

---

---

# VERİTABANI YÖNETİM SİSTEMLERİ - MYSQL

Onur Can KARAMAN - Cihad ARI

---

# Genel bakış

1. Analiz
2. Kavramsal ve Mantıksal Model
3. MYSQL
4. Fiziksel Model

---

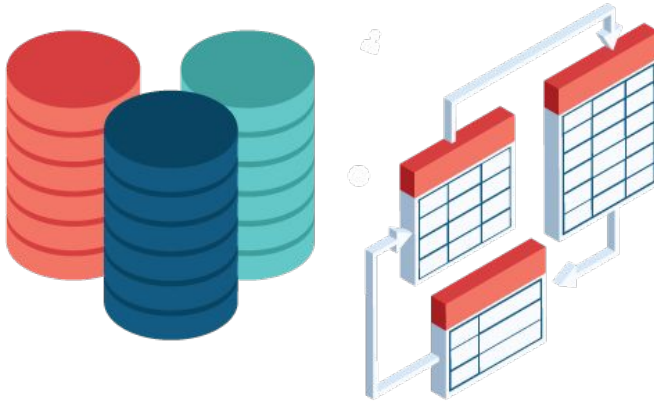
---

# Veri Tabanı Nedir

---

---

# Veri Tabanı



**Verilerin Saklanması  
iin ortam sunan ayrı  
Bir Yazılım mı?**

**Verilerin Saklandığı  
Yer Mi ?**

---

---

# Veri Tabanı Nedir?

---

© Original Artist  
Reproduction rights obtainable from  
[www.CartoonStock.com](http://www.CartoonStock.com)



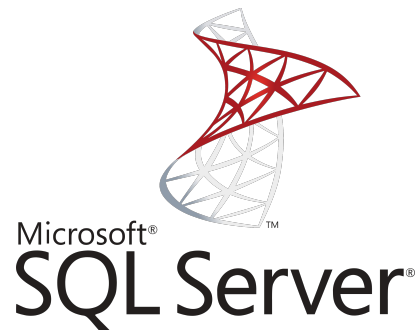
"Ey Veritabanı Yöneticisi,  
bize doğru yolu göster!"

Kullanım amacına göre  
düzenlenmiş verilerin  
saklandığı bilgi depolarıdır

---

# Veri Tabanı Yönetim Sistemleri

---



BigQuery

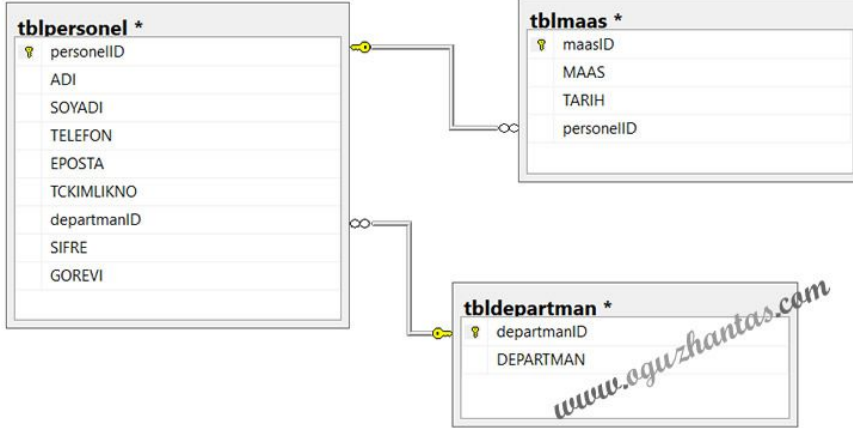




---

# Temel Kavramlar

---



TABLO  
ALAN  
KAYIT  
VERİ TÜRÜ  
ANAHTAR  
YABANCI ANAHTAR

---

# Veri Tabanı Tasarlama Aşamaları

---



## ANALİZ

NE YAPIYORUZ ?

NELER OLACAK ?

MÜŞTERİ NE İSTİYOR ?

NELER SAKLAYACAĞIZ ?

