

**Celal Çeken**

**Veysel Harun Şahin**

# **Veritabanı Yönetim Sistemleri**

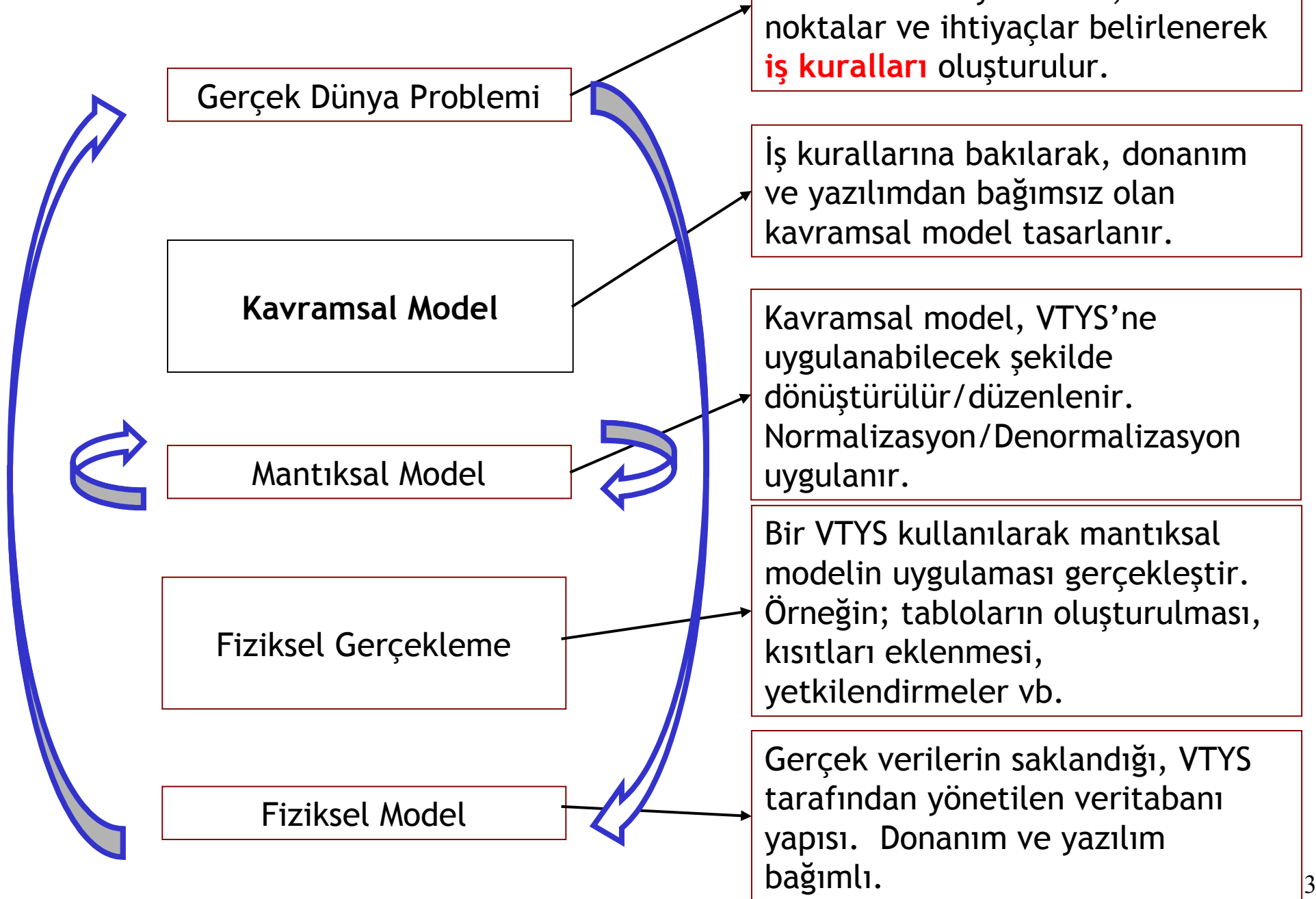
**(Veritabanı Tasarımı)**  
**Varlık Bağintı Modeli**



# Konular

- ✓ Veritabanı Tasarım Aşamaları
- ✓ Kavramsal Tasarım
- ✓ Temel Kavramlar
- ✓ Varlıklar Arası Bağlantılar
- ✓ Var Olma Bağımlılığı (Existence Dependency)
- ✓ Zayıf/Güçlü Bağlantılar
- ✓ Bağıntı Dereceleri
  - ✓ Tekli (Unary) Bağıntı
  - ✓ İkili (Binary) Bağıntı
  - ✓ Üçlü (Ternary) Bağıntı
- ✓ VBM (Varlık Bağıntı Modeli - ERM) ile Tasarım
- ✓ Kaynaklar

