

SQL DİLİ

VERİ SORGU DİLİ (DQL)

SQL KATEGORİLERİ

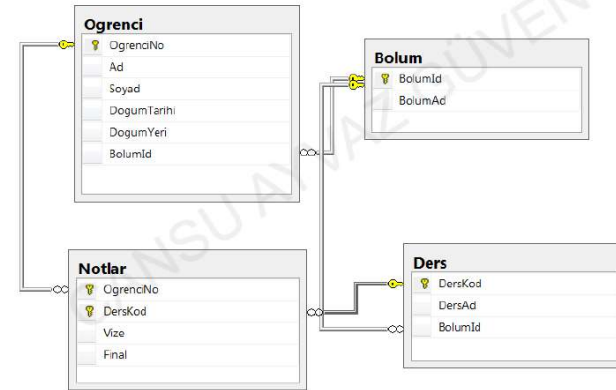
İŞLEV	AÇIKLAMA
DDL (Data Definition Language – Veri Tanımlama Dili)	Veritabanı şemasını tanımlamak ve veritabanındaki veritabanı nesnelerinin yapısını oluşturmak ve değiştirmek için kullanılır.
DQL (Data Query Language - Veri Sorgu Dili)	Veritabanındaki verileri sorgulamak ve sonuçları listelemek için kullanılır.
DML (Data Manipulation Language - Veri İşleme Dili)	Veritabanında bulunan verilerin değiştirilmesi (güncellenmesi, eklenmesi, silinmesi) için kullanılır.
DCL (Data Control Language – Veri Kontrol Dili)	Grup ve kullanıcılara veritabanı sisteminde verilen izinleri ve konan yasakları düzenlemek amacıyla kullanılır.
TCL (Transaction Control Language – İşlem Kontrol Dili)	Veritabanındaki işlemlerle ilgili bütünlük vb. için kullanılır.

Kaynak: “İlişkisel Veritabanı Tasarlama ve SQL ile Geliştirme”, Dr. Selçuk Kıran, Nobel Akademik Yayıncılık

VERİ SORGU DİLİ (DQL)

Select

Örnek Veritabanı



Kaynak: <http://www.cansuayvazguven.com>

SELECT

- SQL' de sorgulama işlemleri, **SELECT** deyimi kullanılarak yerine getirilir.

*Select *, SÜTUN,.. From Tablo_adı*

- Tablonun tüm sütunlarını seçmek için ***** kullanılır.
- Belirli sütunların seçilmesi için sütunların isimleri belirtilmelidir.

Kaynak: <http://www.cansuayvazguven.com>

SELECT

Örnek : *Ogrenci* tablosundaki tüm sütunları listelemek için,

```
SELECT * FROM Ogrenci
```

Örnek : *Ogrenci* tablosundaki *OgrenciNo*, *Ad*, *Soyad* sütunlarını listelemek için,

```
SELECT OgrenciNo, Ad, Soyad  
FROM Ogrenci;
```

Kaynak: <http://www.cansuayvazguven.com>

DISTINCT (Tekrarlı Satırların Engellenmesi)

Aynı değerlere sahip satırlar tekrarlı olarak görüntülenir. Yani kayıt sayısı kadar satır görüntülenir. Tekrarlı satırların yalnızca bir tanesi listelenmek isteniyorsa **SELECT** deyimi **DISTINCT** anahtar kelimesi ile kullanılır.