## KAVRAMSAL MODEL

ÇİZ

KARALA

ŞEKİLLER

## MANTIKSAL MODEL

KALEM KAĞIT İLE YAZ.

## FİZİKSEL MODEL

PROGRAMLAMAYA BAŞLA



---

# 1-ANALİZ

İŞİN HİKAYESİ...

---



Bir okul sistemi yapalım.  
İçerisinde okul, öğretmen,  
ve sınıf sistemi üzerine  
çalışacağız.

Şimdi müşteriye gidelim...

---

# **2-KAVRAMSAL MODEL**

---





Şimdi de bu verileri  
görselleştirelim. Önemli kurallara  
**dikkat et.**

---

---

# 3-MANTIKSAL MODEL

---



Şu grafiği yazılı bir  
hale getirelim.

---

# 4-FİZİKSEL MODEL

---

---

Cihad Hocam... Kolay Gelsinnnn :)  
Neyse fiziksel modeli ben halledeyim  
Sorgular sende ama :D

---

---

# MySQL

---



---

# Select

- Seçilen tablo veya tablolardan veri okumayı sağlar
    - ◆ `SELECT * FROM ogrenciler`
    - ◆ `SELECT * FROM ogretmenler`
    - ◆ `SELECT * FROM okullar`
-



---

# Select

- Seçilen tablo veya tablolardan veri okumayı sağlar
  - ❖ **SELECT** adi,soyadi **FROM** ogretmenler
  - ❖ **SELECT** okulNo,adi,soyadi **FROM** ogrenciler
-

# Order By

- Sonuçların sıralı şekilde gelmesini sağlar.
- ASC artan (A-Z,1-9),  
DESC azalan(Z-A, 9-1)  
şekilde sıralar.



---

# Order By

- Sonuçların sıralı şekilde gelmesini sağlar.
  - ASC artan (A-Z,1-9), DESC azalan(Z-A, 9-1) şekilde sıralar.
- 
- ❖ `SELECT * FROM ogrenciler ORDER BY adi`
  - ❖ `SELECT * FROM ogretmenler ORDER BY adi DESC`
  - ❖ `SELECT * FROM okullar ORDER By ilçe ASC`
-

---

# Distinct

- Tekrarlı verilerin gelmesine engel olur.
  - ❖ `SELECT DISTINCT adi FROM öğrenciler`
  - ❖ `SELECT DISTINCT soyadi FROM öğrenciler`
  - ❖ `SELECT DISTINCT dTarihi FROM öğrenciler`
-

---

# Where

- Sorgu koşulları belirtmek için kullanılır.
  - ❖ `SELECT * FROM ogrenciler WHERE okulNo=340`
  - ❖ `SELECT * FROM okullar WHERE ilçe="Altındağ"`
  - ❖ `SELECT * FROM ogrenciler WHERE dTarihi="2009-06-07 00:28:19.000"`
-

---

# And

- İki veya daha fazla şartın hepsinin sağlanması durumunda sonuç dönmesini sağlar.
  - ❖ `SELECT adi,soyadi , sinifi FROM ogrenciler WHERE sinifi=4 and okulNo<300`
-

---

# And

- İki veya daha fazla şartın hepsinin sağlanması durumunda sonuç dönmesini sağlar.
  - ❖ `SELECT adi, soyadi FROM ogrenciler WHERE adi="Kartal" and soyadi= "Küçük"`
-

---

# And

- İki veya daha fazla şartın hepsinin sağlanması durumunda sonuç dönmesini sağlar.
  - ❖ `SELECT * FROM ogretmenler WHERE id=2 and adi="Cihad"`
-



---

# Or

- İki veya daha fazla şarttan birinin sağlanması durumunda sonuç dönmesini sağlar.
  - ❖ `SELECT * FROM okullar WHERE şehir="İzmir" OR ilçe="Altındağ"`
-

---

# Or

- İki veya daha fazla şarttan birinin sağlanması durumunda sonuç dönmesini sağlar.
  - ❖ `SELECT okuTuru FROM okulturleri WHERE okuTuru="Meslek Lisesi" or okuTuru="Anadolu Lisesi" or okuTuru="İlkokul"`
-

---

# In

- İlgili kolon içerisinde arama yapmak için kullanılır.
  - ❖ **SELECT okuTuru FROM okulturleri WHERE okuTuru IN ("Meslek Lisesi" ,"Anadolu Lisesi","İlkokul")**
-

---

# In

- İlgili kolon içerisinde arama yapmak için kullanılır.

