

# Veritabanı Yönetim Sistemleri

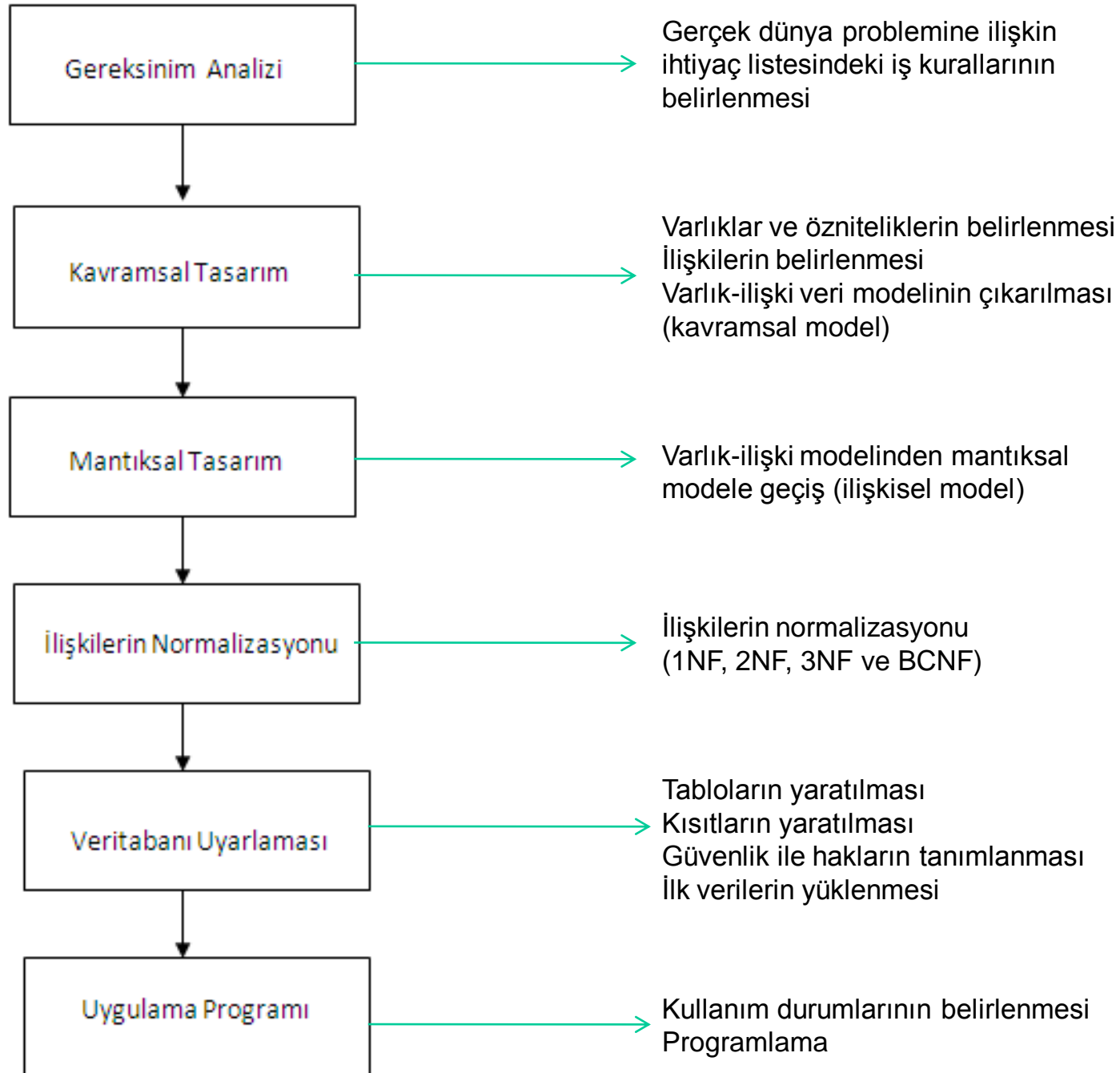
(Veritabanı Tasarımı)  
Varlık İlişki Modeli



