



Koray Toksöz

ktoksoz@aydinorme.com.tr



Genel Kavramlar

- **Veri Nedir?**
 - Bilgisayar ortamında işlenebilecek durumda bulunan kayıtlar
- **Veri Tabanı Nedir?**
 - Birbiri ile ilişkili veriler topluluğu ve veriler arası ilişkiler
- **Veritabanı Yönetim Sistemi (DBMS) Nedir?**
 - Veritabanını yaratmak ve üzerinde çeşitli işlemler gerçekleştirme imkanı sağlayan yazılım



Genel Kavramlar

- İlişkisel Model Nedir?
 - Bilgi Kuralı: bütün veriler tablolarda tutulmalı
 - Erişim Garantisi: Tablolardaki her kayıda bir şekilde ulaşılabilmesi
 - Sistematiik boş (null) değeri desteęi: bütün değeriilerden farklı bir boş değeri işlenebilmeli
 - Aktif ilişkisel katalog: Veritabanı tanımı veritabanı dili kullanılarak sorgulanabilmeli.



Genel Kavramlar

- İlişkisel Model

- Ayrıntılı veri dili: En az bir dil tanımlı olmalı
- View güncelleme kuralı: Güncellenebilen tüm viewler sistem tarafından güncellenebilmeli
- Küme düzeyi kayıt girme, silme ve güncelleme: Sadece seçmek yetmez
- Fiziksel Veri Bağımsızlığı: Uygulama programları, fiziksel erişim değişikliklerinden etkilenmemeli.



Genel Kavramlar

- İlişkisel Model

- Mantıksal veri bağımsızlığı: Tablo yapılarındaki değişikliklerden uygulama programları etkilenmemeli.
- Bütünlük Bağımsızlığı: Bütünlük kuralları tanımlanabilmeli ve atlatılamamalı
- Dağıtım Bağımsızlığı: Uygulamalar verinin dağılımından ekilenmemeli
- “Yıkılmama”: Bütünlük kuralları atlatılamamalı.



Neden Veritabanı?

- Klasik Dosya Sisteminin Sakıncaları:
 - Veri tekrarları
 - Çoklu güncelleme
 - Bellek alanı israfı
 - Erişim dili

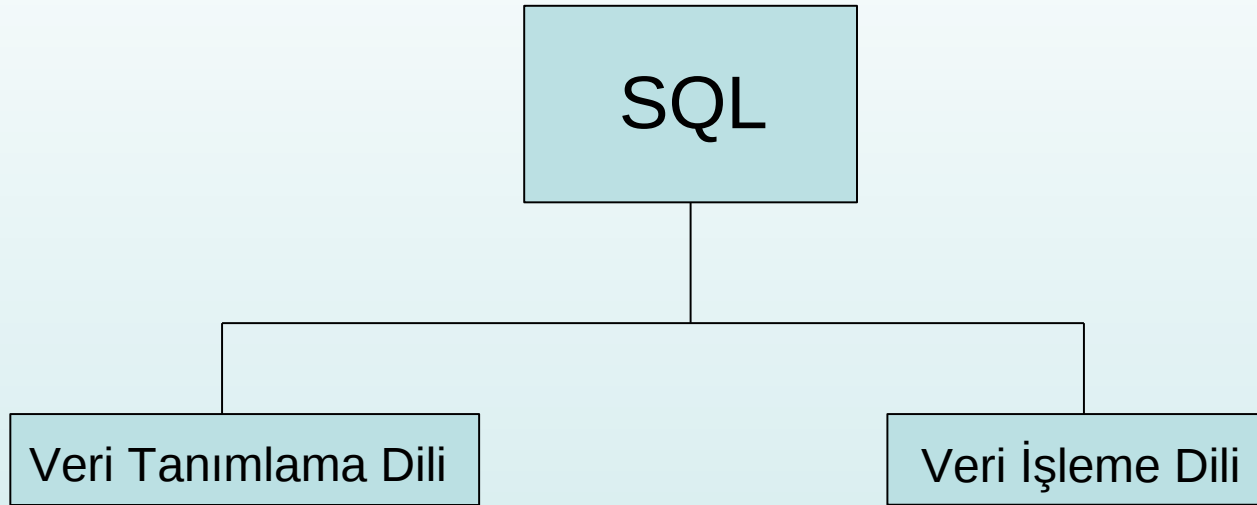


SQL

- Structured Query Language
 - Veritabanı yapısını değiştirme
 - Güvenlik tanımları yapma
 - Veritabanını bilgi için sorgulama
 - Veritabanı içeriğini güncelleme



SQL





Veritabanı Yaratılması

`CREATE DATABASE <isim>`

Tabloları içinde barındıran veritabanını isim adını kullanarak oluşturur.



