

# Veri Tabanı

3. Hafta Dersi



# Dersin Hedefleri

- Veri Tabanı Kullanıcıları
- Veri Modelleri
- Veri Tabanı Tasarımı
- İlişkisel VT Kavramsal Tasarımı (Entity-Relationship, ER) Modeli
- Kavramsal Tasarımın Tablolara Dönüştürülmesi

# VT KULLANICILARI

1. Veri Tabanı Yöneticisi (DBA)
2. Uygulama Programcısı
3. Sorgu Dili Kullanıcıları
4. Son Kullanıcılar

# Veri Tabanı Yöneticisi (DBA)

## Görevleri;

- **VT Tasarımı ve Oluşturulması**

- Kavramsal şema + fiziksel olarak yapılandırılması

- **VT Performans Analizi**

- VT ayarları, donanım ve yazılım ihtiyaçlarının belirlenmesi, performans arttırıcı işlemler

- **Erişim Yetkilerinin Düzenlenmesi**

- Kullanıcıların VT erişebilirliği, yetkisiz erişimlerin düzenlenmesi

# Veri Tabanı Yöneticisi (DBA)

## Görevleri;

- **Yedekleme ve Geri Yükleme**
  - Süreçlerin belirlenmesi, yedekleme ve geri yüklemenin güvenilirliği
- **Veri Bütünlüğü Sağlama**
- **Sistemin Sürekliliğini Sağlama**
  - Sorunlara hazırlıklı olma, Kötü durum senaryolarını oluşturma

# Uygulama Programcısı

- Son kullanıcılara yönelik yazılım geliştiriciler.
- VT direk kullanmazlar.
- Veri işleme dili ve geliştirme araçlarını kullanırlar.



# Sorgu Dili Kullanıcıları

- Veri tabanından bilgi almak
- Bilgi eklemek/silmek
- VT yapısında deęişiklik yapmak
- VT yapısını bilmelidir





# Son Kullanıcılar

- Uygulama programcıları tarafından geliştirilen yazılım üzerinde kendilerinden istenen işlemleri gerçekleştirirler.

Kayıt ekleme, silme, raporlama vb...

**Not:** Bu kullanıcıların üst düzey bir bilgisayar bilgisine ya da veritabanı yapısı hakkında bilgi sahibi olmasına gerek yoktur.





# Veri Modelleri

- Verileri mantıksal düzeyde düzenlemek için kullanılan yapılar, kavramlar ve işlemler topluluğudur.
- Her VTYS belirli bir veri modeli kullanır.
- Veriler ve veriler arası ilişkiler veri modeline göre düzenlenir.
- Veri modeli ile VT kavramsal ve dış şemaları oluşturulur.

# Veri Modelleri

1. Hiyerarşik Veri Modeli
2. Ağ Veri Modeli
3. İlişkisel Veri Modeli
4. Nesneye Yönelik Veri Modeli

# 1. Hiyerarşik Veri Modeli

İlk veri modelidir.  
(1960 – 70)

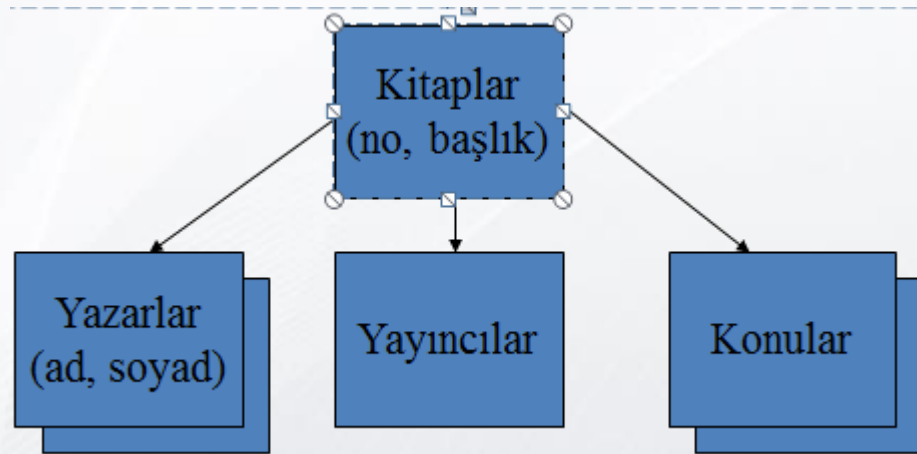
Kayıtlar ağaç yapısında saklanır.