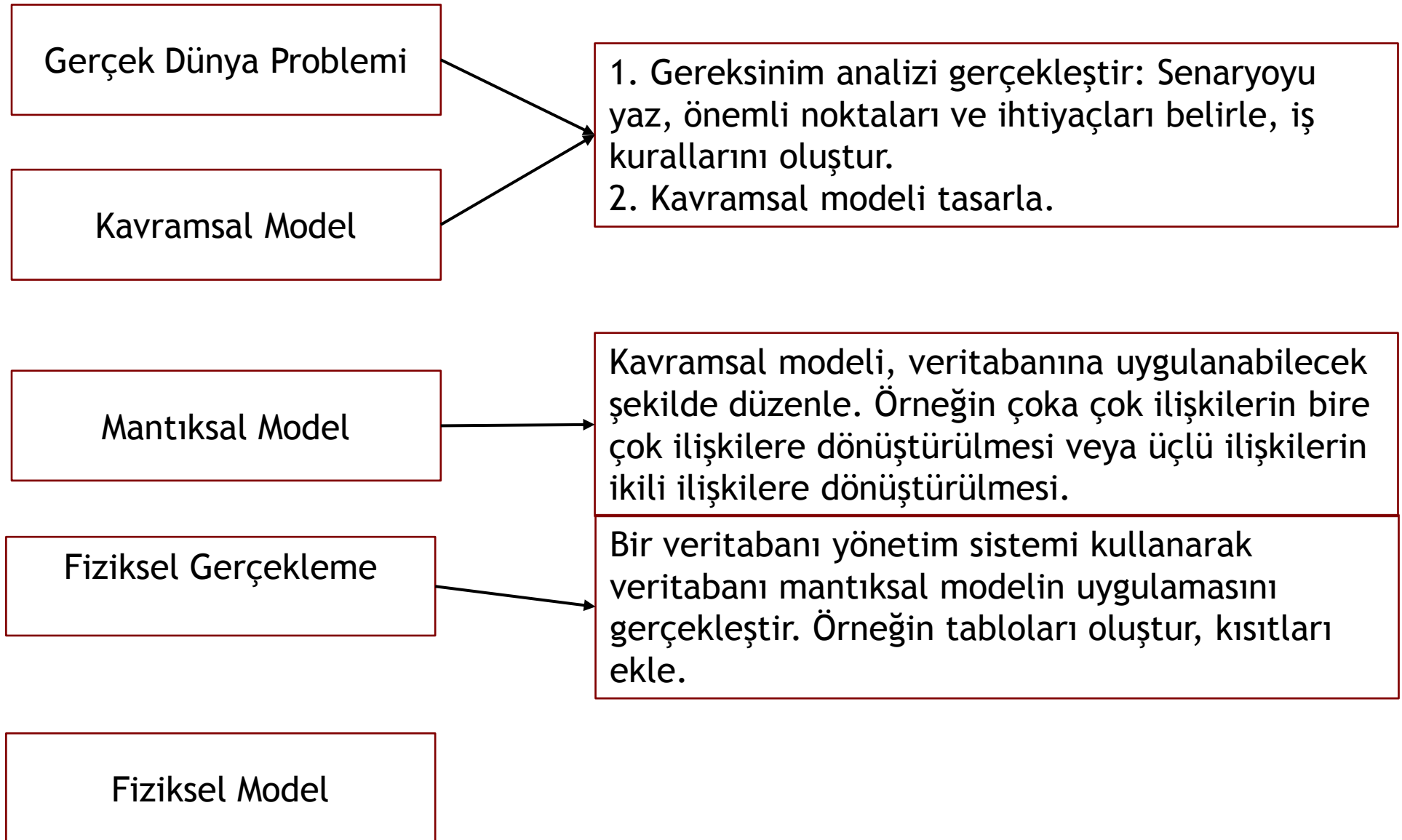
# Konular

- ✓ Veritabanı Tasarım Aşamaları
- ✓ Kavramsal Tasarım
- ✓ Temel Kavramlar
- ✓ Varlıklar Arası İlişkiler
- ✓ Varolma Bağımlılığı (Existence Dependency)
- ✓ Zayıf/Güçlü İlişkiler
- ✓ İlişki Dereceleri
  - ✓ Tekli (Unary) İlişki
  - ✓ İkili (Binary) İlişki
  - ✓ Üçlü (Ternary) İlişki
- ✓ VİM (Varlık İlişki Modeli - ERM) ile Tasarım
- ✓ Örnek Uygulama - Üniversite Bilgi Sistemi
- ✓ VİM ile Uygulama Geliştirme
- ✓ Kaynaklar