Kaynak: <http://www.cansuayvazguven.com>

Örnekler

- Aynı *BolumId*'lerinden sadece birer tanesi listelenmek istenirse,

```
SELECT DISTINCT BolumId  
FROM Ogrenci;
```

- Aynı ders kodlarından sadece birer tanesi listelenmek istenirse,

```
SELECT DISTINCT DersKod  
FROM Notlar;
```

Kaynak: <http://www.cansuayvazguven.com>

Bilgilerin Koşula Bağlı Olarak Seçilmesi

Belirli bir koşulu sağlayan kayıtların süzülmesi için WHERE Şart ifadesi kullanılır.

```
Select [DISTINCT] {*, SÜTUN,...}  
FROM TABLO  
WHERE (ŞART);
```

Kaynak: <http://www.cansuayvazguven.com>

Bilgilerin Koşula Bağlı Olarak Seçilmesi

Örnek: Ogrenci tablosundan doğum yeri İstanbul olan öğrencilerin OgrenciNo, Ad, Soyad bilgilerini listeleyen SQL ifadesi,

```
SELECT OgrenciNo, Ad, Soyad  
FROM Ogrenci  
WHERE DogumYeri = 'İstanbul';
```

Kaynak: <http://www.cansuayvazguven.com>

Bilgilerin Koşula Bağlı Olarak Seçilmesi

Karşılaştırma Operatörleri

= eşit
> büyük
< küçük
=> büyük eşit
<= küçük eşit
<> eşit değil

Örnekler

```
SELECT * FROM Notlar  
WHERE Final >=25
```

```
SELECT Adı, Soyad  
FROM Ogrenci  
WHERE BolumID <> 5
```

Kaynak: <http://www.cansuayvazguven.com>

Bilgilerin Koşula Bağlı Olarak Seçilmesi

Mantıksal Operatörler

AND (ve) : Her iki şartın değeri TRUE (doğru) ise, sonuç olarak TRUE (doğru) döndürür.

OR (veya) : Şartlardan birinin değeri TRUE (doğru) ise, sonuç olarak TRUE (doğru) döndürür.

NOT (değil) : Bu operatörü izleyen şart FALSE (yanlış) ise, sonuç olarak TRUE (doğru) döndürür.

Kaynak: <http://www.cansuayvazguven.com>

Örnekler:

- Bolum numarası 3'ten büyük **ve** doğum yeri İstanbul olan öğrenciyi listeleyen SQL ifadesi:

```
SELECT OgrenciNo, Ad, Soyad  
FROM Ogrenci  
WHERE BolumId > 3 AND DogumYeri = 'İstanbul';
```

- Vize notu 50'den büyük **ya da** vize notu 70'ten küçük olan öğrencileri listeleyen SQL ifadesi:

```
SELECT OgrenciNo, Vize, Final  
FROM Not  
WHERE Vize > 50 OR Vize < 70;
```

Kaynak: <http://www.cansuayvazguven.com>

Örnekler:

- Doğum yeri İstanbul, İzmir ya da Yalova olmayan öğrencileri listeleyen SQL ifadesi:

```
SELECT OgrenciNo, Ad, Soyad  
FROM Ogrenci  
WHERE DogumYeri NOT IN ('İstanbul', 'İzmir', 'Yalova')
```

- Doğum yeri İstanbul, İzmir ya da Yalova olan öğrencileri listeleyen SQL ifadesi:

```
SELECT OgrenciNo, Ad, Soyad  
FROM Ogrenci  
WHERE DogumYeri IN ('İstanbul', 'İzmir', 'Yalova')
```

Kaynak: <http://www.cansuayvazguven.com>

TOP (Limitli Seçim)

TOP fonksiyonu ile herhangi bir sorgu sonucu dönecek verilerin istenildiği kadarını ya da istenilen şekilde limitlenerek gösterilmesi sağlanabilir.

```
SELECT TOP (kayit_adi) alan_adi1, alan_adi2, ...  
FROM tablo_adi
```

Kaynak: <http://www.cansuayvazguven.com>

Örnekler

- Öğrenci tablosundaki ilk 10 kaydı listeleyen sql ifadesi,

```
Select top(10) Ad, Soyad From Ogrenci
```

- Bolumler tablosunda ilk kaydı listeleyen sql ifadesi,

```
Select top(1) * From Bolum
```

Kaynak: <http://www.cansuayvazguven.com>

ORDER BY (Sıralama)

- Sıralama işlemi için **SELECT** ifadesinde **ORDER BY** kullanılır.
- Bu ifadede **ASC** kelimesi kullanılırsa sıralamanın küçükten büyüğe (A dan Z ye),
- **DESC** kullanılırsa büyükten küçüğe doğru (Z den A ya) olacağı belirtilmiş olur.
- Her ikisi de kullanılmazsa sıralama küçükten büyüğe doğru yapılır.