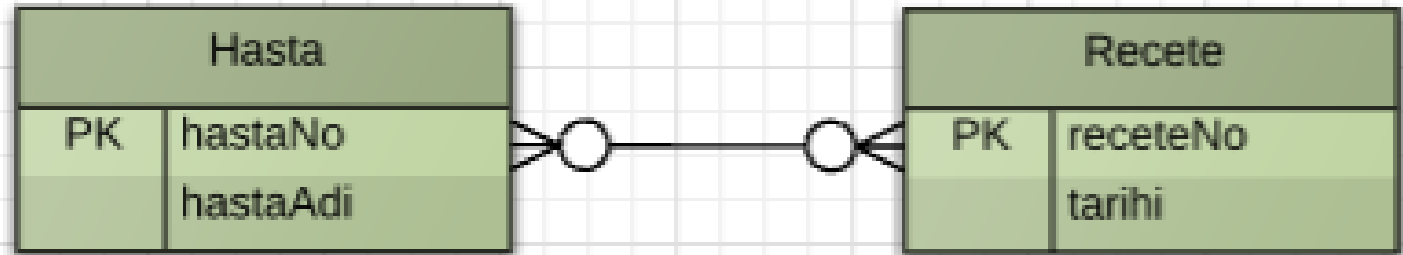
# Veritabanı Tasarımı Yaşam Döngüsü



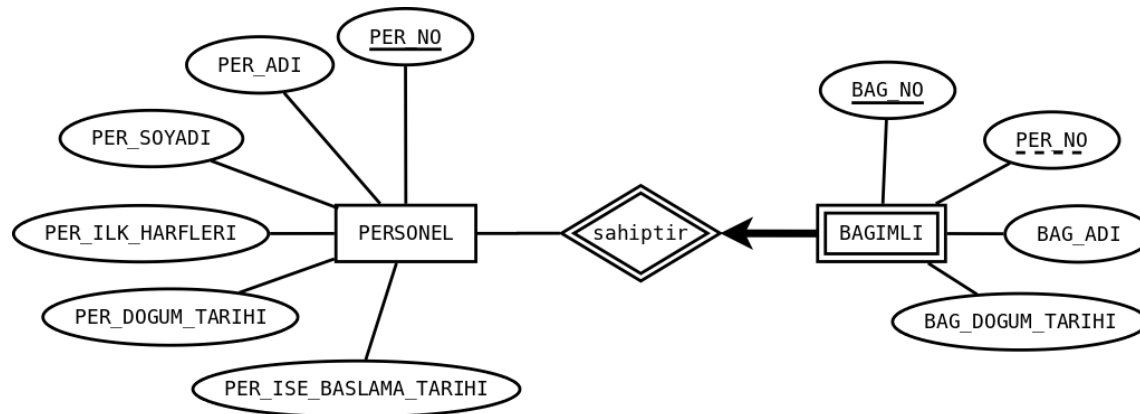
# Temel Kavramlar

- ✓ Varlık Bağlantı Diyagramı (VBD, Entity Relationship Diagram (ERD)) veritabanının kavramsal olarak modellenmesini sağlayan yazılım ve donanımlardan tamamen bağımsız gösterim şekilleridir.
- ✓ Farklı kullanıcılar arasında haberleşmeyi sağlamak için kullanılır.
- ✓ VB Modeline bakarak veritabanının tasarımını gerçekleştirmek daha kolaydır.
- ✓ VBD, veritabanının temel bileşenleri olan varlık (nesne ya da ilişki), özellik (nitelik), varlıklar arası bağıntılar (relationship) ve kısıtlardan meydana gelen çizelgelerdir.
- ✓ Chen notasyonu kavramsal modellemeyi ön plana çıkartır.
- ✓ Crow's Foot notasyonu ise daha çok uygulama (implementasyon) yönelimli yaklaşımı ön plana çıkartır.
- ✓ UML notasyonu, hem kavramsal hem de uygulama (implementasyon) modelleme yöntemleri için kullanılabilir.

**Crow's Foot Gösterimi**



**Chen Gösterimi**



# İş Kurallarını Varlık Bağntı Modeline Dönüştürme

- ✓ Genel olarak iş kurallarındaki isimler varlık, fiiller ise varlıklar arasındaki bağıntı olma adayıdır. Hakkında bilgi bulunan isim ya da isim tamlamaları varlık adayı iken, bilgi bulunmayanlar varlığa ait nitelik adayıdır.
- ✓ **Müşterinin** **ad**, **soyad**, **numara**, **adres** bilgileri saklanır.
- ✓ Bir **müşteri** çok sayıda **fatura** üretir.

İlişkiler iki yönlüdür.

1 öğretim üyesi **çok** sayıda (4) ders verebilir  
1 ders sadece 1 öğretim üyesi tarafından verilebilir.

1 kişi 1 bölüme yönetici olabilir.  
1 bölüm sadece 1 kişi tarafından yönetilebilir.