Tablo Yaratılması

```
CREATE TABLE <tablo> (<kolon tanımı>,  
[kolon tanımı | anahtar tanımları])
```

Örnek:

```
Create table 'bolumler'  
(bolumID int not null,  
bolumADI char(10))
```



Tabloya Veri Girilmesi

INSERT INTO <tablo> VALUES (<değerler>)

<değerler> ile belirtilen verileri <tablo> tablosuna ekler.

Örnek:

Insert into personel values (1,'Koray','LKD')



Sorgulama

SELECT <kolonlar> form <tablo>

Select komutunun en basit hali. <kolonlar> sonuç olarak döner. Sorgu <tablo> tablosundan yapılır.

Örnek:

```
Select * from personel;
```



Tekrarsız Sorgulama

`SELECT DISTINCT <kolonlar> from <tablo>`

Birbirinin aynı satırlar listeleme esnasında bir kez yazılır.

Örnek:

`Select distinct from dersler;`



Sıralamalı Sorgulama

```
SELECT <kolonlar> from <tablo>  
ORDER BY <alan> [ASC|DESC]  
<alan>'a göre artan/azalan sıralamada  
seçim yapar
```

Örnek:

```
Select sicil,ad,soyad from personel  
order by burut;
```



Koşula Bağlı Sorgulama

```
SELECT <kolonlar> FROM <tablo>  
WHERE <alan><operatör><değer>
```

Örnek:

```
Select * from personel where ad='Koray';
```



Koşula Bağlı Sorgulama

- Operatörler

- < Küçük
- > Büyük
- = Eşit
- <= Küçük Eşit
- >= Büyük Eşit
- <> Eşit Değil



Aralık Sorgulaması

```
SELECT <kolonlar> FROM <tablo>WHERE  
<alan> BETWEEN <deger1> AND  
    <deger2>
```

<alan> değeri <deger1> ve <deger2>
arasındakileri listeler

Örnek:

```
Select * from personel  
where
```

```
Brut between 500 and 1000
```



Birden Fazla Tabloda Sorgu

```
SELECT <kolonlar> FROM <tablo1>  
,<tablo2> where  
<tablo1.alan1>=<tablo2.alan2>
```

```
SELECT <kolonlar> FROM <tablo1> JOIN  
<tablo2> ON  
<tablo1.alan1>=<tablo2.alan2>
```



Güncelleme İşlemi

UPDATE <tablo> SET <kolon>=<deger>,..

Örnek:

Update personel set brut=10000 where
ad='Koray'



Silme işlemi

DELETE FROM <tablo>

Örnek:

Delete from personel where ad='Koray'



Aritmetiksel İfadeler

- SUM `select sum(brut) from personel`
- AVG `select avg(net) from personel`
- MAX `select max(brut) from personel`
- MIN `select min(brut) from personel`
- COUNT `select count(*) from personel`



View Oluşturmak

- Veri güvenliği
- Sorgulamanın basitleştirilmesi
- Sadece view kullanılarak gerçekleştirilebilen sorgular



View Oluşturmak

```
CREATE VIEW <isim>(<kolonlar>) AS  
SELECT...
```

Örnek:

```
CREATE VIEW perslist(persno,persadi) AS  
Select persno,persadi from personel
```



Index Oluşturmak

```
CREATE INDEX <index adı>  
ON <tablo> (<kolon>,...)
```

Örnek:

```
Create index indxPers on personel  
(ad,soyad)
```




SON

Koray Toksöz

ktoksoz@aydinorme.com.tr