Kaynak: <http://www.cansuayvazguven.com>

Kullanım Şekli

- Artan Sıralama;

```
SELECT * FROM tablo_adi ORDER BY referans_sütun ASC
```

- Azalan Sıralama;

```
SELECT * FROM tablo_adi ORDER BY referans_sütun DESC
```

Kaynak: <http://www.cansuayvazguven.com>

Örnekler

- Öğrenci adlarını alfabetik sırada listeleyen SQL ifadesi;

```
SELECT * FROM Ogrenci ORDER BY Ad ASC
```

- Öğrenci numaralarını büyükten küçüğe sıralayan SQL ifadesi;

```
SELECT * FROM Ogrenci ORDER BY OgrenciNo DESC
```

Kaynak: <http://www.cansuayvazguven.com>

NEWID() (Rastgele Kayıt Seçimi)

- Bir tablodan rastgele kayıt seçmek için **NEWID()** sistem fonksiyonu kullanılır.

- Kullanım Şekli

```
SELECT * FROM tablo_adi ORDER BY NEWID()
```

Kaynak: <http://www.cansuayvazguven.com>

Örnekler

- Öğrenci tablosundaki kayıtları rastgele sıralamak için,

`Select * From Ogrenci Order By NEWID()`

- Öğrenci tablosundan rastgele 5 kayıt için,

`Select Top 5 * From Ogrenci Order By NEWID()`

Kaynak: <http://www.cansuayvazguven.com>

AS (Sütunlar İçin Takma İsim Kullanılması)

- Başlıkların alan adları dışında bir isimle görüntülenmesi için **AS** anahtar kelimesi kullanılır.
- Eğer takma isim birden fazla kelimeden oluşuyorsa çift tırnak (" ") işaretleri içerisinde yer almalıdır.

Kaynak: <http://www.cansuayvazguven.com>

Örnekler

- `SELECT` Ad `AS` İSİM `FROM` Ogrenci;
- `SELECT` Ad `AS` "Öğrenci Adı" `FROM` Ogrenci;
- `SELECT` Ad `AS` "Öğrenci Adı", Soyad `AS` "Öğrenci Soyadı" `FROM` Ogrenci;

Kaynak: <http://www.cansuayvazguven.com>

Sütunlar Üzerinde Matematiksel İşlemler

- Aritmetik ifadeler tablodan çekilen veriler üzerinde matematiksel işlemler yapmak için kullanılır.
- Aritmetik işlem yapılacak sütunun türü sayısal tipte olmalıdır.

Kaynak: <http://www.cansuayvazguven.com>

Örnek

- Notlar tablosunda vize notunun %40'ı, final notunun %60'ı alınarak başarı puanlarının hesaplanmasını sağlayan SQL ifadesi;

```
Select OgrenciNo, Vize, Final,  
((Vize*40/100)+(Final*60/100)) AS Ortalama  
From Notlar
```

Kaynak: <http://www.cansuayvazguven.com>

Sütunların Birleştirilmesi

- Öğrenci tablosunda ad, soyad sütunlarını birleştiren SQL ifadesi;

```
Select OgrenciNo, Ad + ' ' + Soyad AS "Ad Soyad"  
From Ogrenci
```

Kaynak: <http://www.cansuayvazguven.com>

(Between ... And...) Operatörü

İki değer arasında kalanları seçme işleminde kullanılan ifadedir. **Between** ifadesi başlangıç ve bitiş değerleri dahil olmak üzere sorgulama yapar.

```
SELECT * FROM tablo_adı  
WHERE referans_sütun BETWEEN ilk_değer AND  
son_değer
```

