

Veritabanı Yönetim Sistemleri

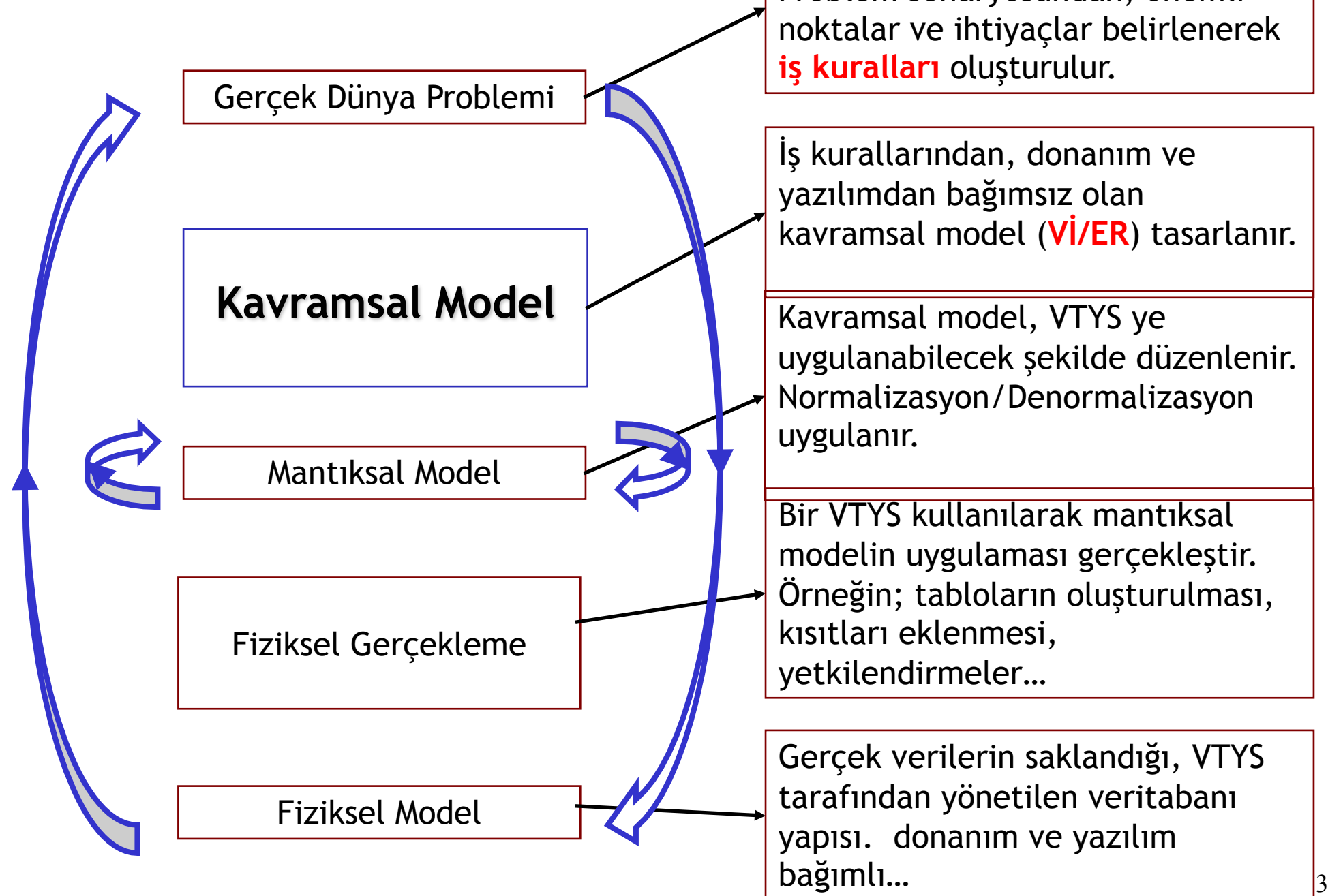
(Veritabanı Tasarımı)
Varlık İlişki Modeli



Konular

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

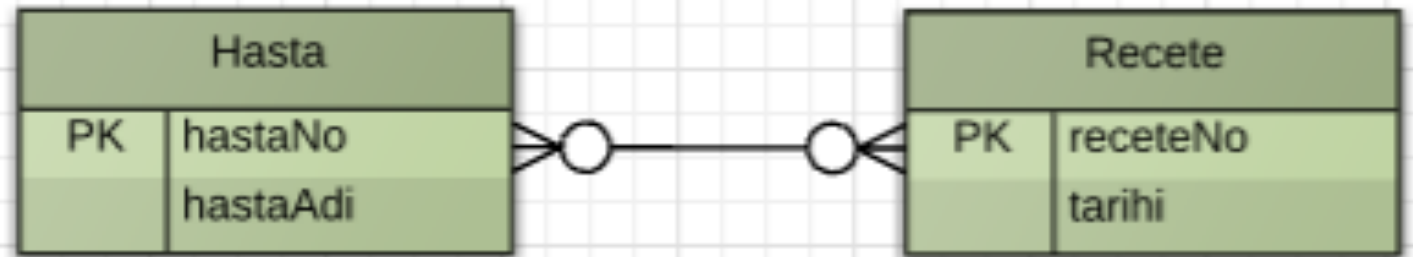
Veritabanı Tasarımı Yaşam Döngüsü



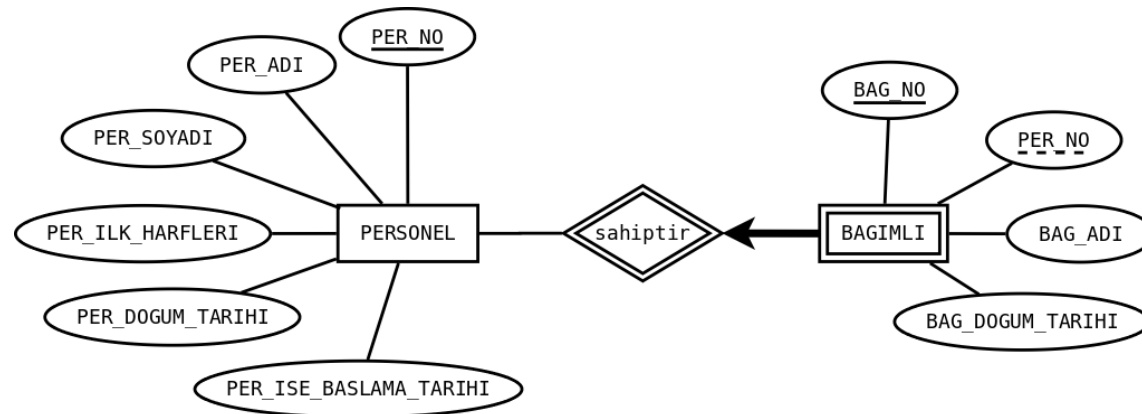
Temel Kavramlar

- ✓ Varlık İlişki Diyagramı (VID, Entity Relationship Diagram (ERD)) veritabanının kavramsal olarak modellenmesini sağlayan yazılım ve donanımlardan tamamen bağımsız gösterim şekilleridir.
- ✓ Farklı kullanıcılar arasında haberleşmeyi sağlamak için kullanılır.
- ✓ Vi Modeline bakarak veritabanının tasarımını gerçekleştirmek daha kolaydır.
- ✓ VID, veritabanının temel bileşenleri olan varlık (nesne ya da ilişki), özellik (nitelik), varlıklar arası ilişkiler 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 İlişki Modeline Dönüştürme

- ✓ Genel olarak iş kurallarındaki isimler varlık, fiiller ise varlıklar arasındaki ilişki 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.

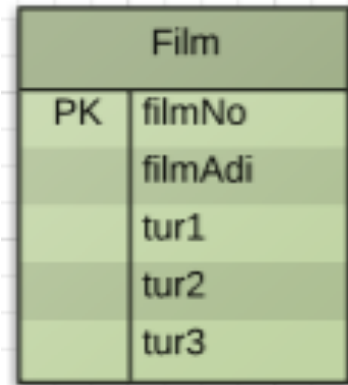
1 öğrenci **çok** sayıda derse kayıt yaptırabilir
1 ders **çok** sayıda öğrenci tarafından alınabilir.