# Veritabanı Tasarım Aşamaları

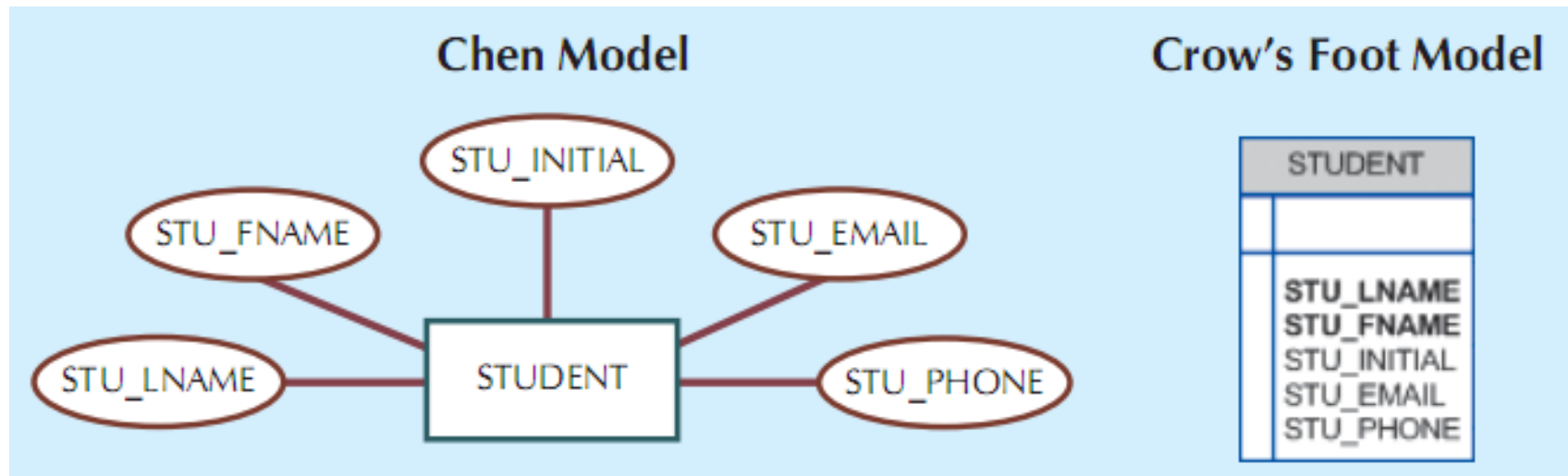


# Kavramsal Tasarım



# Temel Kavramlar

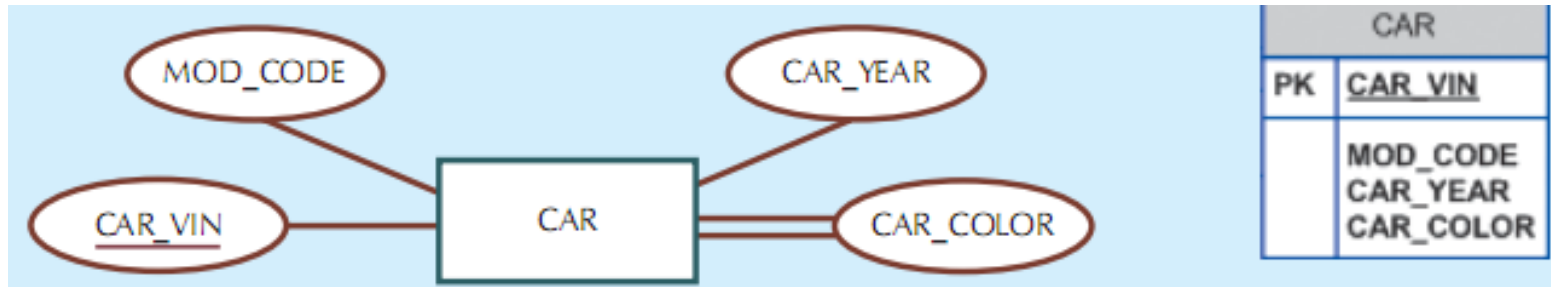
- ✓ Varlık İlişki Modeli (Entity Relationship Diagram (ERD)) veritabanının kavramsal olarak modellenmesini sağlayan yazılım ve donanımlardan tamamen bağımsız gösterim şekilleridir.
- ✓ Farklı kullanıcılar arasında haberleşmeyi sağlamak için kullanılır.
- ✓ ER Modeline bakarak veritabanının tasarımını gerçekleştirmek daha kolaydır.
- ✓ ERD, veritabanının temel bileşenleri olan varlık (nesne yada 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.