Kaynak: <http://www.cansuayvazguven.com>

Örnek

Notlar tablosunda vize notu 40 la 55 arasında olan vize notlarını listeleyen SQL ifadesi;

```
SELECT * FROM Notlar  
WHERE Vize BETWEEN 40 AND 55
```

Kaynak: <http://www.cansuayvazguven.com>

Like Operatörü

Karakter grubu ile karşılaştırma yapmak için kullanılır.

- **%** sıfır veya daha fazla herhangi bir karakteri ifade eder.
- **_** (alt çizgi) herhangi bir tek karakteri ifade eder.
- **[HARF]** Herhangi bir karakter yerine gelebilecek karakter kümesini ifade eder.
- **[^HARF]** Herhangi bir karakter yerine gelemeyecek karakter kümesini ifade eder.
- **[A-Z]** Herhangi bir karakter yerine A ile Z arasında karakter gelebileceğini ifade eder.

Kaynak: <http://www.cansuayvazguven.com>

Örnekler

- Adı A ile başlayanlar
`SELECT * FROM Ogrenci WHERE Ad Like 'a%'`
- Adı A ile bitenler
`SELECT * FROM Ogrenci WHERE Ad Like '%a'`
- Adında A harfi olanlar
`SELECT * FROM Ogrenci WHERE Ad Like '%a%'`
- Adının ikinci harfi a olanlar
`SELECT * FROM Ogrenci WHERE Ad Like '_a%'`

Kaynak: <http://www.cansuayvazguven.com>

Örnekler

- Adının ikinci harfi e son harfi a olanlar
`SELECT * FROM Ogrenci WHERE Ad Like '_e%a'`
- Adının ikinci harfi e, son harfi r olmayanlar
`SELECT * FROM Ogrenci WHERE Ad Like '_e%[^r]'`
- Adının ilk harfleri N ve R olanlar
`SELECT * FROM Ogrenci WHERE Ad Like '[NR]%'`
- Adı D ve S arasında olanlar
`SELECT * FROM Ogrenci WHERE Ad Like '[D-S]%'`

Kaynak: <http://www.cansuayvazguven.com>

NULL Değer Sorgulama

- NULL değer var olmayan bir değer olarak kabul edilir. Eğer bir sütün NULL değerler içeriyorsa SELECT deyimi çalıştırıldığında bu değerler sıfır olarak değil, boş olarak görüntülenir.
- Sorgulamada değerın NULL olup olmadığını bulmak için **IS NULL** ifadesi kullanılır.
- NULL değer içermeyen kayıtları bulmak için **IS NOT NULL** ifadesi kullanılır.

Kaynak: <http://www.cansuayvazguven.com>

Örnek

- Notlar tablosunda vize sınavına girmeyen öğrencilerin öğrenci numarası ve ders kodu bilgilerini listeleyen SQL ifadesi

```
SELECT OgrenciNo, DersKod FROM Notlar  
WHERE Vize IS NULL
```

- Notlar tablosunda NULL olmayan final notlarını listeleyen SQL ifadesi

```
SELECT Final FROM Notlar  
WHERE Final IS NOT NULL
```

Kaynak: <http://www.cansuayvazguven.com>

JOIN Komutu

İki veya daha fazla tabloyu birleştirmek için kullanılır.

Örnek:

```
SELECT Musteriler.MusteriAdi, Siparisler.SiparisNo FROM  
Musteriler INNER JOIN Siparisler ON Musteriler.MusteriID =  
Siparisler.MusteriID
```

Bu sorgu, "Musteriler" ve "Siparisler" tablolarını MusteriID sütunu üzerinden birleştirerek, her müşteri adı ile ilgili sipariş numaralarını gösterecektir.