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

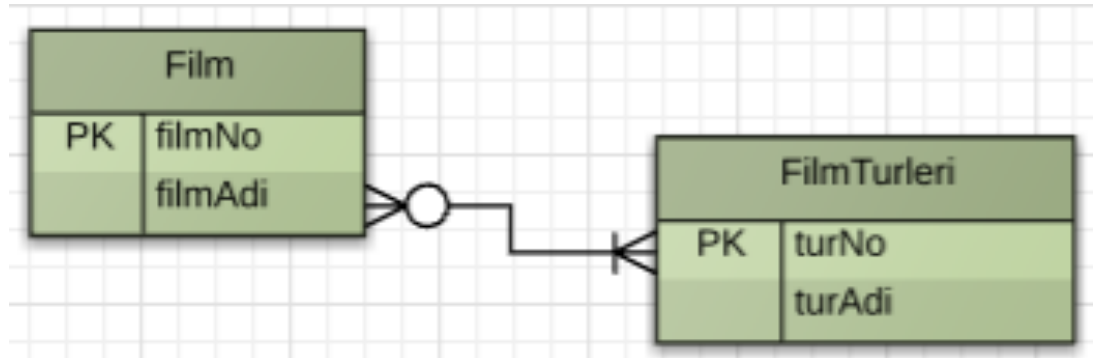
Temel Kavramlar

- ✓ 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
```

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 İlişkiler (1-Çok İlişkisi)

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

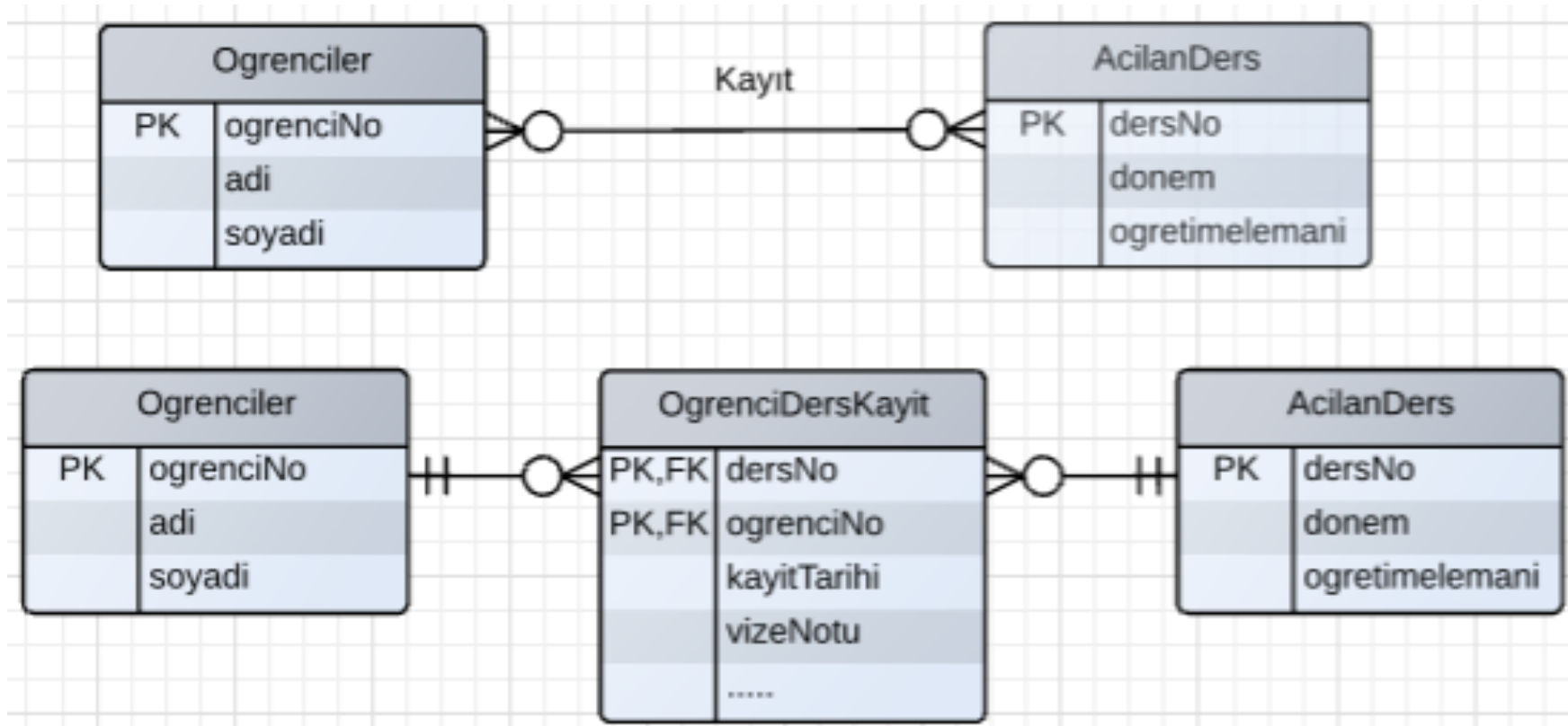
1 ders sadece **1 öğretim üyesi** tarafından verilebilir.



