine özgü deyimleri ve kuralları vardır.
- Hemen hemen tüm ilişkisel veri tabanı yönetim sistemlerinin ortak dili olarak kabul edilmiştir.
- Ortak bir dil olarak standart bir yapıya sahip olsa da veritabanına göre değişebilen bir takım farklı ifade biçimleri vardır.
- SQL veriler üzerinde iki ana kontrol alanı sağlamaktadır:
 - Veri Tanımlama (Create, Alter, Drop, ...)
 - Veri İşleme (Select, Update, Insert, Delete, ...)



SQL: Sorgulama İşlemleri

- 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 sütunların seçilmesi işlemidir.
 - **Birleştirme İşlemi (Join):** Bir tablonun içinde yer alan sütunların (ya da birden fazla tablonun belirlenen sütunlarının ilişkiler yardımıyla) birleştirilmesi.



Sorgulama İşlemleri: Select Deyimi

SELECT deyimi en basit biçimde şu şekilde ifade edilmektedir;

```
SELECT [DISTINCT] { * | sütun, .... }  
FROM tablo;
```

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.



Sorgulama İşlemleri: Select Deyimi

```
SELECT [DISTINCT] { * | sütun, .... }  
FROM tablo;
```

Yukarıdaki ifadenin her bir işaretin veya farklı gösterimin hangi anlama geldiğini aşağıdaki tablodan görebilirsiniz.

[]	Kullanılması zorunlu olmayan SQL sözcükleri, bu işaretler arasında tanımlanır.
<u>tablo</u>	Altı çizili ve italik olan bu ifadeler, kullanıcı tarafından verilen isimleri ifade etmektedir. Bunlar SQL sözcüğü değildir.
{...,... }	Bu biçimde gösterilen ifadeler, birden fazla seçeneğin varlığını ve bu seçeneklerden birinin mutlaka seçilmesi gerektiğini ifade eder. Seçenekler birbirlerinden , işareti ile ayrılmaktadır.
*	Tek bir sütunu değil, tüm sütunları ifade eder.



Sorgulama İşlemleri: Select Deyimi

SELECT yazım kuralları:

- 1.Kural: SQL deyimleri büyük – küçük harf ayırımından etkilenmez. SQL deyiminin bir kısmı büyük; diğer bir kısmı küçük harflerle ifade edilebilir.
- 2.Kural: Bir SQL deyimi bir veya daha fazla satırda yer alabilir.
- 3.Kural: Deyimin bittiği yeri belirlemek için en sonuna (;) işareti kaydedilir.

SQL deyimleri düzenlenirken bu kuralları ve önerileri göz önünde tutmak gerekmektedir. Aksi takdirde sorgu hata verebilir.



Select Deyimi: Seçme İşlemi

Bir tablonun tüm satırlarını ve sütunlarını seçmek

Örnek Öğrenci Tablosu (ogrenci)				
sirano	ogrencino	ad	soyad	eposta
1	345627	Ahmet	Dervek	ahmet@dervek.com
2	567483	Deniz	Kulcu	deniz@kulcu.net
3	785942	Ebru	Gündüz	ebru@gunduz.org

SELECT * FROM ogrenci;

SELECT *
FROM ogrenci;



sirano	ogrencino	ad	soyad	eposta
1	345627	Ahmet	Dervek	ahmet@dervek.com
2	567483	Deniz	Kulcu	deniz@kulcu.net
3	785942	Ebru	Gündüz	ebru@gunduz.org



Select Deyimi: Atma İşlemi

Bir tablonun belirli sütunlarını seçmek

Örnek Öğrenci Tablosu (ogrenci)				
sirano	ogrencino	ad	soyad	eposta
1	345627	Ahmet	Dervek	ahmet@dervek.com
2	567483	Deniz	Kulcu	deniz@kulcu.net
3	785942	Ebru	Gündüz	ebru@gunduz.org

SELECT ad, soyad, eposta
FROM ogrenci;



ad	soyad	eposta
Ahmet	Dervek	ahmet@dervek.com
Deniz	Kulcu	deniz@kulcu.net
Ebru	Gündüz	ebru@gunduz.org