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

# Temel Kavramlar

- ✓ Birincil anahtarlar (Primary Key) tek bir alan olabileceği gibi birden fazla alanın birleşiminden de oluşabilir. (dersKodu+bolumKodu+donem)
- ✓ 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) oluşturmak.



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

# Temel Kavramlar

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

$\text{INT}((\text{DATE}() - \text{EMP\_DOB})/365)$

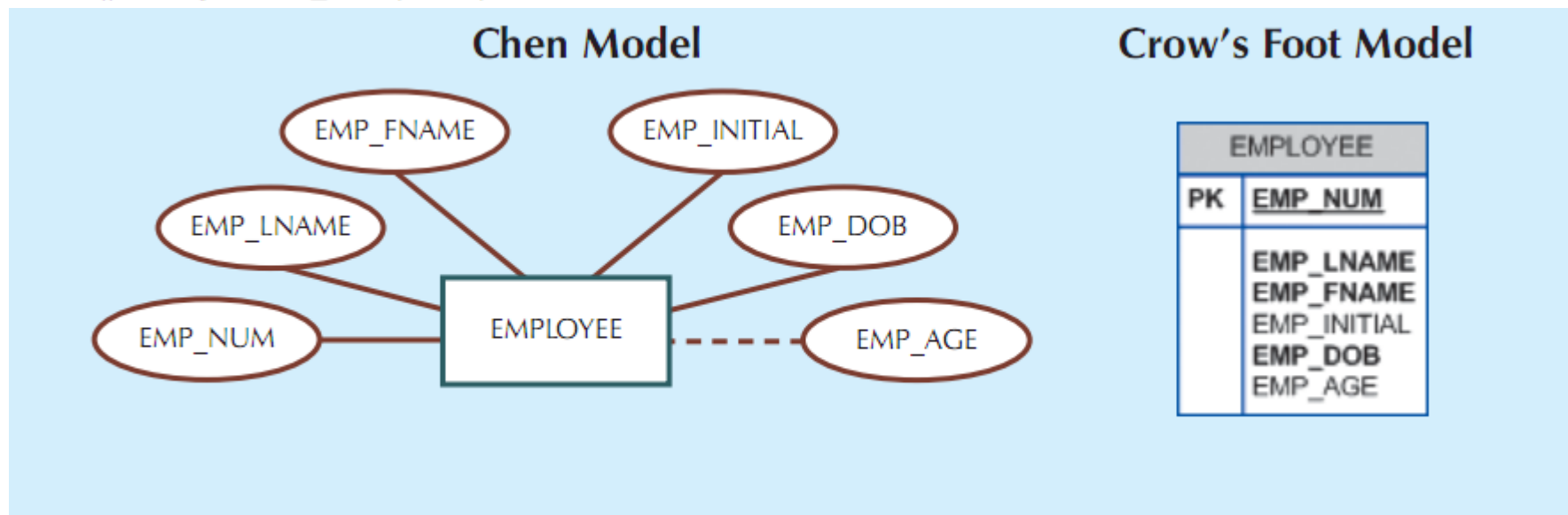


TABLE 4.2

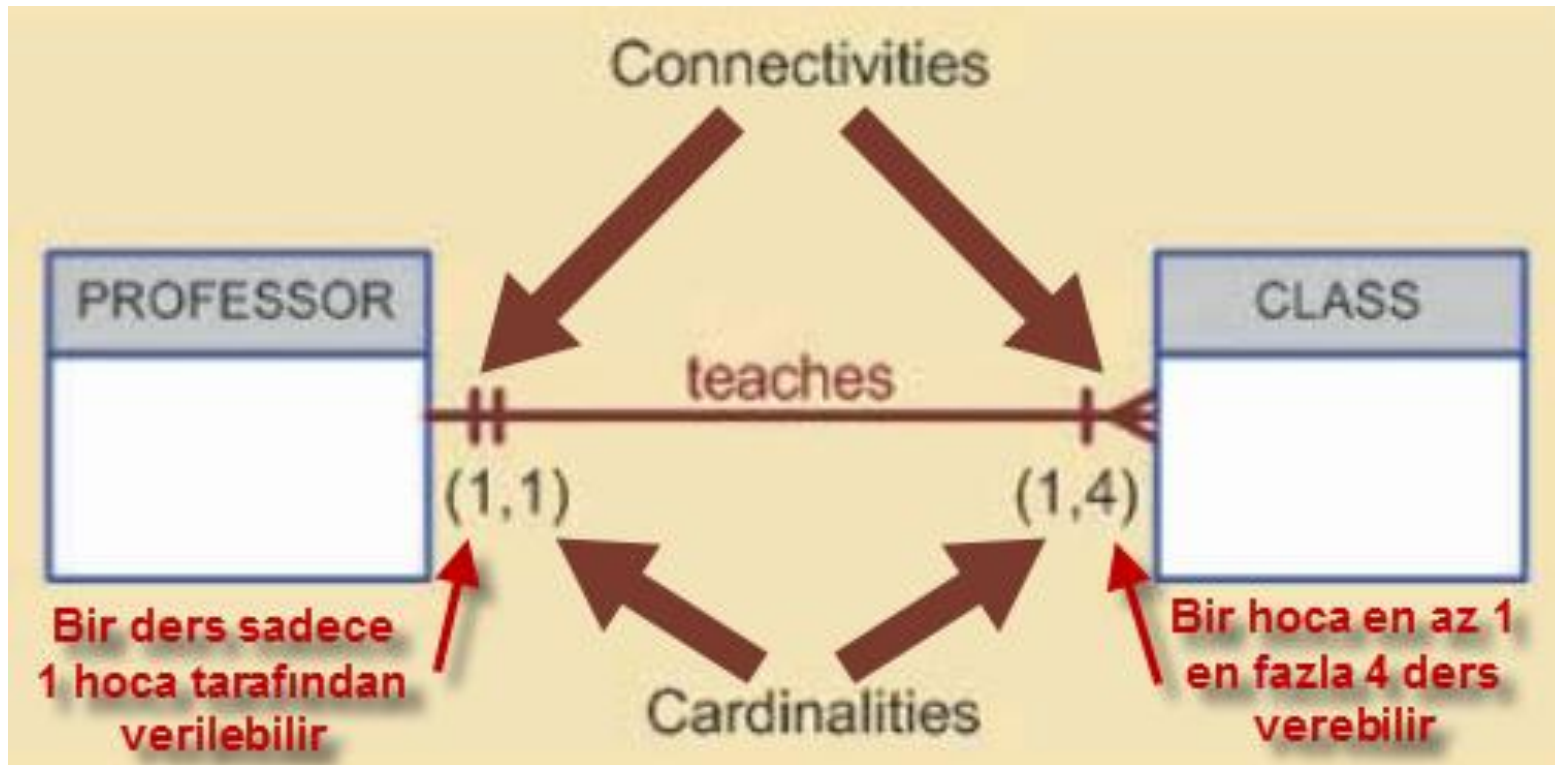
Advantages and Disadvantages of Storing Derived Attributes

	DERIVED ATTRIBUTE	
	STORED	NOT STORED
Advantage	Saves CPU processing cycles Saves data access time Data value is readily available Can be used to keep track of historical data	Saves storage space Computation always yields current value
Disadvantage	Requires constant maintenance to ensure derived value is current, especially if any values used in the calculation change	Uses CPU processing cycles Increases data access time Adds coding complexity to queries

## Varlıklar Arası İlişkiler (1-Çok İlişkisi)

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

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



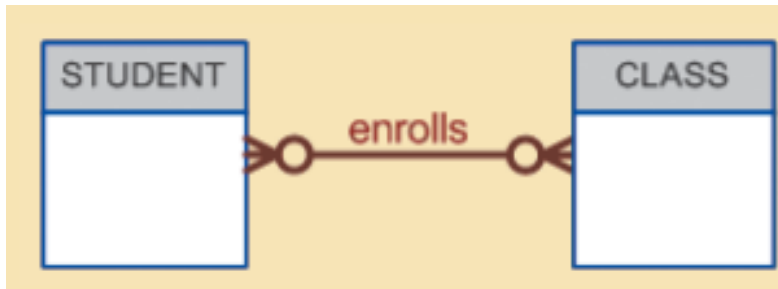
Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management