- ✓ 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 İlişkiler (Çok-çok ilişkisi)

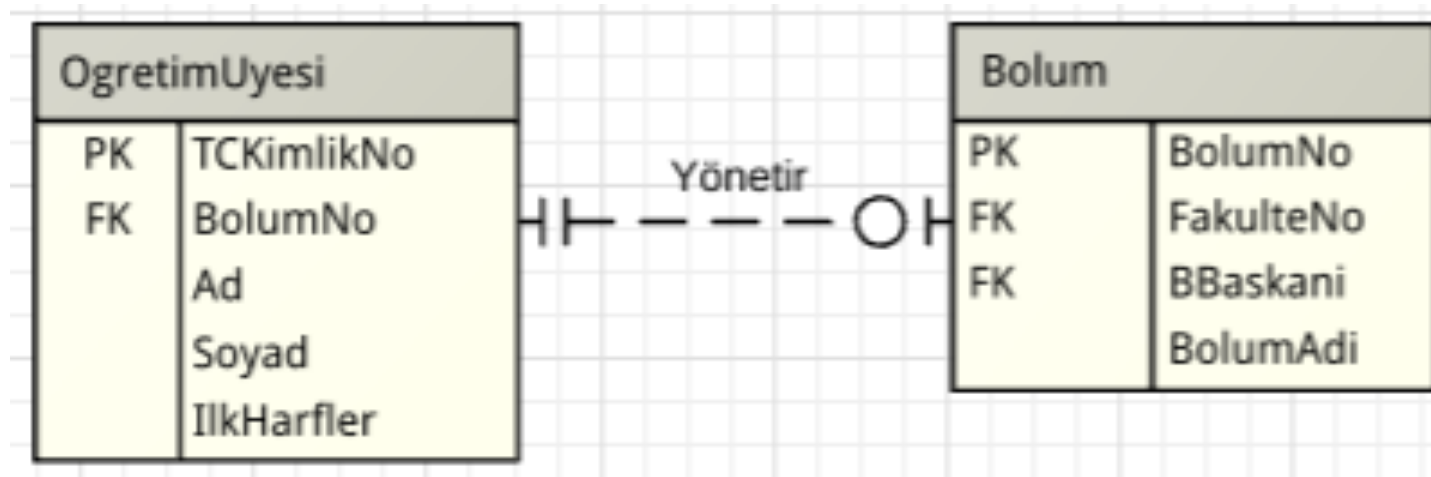
1 öğrenci **çok** sayıda derse kayıt yaptırabilir
1 ders **çok** sayıda öğrenci tarafından alınabilir.



Varlıklar Arası İlişkiler (1 - 1 ilişkisi)

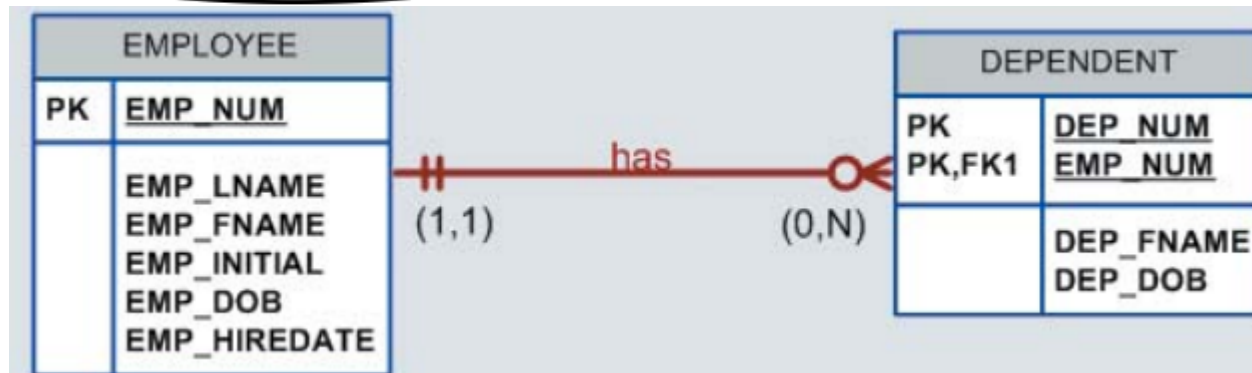
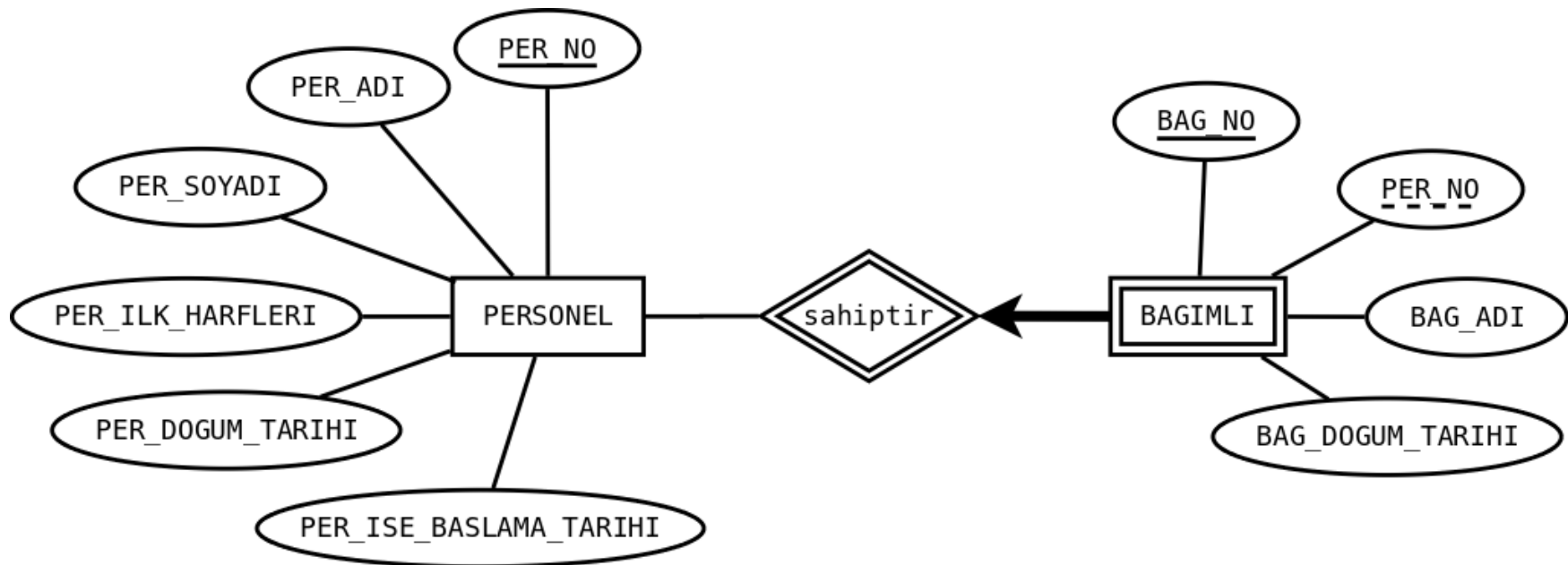
1 öğretim üyesi 1 bölüm yönetebilir

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 İlişkiler

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

COURSE(CRS_CODE, DEPT_CODE, CRS_DESCRIPTION, CRS_CREDIT)

CLASS(CLASS_CODE, CRS_CODE, CLASS_SECTION, CLASS_TIME, ROOM_CODE, PROF_NUM)

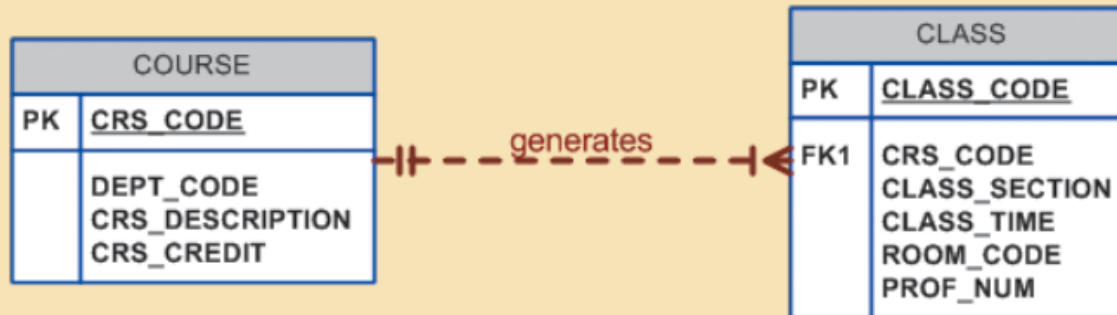


Table name: COURSE

Database name: Ch04_

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Intro. to Microcomputing	3
CIS-420	CIS	Database Design and Implementation	4