Ebeveyn-Çocuk ilişkisi

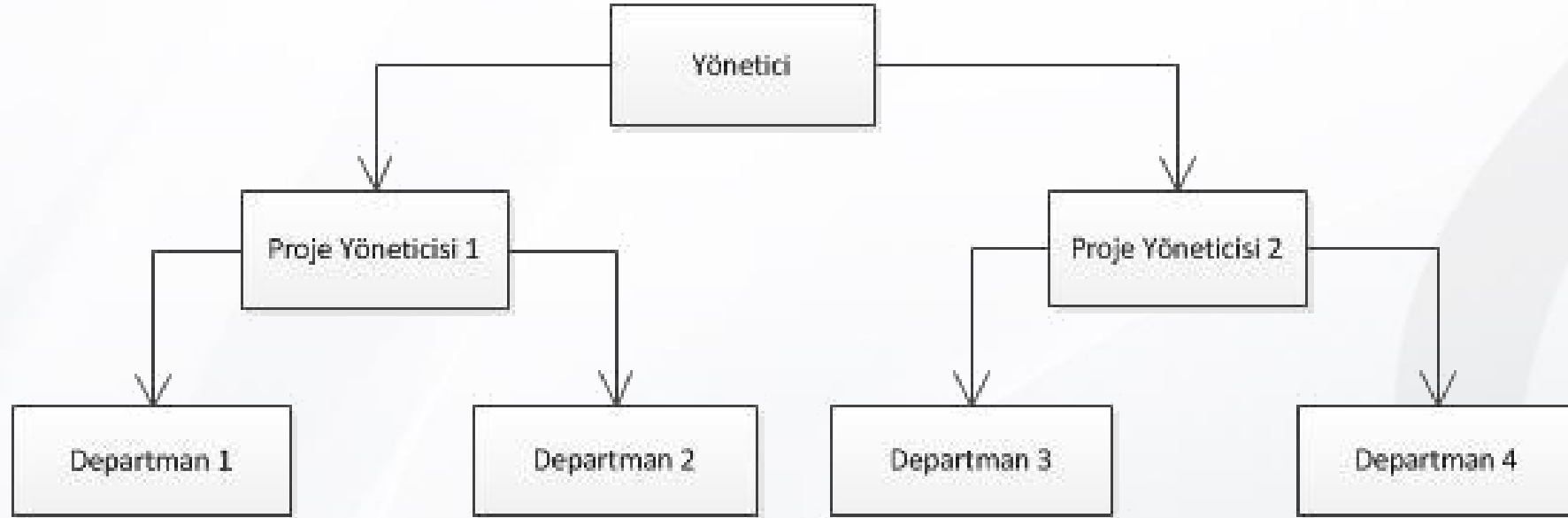
Her varlık bir sefer ilişki oluşturabilir.

# 1. Hiyerarşik Veri Modeli

- Aynı varlık 1 den fazla ilişkide kullanılacaksa tekrar oluşturulmalı.
- Gereksiz veri tekrarına sebep olur.
- Bir kayıt silinirse altındaki dallar da silinir.

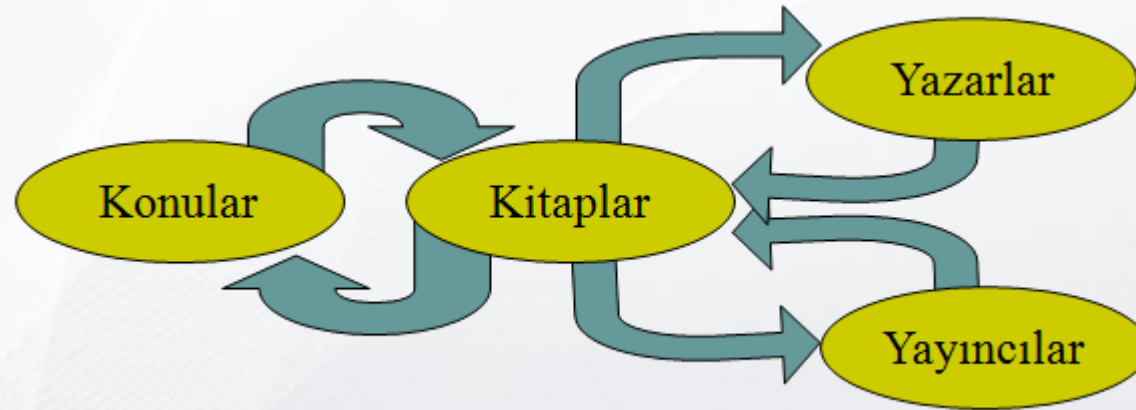


# 1. Hiyerarşik Veri Modeli



## 2. Ağ Veri Modeli

- 1970 – 80 arası kullanılmıştır.
- Hiyerarşik modelden farklı olarak bir veri birden fazla veri ile ilişki kurabilmektedir.
- Ortak bilgi alanları ile birden fazla veri ile ilişki kurulabilir.