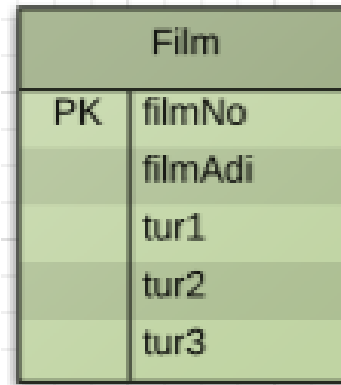
1 öğrenci **çok** sayıda derse kayıt yaptırabilir  
1 ders **çok** sayıda öğrenci tarafından alınabilir.

- ✓ İsimlendirme kuralları
  - Kodlama türleri
  - Linux Coding Style, Linus Torvalds
  - Hungarian Notation,
  - GNU Coding Standards
  - Java Coding Style Guide

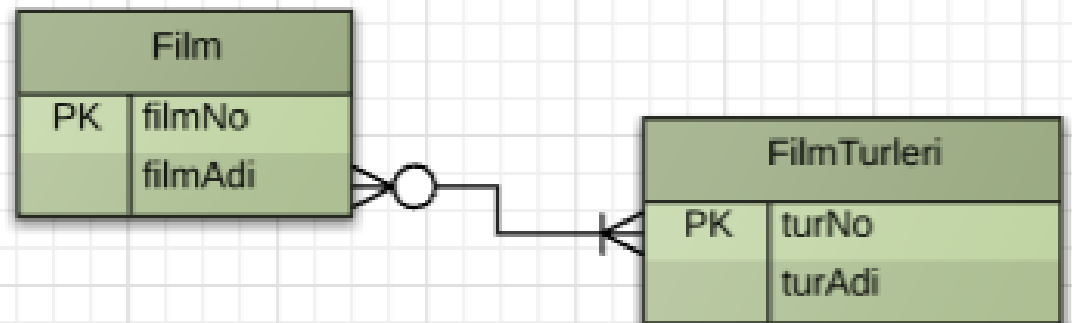
# Temel Kavramlar

- ✓ Birincil anahtarlar (Primary Key) tek bir alan olabileceği gibi birden fazla alanın birleşiminden de oluşabilir. (urunKodu, dersKodu + ogrenciNo)
- ✓ Birleşik özellikler: Adres (Cadde, Şehir, Ülke, posta kodu, ...) Detaylı sorgular için basit özellikli alanlara dönüştürülmeliler.
- ✓ Basit özellikler: Yaş, ad, soyad, cinsiyet, ...
- ✓ Çoklu değer alan özellikler: (Film türleri: Komedi, Macera, Korku, Komedi-Macera...)

Çözüm 1: Farklı alanlarda saklanabilir.



Çözüm 2: Yeni bir ilişki (tablo, varlık) oluşturulabilir.



# Temel Kavramlar

- ✓ Türetilmiş Özellikler: Değeri diğer özellikler kullanılarak oluşturulan özellikler.

```
SELECT AVG(AGE(dogumTarihi))  
FROM Kisiler
```

- ✓ Saklanmalı mı, hesaplanmalı mı?

# Temel Kavramlar

- ✓ Türetilmiş Özellikler: Değeri diğer özellikler kullanılarak oluşturulan özellikler.

```
SELECT AVG(AGE(dogumTarihi))  
FROM Kisiler
```

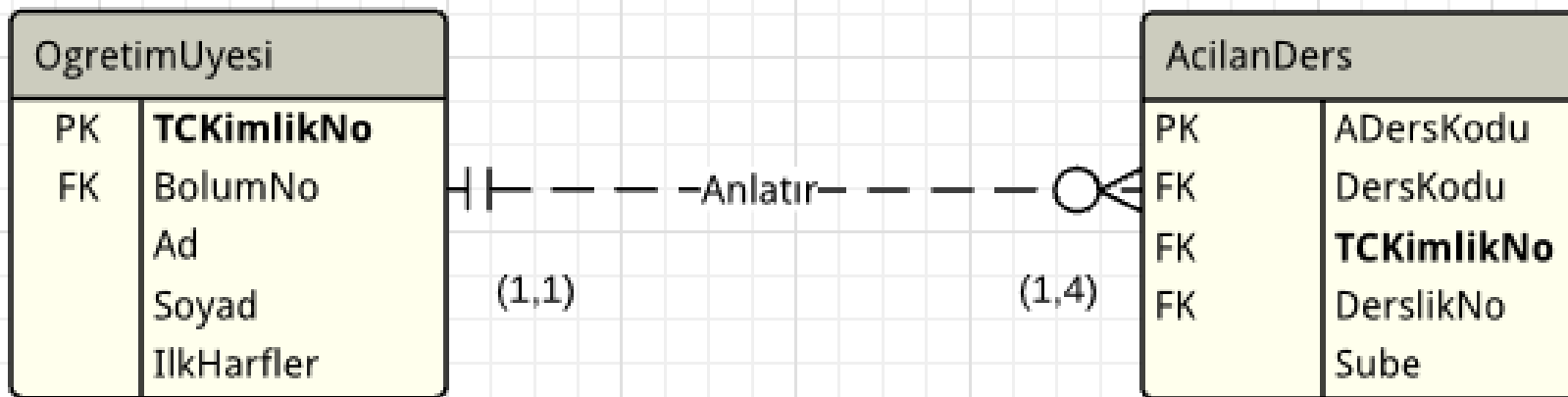
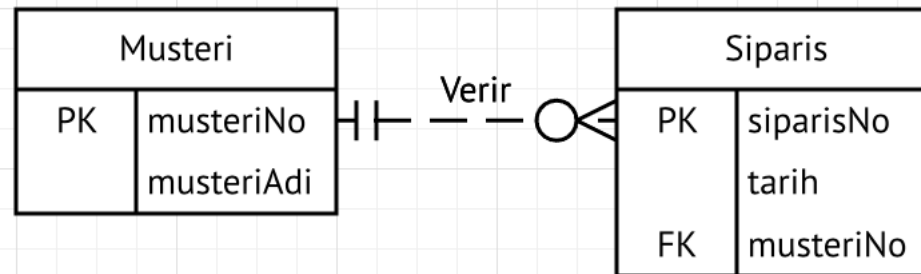
- ✓ Saklanması Durumunda:
  - + Az işlemci gücü gerekir, veriye daha hızlı erişim, geçmiş bilgisi için kullanılabilir
  - Güncel değer için sürekli denetlenmelidir, fazladan yer kaplar
- ✓ Hesaplanması Durumunda:
  - + Yer tasarrufu, her an güncel değer
  - Çok işlemci gücü gerekir, veriye daha yavaş erişim, sorgular daha karmaşık olur