❖ `SELECT * FROM ogretmenler WHERE id IN (3,4,1)`

---

---

# In

- İlgili kolon içerisinde arama yapmak için kullanılır.
  - ❖ **SELECT \* FROM ogrenciler WHERE adi IN ("Barış","Ali","Ayşe","Serçe")**
-

---

# Like

- İlgili kolondaki kayıt içerisinde arama yapmak için kullanılır.
  - ❖ `SELECT * FROM ogrenciler WHERE adi LIKE "_a%"`
  - ❖ `SELECT * FROM ogrenciler WHERE adi LIKE "%r%"`
  - ❖ `SELECT * FROM ogrenciler WHERE adi LIKE "%r"`
  - ❖ `SELECT * FROM ogrenciler WHERE adi LIKE "%r_"`
  - ❖ `SELECT * FROM ogrenciler WHERE dTarihi LIKE="2009%"`
-

---

# Between

- Değer aralıklarını belirlemek için kullanılır.
  - ❖ **SELECT \* FROM ogrenciler WHERE okulNo BETWEEN 100 AND 500**
-

---

# Between

- Değer aralıklarını belirlemek için kullanılır.
  - ❖ **SELECT \* FROM ogrenciler WHERE adi BETWEEN "Ali" and "Hüsamettin"**
-



---

# Between

- Değer aralıklarını belirlemek için kullanılır.
  - ❖ **SELECT \* FROM ogrenciler WHERE dTarihi BETWEEN "0000-00-00 00:00:00.000" and "2004-12-16 00:00:00.0"**
-

---

# Limit

- Döndürülecek değer sayısı belirtir. Başlangıç satırı ve gösterilecek satır sayısı değerleri alır.
  - ❖ `SELECT * FROM ogrenciler LIMIT 10`
  - ❖ `SELECT * FROM ogrenciler LIMIT 5,10`
-

---

# Is Null

- ilgili sütunda veri olmayan kayıtlar getirilir.
  - ❖ `SELECT * FROM ogrenciler WHERE dTarihi IS NULL`
-

---

## Is Not Null

- ilgili sütunda veri olan kayıtlar getirilir.

❖ `SELECT * FROM ogrenciler WHERE dTarihi IS NOT NULL`

---

---

# Inner Join

- İki tabloda bulunan verileri belirtilen kolondaki değerler ile birleştirerek iki tablonun ortak kayıtları getirilir.
  - ❖ `SELECT adi, soyadi FROM ogrenciler INNER JOIN kademeler ON ogrenciler.sinifi=kademeler.id`
-

---

# Inner Join

- İki tabloda bulunan verileri belirtilen kolondaki değerler ile birleştirerek iki tablonun ortak kayıtları getirilir.
  - ❖ `SELECT adi, soyadi, kademeler.sube, kademeler.kademe  
FROM ogrenciler INNER JOIN kademeler ON  
ogrenciler.sinifi=kademeler.id`
-

---

# Left Join

- Birleştirilmek istenilen iki tablodan birinci tablonun bütün verileri, ikinci tablodan ise sadece birinci tabloda ilgili kolon ile eşleşen veriler yer alır.
  - ❖ `SELECT adi, soyadi FROM ogretmenler LEFT JOIN sinifogretmenleri ON ogretmenler.id=sinifogretmenleri.ogretmen`
-

---

# Left Join

- ❖ **SELECT adi, soyadi, sinifogretmenleri.ogretmen FROM ogretmenler LEFT JOIN sinifogretmenleri ON ogretmenler.id=sinifogretmenleri.ogretmen**
  - ❖ **SELECT adi, soyadi, sinifogretmenleri.ogretmen FROM ogretmenler LEFT JOIN sinifogretmenleri ON ogretmenler.id=sinifogretmenleri.ogretmen where ogretmenler.id >3**
-