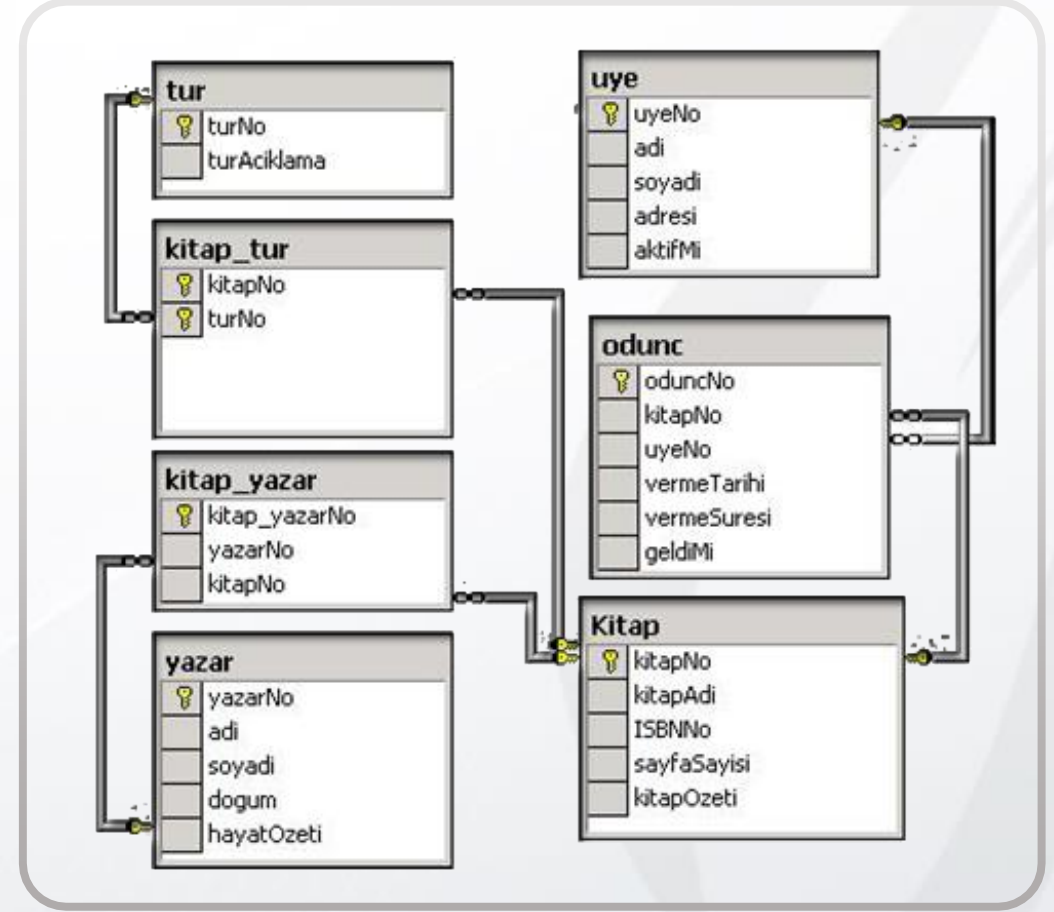
## 3. İlişkisel Veri Modeli

- 90 lı yıllardan beri hemen hemen tüm sistemlerde kullanılmaktadır.
- Ortak özelliklerdeki veriler tablolarda saklanır.
- Tablo = Sütun + Satır
- Her sütun farklı bir veri tutar.
- Her satırdaki veriler aynı kaydı ifade eder.



## 3. İlişkisel Veri Modeli

- Anahtarlar sayesinde indeksleme yapıldığı için işlemler daha hızlı gerçekleşir.
- Birden fazla tablodaki bilgiler ortak alanlar tanımlanarak birbirleriyle ilişkilendirilir.
- Raporlama kolaylığı sağlanır.



## 3. İlişkisel Veri Modeli

 1-1 ilişki

 1-n ilişki

 M-n ilişki

## 4. Nesneye Yönelik Veri Modeli

- Son 10 yıldır gündemde.
- Nesneye yönelik model C++ gibi nesneye dayalı bir dille yazılan ve yine nesneye dayalı bir dille kullanılan veritabanı anlamına gelir.
- Henüz yaygın olarak kullanılmamaktadır.
- Aslında OODBMS Nesne yönelimli programlama(OOP) ilkeleri ile Veri Tabanı Yönetim ilkelerinin birleşmesinden oluşur.