## Varlıklar Arası Bağlantılar (1 - Çok Bağıntısı)

1 ders sadece **1 öğretim üyesi** tarafından verilebilir.

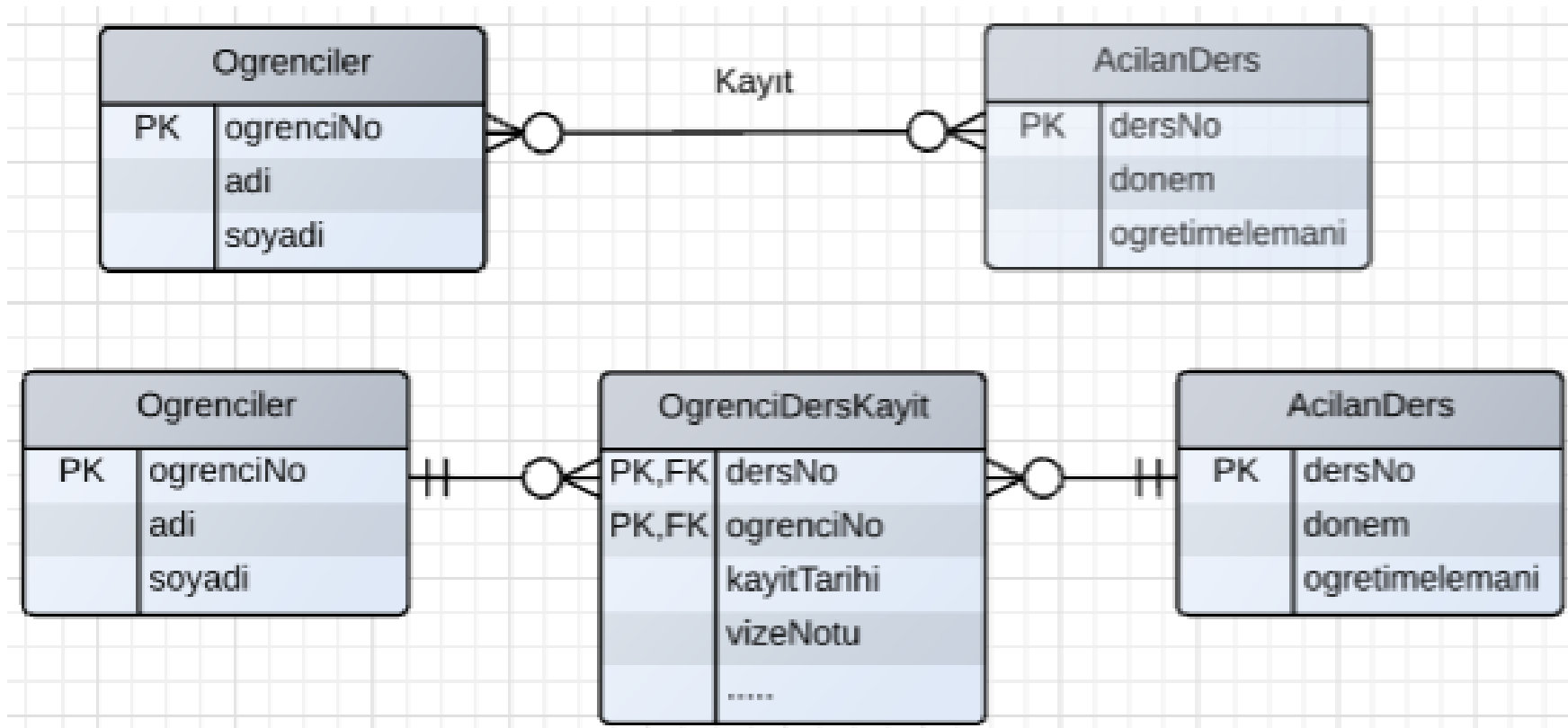
1 öğretim üyesi **çok** sayıda (4) **ders** verebilir