MATH-243	MATH	Mathematics for Managers	3
QM-261	CIS	Intro. to Statistics	3
QM-362	CIS	Statistical Applications	4

Table name: CLASS

CLASS_CODE	CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
10012	ACCT-211	1	M/W/F 8:00-8:50 a.m.	BUS311	105
10013	ACCT-211	2	M/W/F 9:00-9:50 a.m.	BUS200	105
10014	ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
10015	ACCT-212	1	M/W/F 10:00-10:50 a.m.	BUS311	301
10016	ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
10017	CIS-220	1	M/W/F 9:00-9:50 a.m.	KLR209	228

Zayıf/Güçlü İlişkiler

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

COURSE(CRS_CODE, DEPT_CODE, CRS_DESCRIPTION, CRS_CREDIT)

CLASS(CRS_CODE, CLASS_SECTION, CRS_CODE, CLASS_TIME, ROOM_CODE, PROF_NUM)

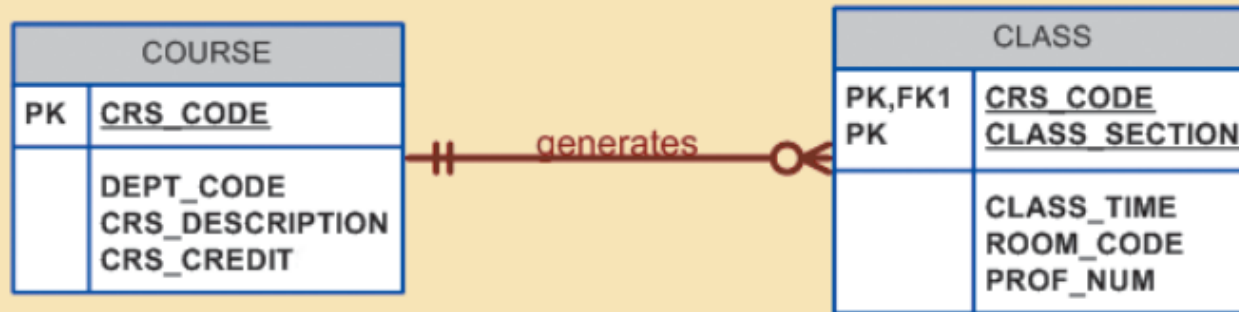


Table name: COURSE

Database name: Ch04_TinyCollege_Alt

CRS_CODE	DEPT_CODE	CRS_DESCRIPTION	CRS_CREDIT
ACCT-211	ACCT	Accounting I	3