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



# Çok-çok ilişkisi

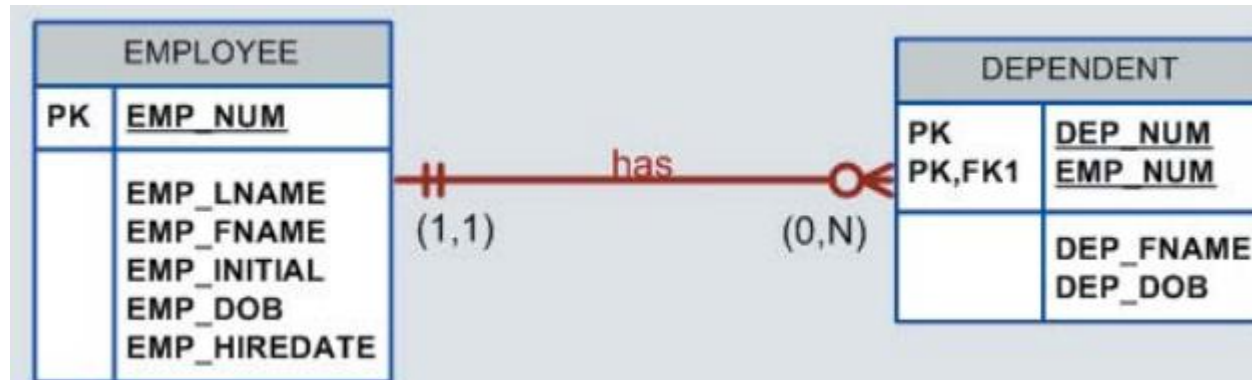
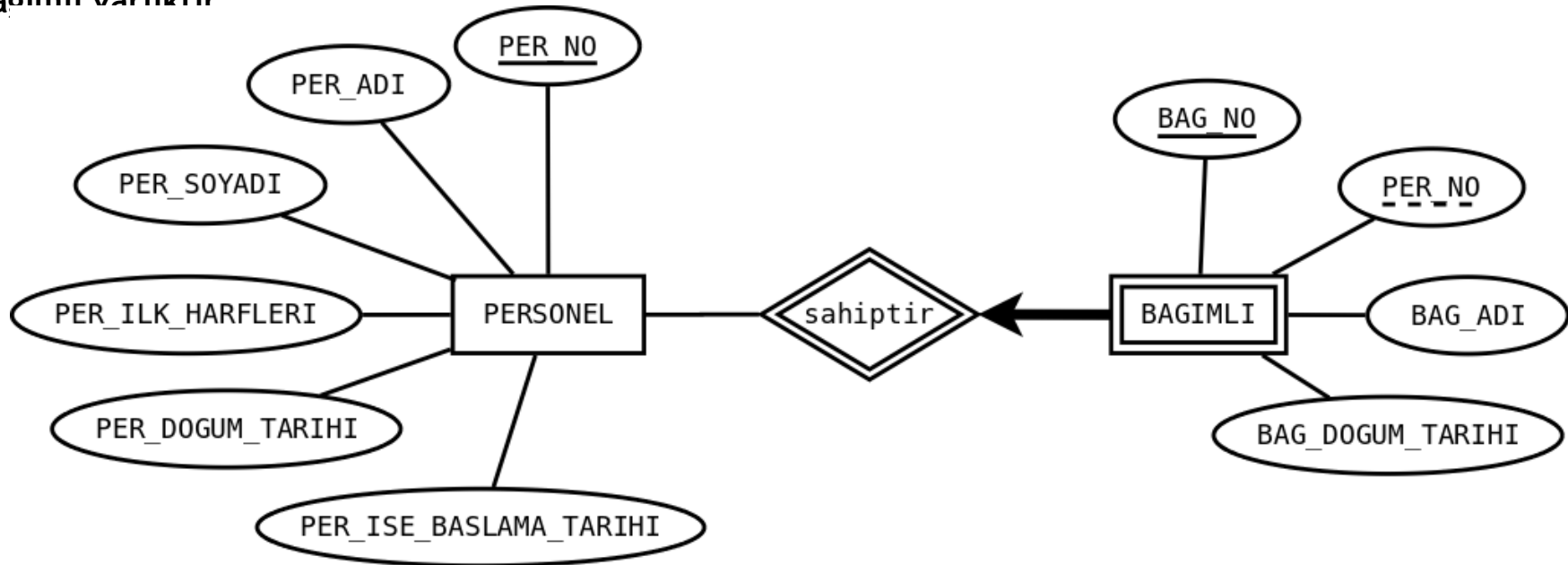
1 öğrenci **çok** sayıda derse kayıt yaptırabilir  
1 ders **çok** sayıda öğrenci tarafından alınabilir.