- ✓ Normal şartlarda tablo içerisindeki kayıt sayısı sınırlanamaz. Bunun için uygulama yazılımları ya da tetikleyiciler (trigger) kullanılabilir.
- ✓ Kayıt sayılarının gösterilmesi uygulama yazılımı geliştirilirken çok faydalıdır. Sınıfın açılabilmesi için en az 10 kayıt, en fazla 30 kayıt gibi.
- ✓ Sayılar iş kurallarına bakılarak belirlenir.

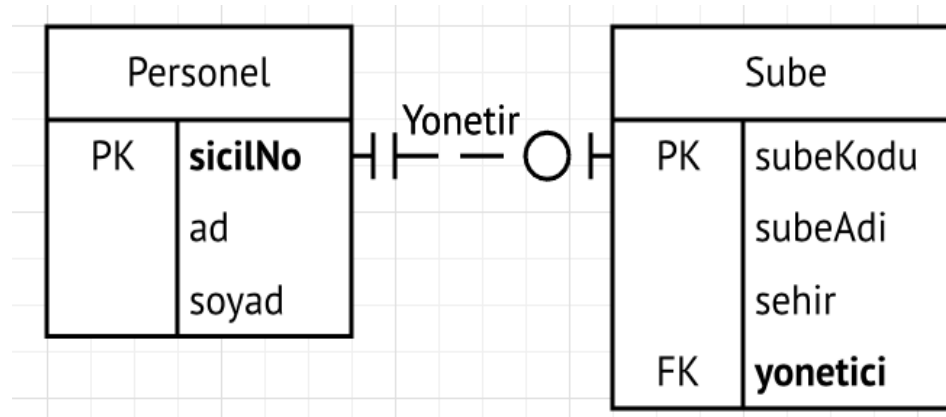
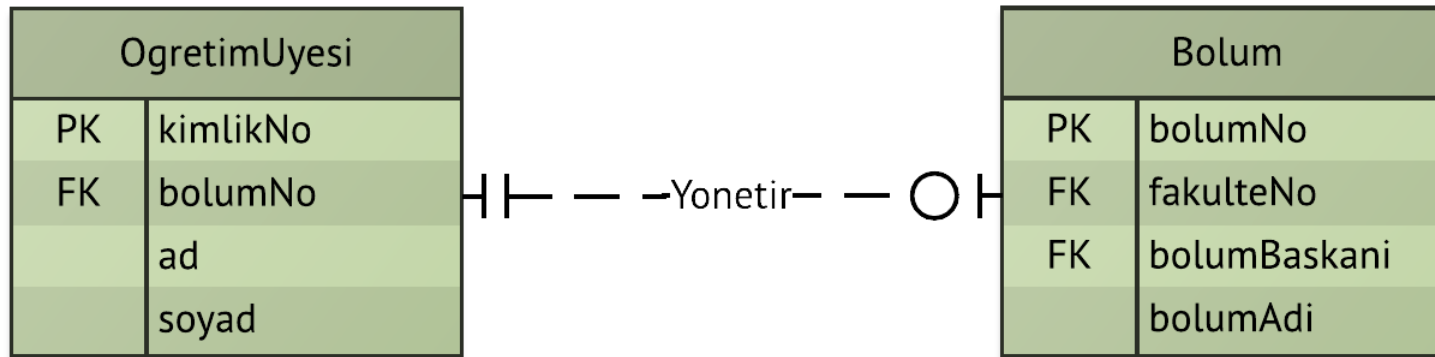
## Varlıklar Arası Bağlantılar (Çok - çok Bağıntısı)

- 1 öğrenci **çok** sayıda **derse** kayıt yaptırabilir  
1 ders **çok** sayıda **öğrenci** tarafından alınabilir.