ACCT-212	ACCT	Accounting II	3
CIS-220	CIS	Intro. to Microcomputing	3
CIS-420	CIS	Database Design and Implementation	4
MATH-243	MATH	Mathematics for Managers	3
QM-261	CIS	Intro. to Statistics	3
QM-362	CIS	Statistical Applications	4

Table name: CLASS

CRS_CODE	CLASS_SECTION	CLASS_TIME	ROOM_CODE	PROF_NUM
ACCT-211	1	MWF 8:00-8:50 a.m.	BUS311	105
ACCT-211	2	MWF 9:00-9:50 a.m.	BUS200	105
ACCT-211	3	TTh 2:30-3:45 p.m.	BUS252	342
ACCT-212	1	MWF 10:00-10:50 a.m.	BUS311	301
ACCT-212	2	Th 6:00-8:40 p.m.	BUS252	301
CIS-220	1	MWF 9:00-9:50 a.m.	KLR209	228

Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management

İlişki Katılımı (Relationship Participation)

CROW'S FOOT SYMBOL	CARDINALITY	COMMENT
	(0,N)	Zero or many. Many side is optional.
	(1,N)	One or many. Many side is mandatory.
	(1,1)	One and only one. 1 side is mandatory.
	(0,1)	Zero or one. 1 side is optional.