Select Deyimi: Aritmetik İfadeler

Aritmetik ifadelerin kullanımı

Örnek Öğrenci Tablosu (ogrenci)				
sirano	ogrencino	ad	soyad	vize
1	345627	Ahmet	Dervek	15
2	567483	Deniz	Kulcu	12
3	785942	Ebru	Gündüz	6

SELECT ad, soyad, vize * 5
FROM ogrenci;



ad	soyad	vize * 5
Ahmet	Dervek	75
Deniz	Kulcu	60
Ebru	Gündüz	30



Select Deyimi: Aritmetik İfadeler

Aritmetik İfadelerde işlem sırası: * / + -

Örnek Öğrenci Tablosu (ogrenci)				
sirano	ogrencino	ad	soyad	vize
1	345627	Ahmet	Dervek	15
2	567483	Deniz	Kulcu	12
3	785942	Ebru	Gündüz	6

SELECT ad, soyad, vize * (5 + 10)
FROM ogrenci;



ad	soyad	vize * 5 + 10
Ahmet	Dervek	85
Deniz	Kulcu	70
Ebru	Gündüz	40



Select Deyimi: Aritmetik İfadeler

Aritmetik İfadelerde Parantez Kullanımı

Parantezler, aritmetik ifadeler içinde diğer tüm işlemlere göre bir önceliğe sahiptir. Önce parantez içindeki işlemler gerçekleştirilir.

Örnek Öğrenci Tablosu (ogrenci)		
ad	soyad	aylikyardim
Ahmet	Dervek	500
Deniz	Kulcu	600
Ebru	Gündüz	700

```
SELECT ad, soyad, 12 * yardım + 100  
FROM ogrenci;
```

ad	soyad	aylikyardim
Ahmet	Dervek	6100
Deniz	Kulcu	7300
Ebru	Gündüz	8500

Örnek Öğrenci Tablosu (ogrenci)		
ad	soyad	aylikyardim
Ahmet	Dervek	500
Deniz	Kulcu	600
Ebru	Gündüz	700

```
SELECT ad, soyad, 12 * (yardim + 100)  
FROM ogrenci;
```

ad	soyad	aylikyardim
Ahmet	Dervek	7200
Deniz	Kulcu	8400
Ebru	Gündüz	9600



Select Deyimi: Birleştirme İşlemi

Bir tablonun belirli sütunları birleştirmek

Örnek Öğrenci Tablosu (ogrenci)				
sirano	ogrencino	ad	soyad	eposta
1	345627	Ahmet	Dervek	ahmet@dervek.com
2	567483	Deniz	Kulcu	deniz@kulcu.net
3	785942	Ebru	Gündüz	ebru@gunduz.org

SELECT CONCAT(ad, soyad) AS adisoyadi
FROM ogrenci;



adisoyadi
AhmetDervek
DenizKulcu
EbruGündüz

SELECT CONCAT(ad, ' ', soyad) AS adisoyadi
FROM ogrenci;



adisoyadi
Ahmet Dervek
Deniz Kulcu
Ebru Gündüz



Select Deyimi: DISTINCT Özelliği

Bir SELECT deyimi çalıştırıldığında, aynı değerlere sahip satırların görüntülenmesi engellenemez.

Tekrar eden bu değerlerden sadece bir tanesinin görüntülenmesi hedefleniyorsa SELECT deyimi DISTINCT ifadesi ile birlikte kullanılır

Öğrenci Tablosu (ogrenci)		
ad	soyad	semt
Oktay	Bağcı	Cebeci
Hakan	Ars	Etlik
İbrahim	Bağcı	Etlik
Halil	Bağcı	Etlik
Savaş	Özbek	Balgat
Tolga	Ergin	Balgat
Erkin	Şahin	Cebeci
Hakan	Kuşçu	Emek

SELECT semt
FROM ogrenci;



semt
Cebeci
Etlik
Etlik
Etlik
Balgat
Balgat
Cebeci
Emek

SELECT **DISTINCT** semt
FROM ogrenci;



semt
Cebeci
Etlik
Balgat
Emek