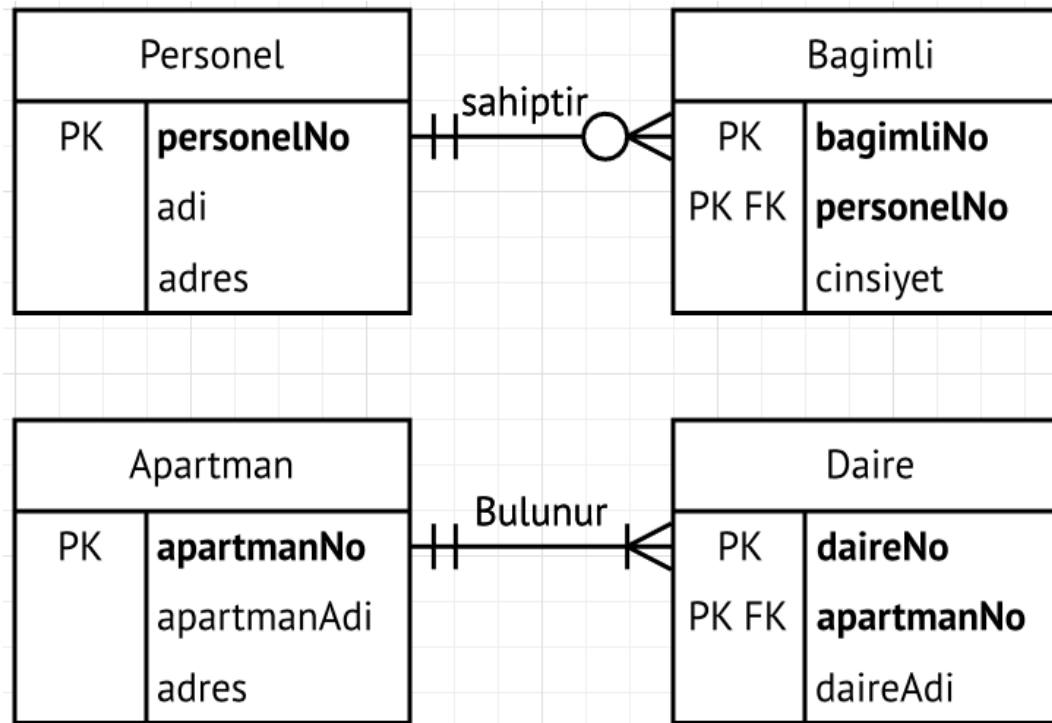
## Varlıklar Arası Bağlantılar (1 - 1 Bağıntısı)

- 1 öğretim üyesi 1 bölüm yönetebilir
- 1 bölüm 1 öğretim üyesi tarafından yönetilebilir.



## Var Olma Bağımlılığı (Existence Dependency)

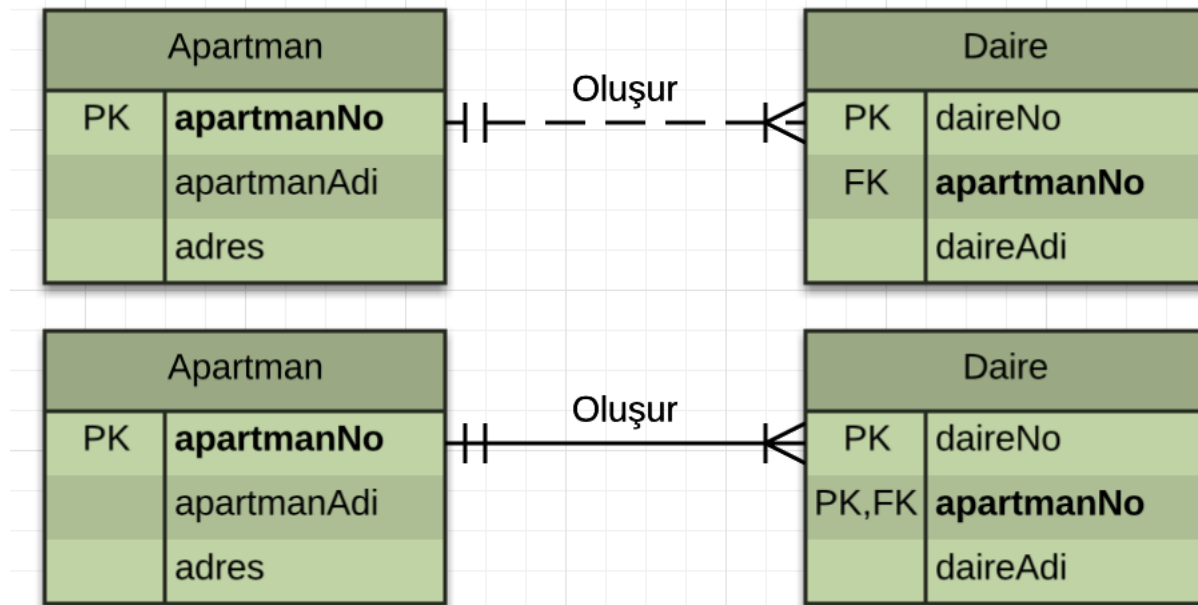
Bir personel bilgi sisteminde Personel ve Bagimli (Personele bagimli olan kişiler. Örneğin çocuk, eş vb.) varlık kümelerini düşünelim. Böyle bir sistemde «bir personele bağlı olmayan bagimli varlığı olamaz» kuralı mevcuttur. Örneğin bagimli varlık kümesine hiçbir personele ait olmayan bir çocuğun kaydını yapamayız, yapmamalıyız. Bu örnekte, bagimli ve personel arasında var olma bağımlılığı vardır denir. Bu durumda personel **üstün varlık**, bagimli ise **bağımlı varlıktır**.