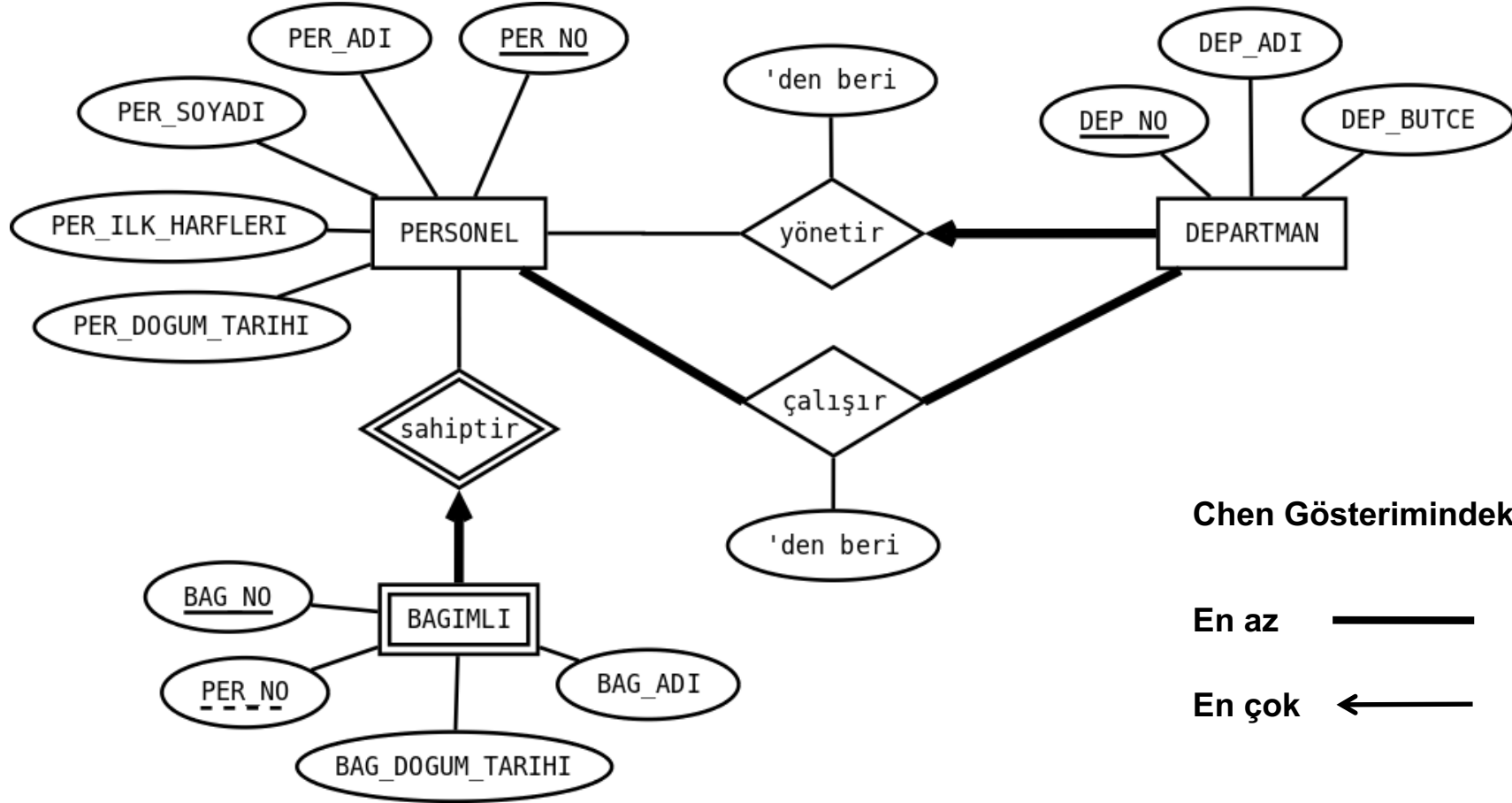
CLASS is optional to COURSE



COURSE and CLASS in a mandatory relationship



Chen Gösteriminde Kısıtlar



Chen Gösterimindeki Kısıtlar

En az —

En çok ←

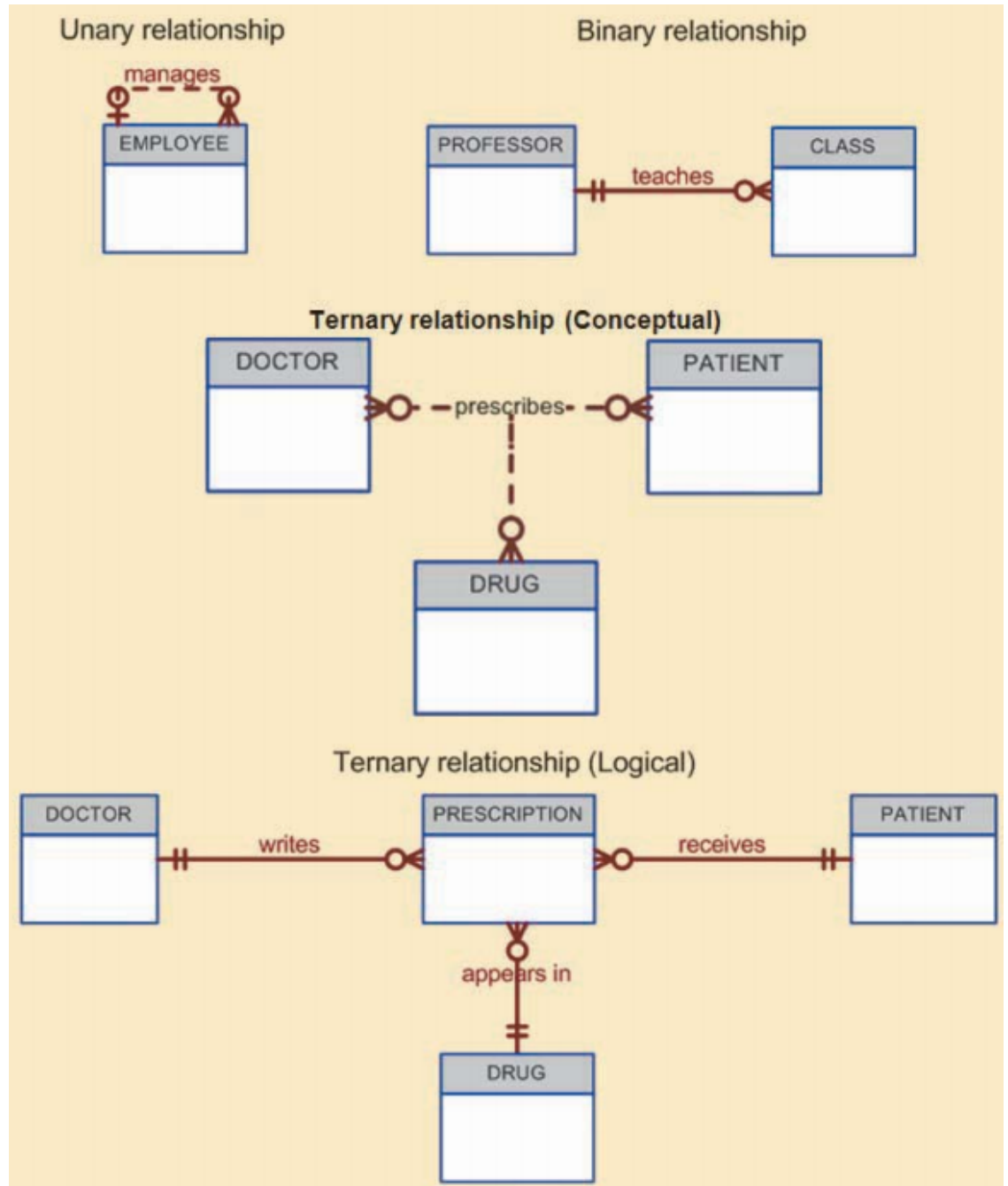
1 bölüm **en az, en çok 1** personel tarafından yönetilir.
1 personel **çok** sayıda bölümü yönetebilir.

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.

İlişki Dereceleri

- ✓ **Tekli (Unary) İlişki:** Bir varlık kendisi ile bağıntılı (ilişkili) ise bu tür bir ilişkiye tekli ilişki 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) İlişki:** İki varlığın bağıntısına (ilişkisine) ikili ilişki denir.
- ✓ **Üçlü (Ternary) İlişki:** Aynı anda 3 varlık birbirine bağlanıyorsa, bu tür ilişkiye üçlü ilişki 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.



VİM (Varlık İlişki 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 ilişki diyagramının geliştirilmesi genellikle aşağıdaki adımları içermektedir.

- ✓Organizasyonun (kurumun), işlerinin (operasyonlarının) tanımını içeren detaylı bir senaryo (hikaye) oluşturulur. (Hikaye 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 ilişkiler oluşturulur.
- ✓İlk varlık ilişki diyagramı geliştirilir.
- ✓Varlıkları net bir şekilde tanımlayan özellikleri ve birincil anahtarları oluşturulur.
- ✓Varlık ilişki 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.
- ✓ Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, Mc Graw Hill