## 4. Nesneye Yönelik Veri Modeli

- Veriler nesne olarak modellenir. Bu modelde bir sorgunun karşılığında mutlaka önceden tanımlanmış bir nesne kümesi olmalıdır.
- Sınıf ve miras kavramları mevcuttur. (Sınıf kavramı ilişkisel modeldeki ilişki kavramına denk gelir)
- Büyük tablolarda ilişkisel veri modeline göre yüksek performans sunar.
- Arama işlemleri ilişkisel modele göre çok daha hızlıdır ancak çalışma mantıkları tamamen farklıdır.

# İlişkisel Veri Tabanı Tasarımı

- Bir projenin tasarım aşamasında veri tabanının tasarımı çok önemlidir.
- Veri tabanı oluşturulduktan sonra yapılacak düzenlemeler daha zor olabilir hatta sorunlara yol açabilir.

## 1. Gereksinim Analizi

- Mevcut sistemin özellikleri ve ihtiyaçların belirlenmesi
- Yazılım ve donanım maliyet analizi

# İlişkisel Veri Tabanı Tasarımı

## 2. Kavramsal Tasarım

- Varlıklar ve özniteliklerin belirlenmesi
- İlişkilerin belirlenmesi
- Varlık-İlişki veri modellerinin çıkarılması

# İlişkisel Veri Tabanı Tasarımı

## 3. Mantıksal Tasarım

- Varlık – İlişki modelini kullanarak mantıksal modelin oluşturulması
- Tablolar
- Satır ve Sütunlar
- Anahtarlar

## 4. Normalizasyon

- Mantıksal modele normalizasyon kurallarının uygulanması
- İlişkilerden yeni tabloların oluşturulması



# İlişkisel Veri Tabanı Tasarımı

## 5. Veri Tabanının Oluşturulması

- Tabloların oluşturulması
- Güvenlik ile kullanıcı yetkilerinin tanımlanması
- Test verileri ile denemeler yapılması

## 6. Uygulama Programı

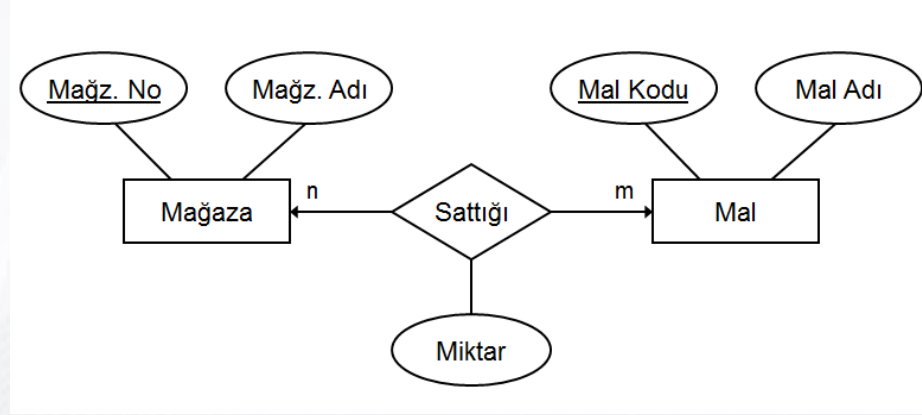
- Kullanım durumlarının belirlenerek programla aşamasına geçilmesi

# İlişkisel Veri Tabanı Kavramsal Tasarım

- VT saklanacak verilerin üst seviyede gösterilmesi için kullanılır.
- ER (Entity-Relationship, Varlık-İlişki) Modeli
- VTYS den bağımsız modelleme yapılır ve ilişkiler tanımlanır.
- Herhangi bir VTYS ile veri tabanına dönüştürülebilir.

# Varlık – İlişki Modeli

- Veri çözümleme ve modellemede ilişkilerin ortaya konması için kullanılan araçtır.
- Niteliklerin kümesi tarafından tanımlanır.
- Görsel olarak varlık-ilişki tablosu ile gösterilir:





CELAL BAYAR  
ÜNİVERSİTESİ  
UZAKTAN EĞİTİM

TEŞEKKÜRLER...

[www.aliosmangokcan.com](http://www.aliosmangokcan.com)

[mail@aliosmangokcan.com](mailto:mail@aliosmangokcan.com)

**KAYNAK: Turgut ÖZSEVEN- VTYS1**