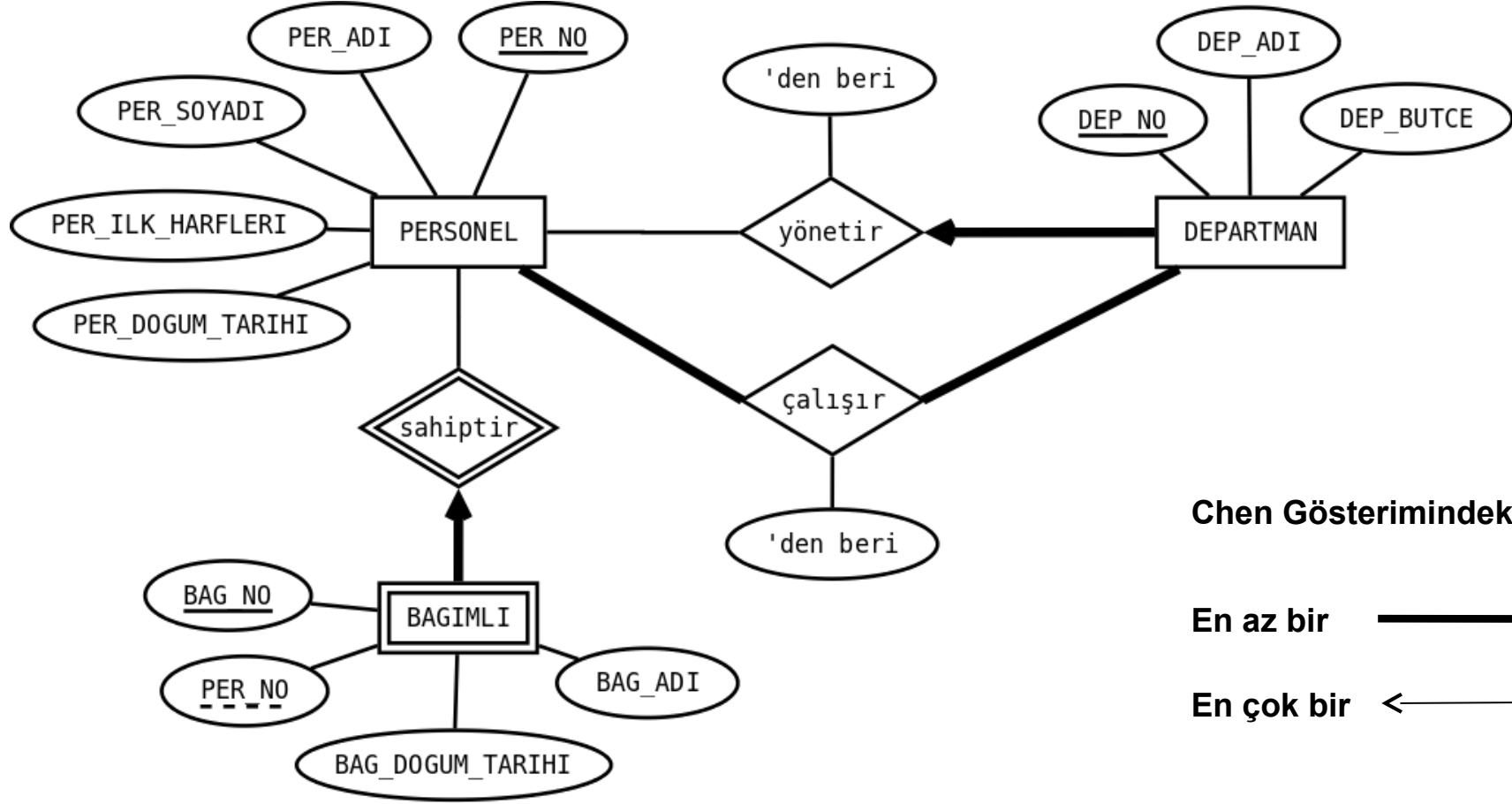
## Zayıf/Güçlü Bağıntılar

Bağıntı kurulan varlığın (Class) birincil anahtarı içerisinde, bağıntı kuran varlığın (Course) birincil anahtar bilgisi yer almıyorsa “iki varlık arasında **zayıf** bir ilişki vardır” denir.

Bağıntı kurulan varlığın (Class) birincil anahtarı içerisinde, bağıntı kuran varlığın (Course) birincil anahtarı yer alıyorsa “iki varlık arasında **güçlü** bir ilişki vardır” denir.