---

# Right Join

- Left joinin tam tersi işlemi yapar.
  - ❖ `SELECT *, okulturleri.* FROM okullar RIGHT JOIN okulturleri on okullar.turu=okulturleri.id`
-

---

## As

- Seçilen tablo adı veya tablo sütunlarına sorguda takma ad verir
  - ◆ **SELECT adi, soyadi, b.sube, b.kademe FROM ogrenciler as a INNER JOIN kademeler as b ON a.sinifi=b.id**
-

---

# Group By

- Sorgunun grup halinde tekrarsız olarak gelmesini sağlar
  - ❖ `SELECT dTarihi, count(*) FROM ogrenciler GROUP BY dTarihi`
  - ❖ `SELECT dTarihi, MIN(okulNo) FROM ogrenciler GROUP BY dTarihi`
-

---

# Insert

- Tabloya kayıt veya kayıtlar eklemek için kullanılır.
  - ❖ **INSERT INTO** ogrenciler **VALUES**("","1234","Yeni","Öğrenci","2007-09-03 00:00:00.000",19)
  - ❖ **INSERT INTO** ogrenciler(okulNo, adi, soyadi, dTarihi, sinifi) **VALUES**("1357","Yeni","Öğrenci3","2009-09-03 00:00:00.000",20)
-

---

# Update

- Tablodaki kayıt veya kayıtları güncellemek için kullanılır.
  - ◆ **UPDATE** sinifogretmenleri SET sinif=5 where id=1
  - ◆ **UPDATE** ogrenciler SET okulNo=1235, soyadi="ŞİMŞEK" where id=27
-

---

# Delete

- Tablodaki kayıt veya kayıtları siler.

❖ **DELETE FROM** ogrenciler where id=26

❖ **DELETE FROM** ogrenciler where id=28 OR id=29

---

---

# Avg()

- ilgili sütunun ortalamasını alır.
  - ❖ `SELECT Avg (okulNo) FROM ogrenciler`
  - ❖ `SELECT Avg (okulNo) FROM ogrenciler where id>10`
-

---

# Count()

- **Kayıt sayısı döndürür.**
  - ◆ **SELECT count(\*) FROM ogrenciler**
  - ◆ **SELECT count(\*) FROM ogrenciler where dTarihi BETWEEN "2004-01-01" AND "2006-01-01"**
-



---

# Max()

- İlgili sütunda en büyük kaydı getirir.
  - ❖ **SELECT MAX(okulNo) FROM ogrenciler**
  - ❖ **SELECT MAX(adi) FROM ogrenciler**
-

---

# Min()

- İlgili sütunda en küçük kaydı getirir.
  - ❖ **SELECT MIN(okulNo) FROM ogrenciler**
  - ❖ **SELECT MIN(adi) FROM ogrenciler**
-

---

# Sum()

- İlgili sütundaki kayıtları toplar.

❖ **SELECT SUM(okulNo) FROM ogrenciler**

---

---

# Alter Table

## Add

- Tabloya sütun eklemek için kullanılır.
  - ❖ `ALTER TABLE öğrenciler ADD yorum VARCHAR(255)`
-

---

# Alter Table

## Modify

- Tablodaki mevcut sütunun özelliklerini değiştirir.

❖ **ALTER TABLE** öğretmenler **MODIFY** yorum int

---

---

# Alter Table

## Change Column

- Tablodaki mevcut sütunun adını değiştirir.
  - ❖ **ALTER TABLE** öğretmenler **CHANGE COLUMN** yorum  
gorev\_suresi int
-

---

# Alter Table

## Rename To

- Tablonun adını değiştirir.

❖ **ALTER TABLE** atamalar **RENAME TO** eski\_gorevler

---

---

# Alter Table

## Drop Column

- Tablodaki mevcut sütunu siler.
  - ❖ **ALTER TABLE** öğretmenler **DROP COLUMN** gorev\_suresi
-



---

# Alter Table

## Drop Table

- Mevcut tablonun silinmesi için kullanılır.
- ❖ DROP TABLE eski\_gorevler