# Chen Gösteriminde Kısıtlar



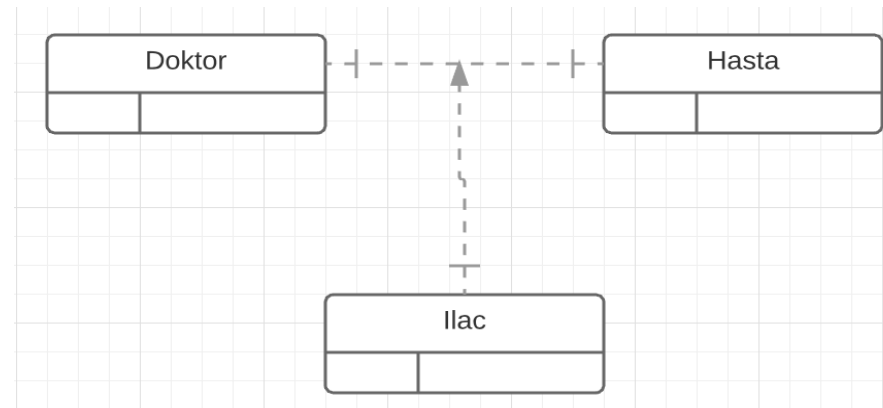
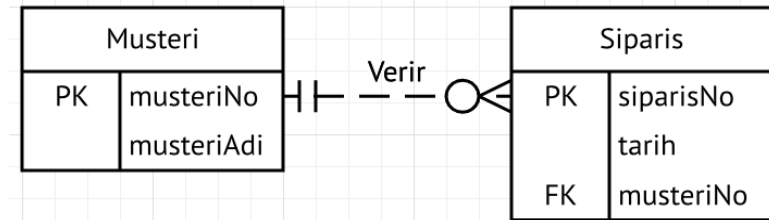
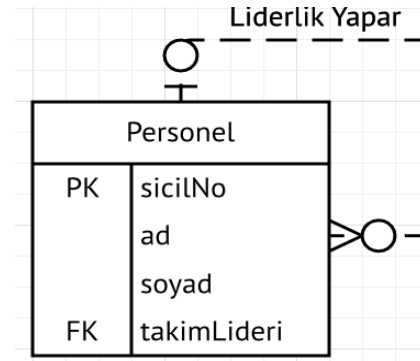
1 bölüm **en az, en çok 1** personel tarafından yönetilir.  
1 personel **çok** sayıda bölümü yönetebilir.

1 bölümde **çok** sayıda personel çalışabilir. **En az 1** personel çalışmak zorundadır.  
1 personel **çok** sayıda bölümde çalışabilir. **En az 1** bölümde çalışmak zorundadır.

1 çalışan **çok** bağımlıya sahip olabilir.  
1 bağımlı **en az, en çok 1** personele bağımlıdır.

## Bağıntı Dereceleri

- ✓ **Tekli (Unary) Bağntı:** Bir varlık kendisi ile bağıntılı (ilişkili) ise bu tür bir bağıntıya tekli bağıntı adı verilir. Örneğin, ÇALIŞAN tablosu içerisindeki bir çalışan, sıfır veya daha fazla çalışanın aynı zamanda yöneticisidir. Bir çalışanın sıfır ya da bir yöneticisi olmalıdır.
- ✓ **İkili (Binary) Bağntı:** İki varlığın bağıntısına (ilişkisine) ikili bağıntı denir.
- ✓ **Üçlü (Ternary) Bağntı:** Aynı anda 3 varlık birbirine bağlanıyorsa, bu tür bağıntıya üçlü bağıntı adı verilir. Kavramsal tasarımda her ne kadar 3 varlık mevcut ise de bunu gerçekleştirebilmek için 4. bir varlığa gereksinim duyulur.
  - ✓ Bir doktor sıfır veya çok reçete yazar.
  - ✓ Bir hasta sıfır veya çok reçete alabilir.
  - ✓ Bir ilaç, sıfır veya çok reçetede yer alabilir.



## VBM (Varlık Bağintı Modeli - ERM) ile Tasarım

Veritabanı tasarımı, lineer ya da ardışıl (yazılım geliştirmedeki şelale modeli gibi) olmaktan çok tekrarlı (yazılım geliştirmedeki spiral yöntem ya da iteratif yaklaşım gibi) bir süreçtir. Tekrar fiili, “tekrar tekrar yap” anlamındadır ve **tekrarlı bir süreç**, süreçlerin ve prosedürlerin tekrarlanması temeline dayanır. Bir varlık bağintı diyagramının geliştirilmesi genellikle aşağıdaki adımları içermektedir.

- ✓Organizasyonun (kurumun), işlerinin (operasyonlarının) tanımını içeren detaylı bir senaryo (hikaye) oluşturulur. Senaryo özellikle organizasyon içerisindeki rol temsilcilerine danışılarak oluşturulursa çok daha gerçekçi ve etkili olur.
- ✓Senaryoda geçen işlerin tanımları baz alınarak iş kuralları oluşturulur.
- ✓İş kuralları baz alınarak ana varlıklar ve varlıklar arasındaki bağintılar oluşturulur.
- ✓İlk varlık bağintı diyagramı geliştirilir.
- ✓Varlıkları net bir şekilde tanımlayan özellikleri ve birincil anahtarları oluşturulur.
- ✓Varlık Bağintı Diyagramı gözden geçirilerek gerekirse yukarıdaki adımlar, istenilen duruma gelinceye kadar tekrarlanır.



# Kaynaklar

- ✓ Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management, Cengage Learning.