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

## Varolma 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 varolma 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 var denir.

COURSE(CRS\_CODE, DEPT\_CODE, CRS\_DESCRIPTION, CRS\_CREDIT)

CLASS(CLASS\_CODE, CRS\_CODE, CLASS\_SECTION, CLASS\_TIME, ROOM\_CODE, PROF\_NUM)

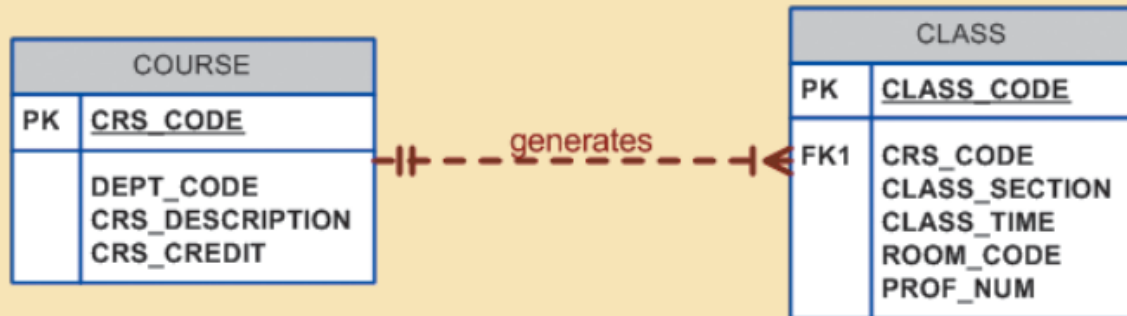


Table name: COURSE

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 varlık arasında **güçlü** bir ilişki var denir.

COURSE(CRS\_CODE, DEPT\_CODE, CRS\_DESCRIPTION, CRS\_CREDIT)

CLASS(CLASS\_CODE, CLASS\_SECTION, CRS\_CODE, CLASS\_TIME, ROOM\_CODE, PROF\_NUM)

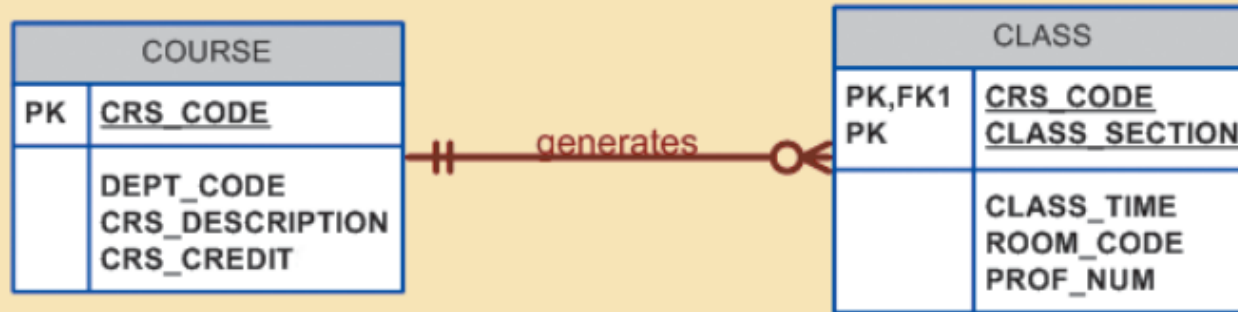


Table name: COURSE

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

# 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.

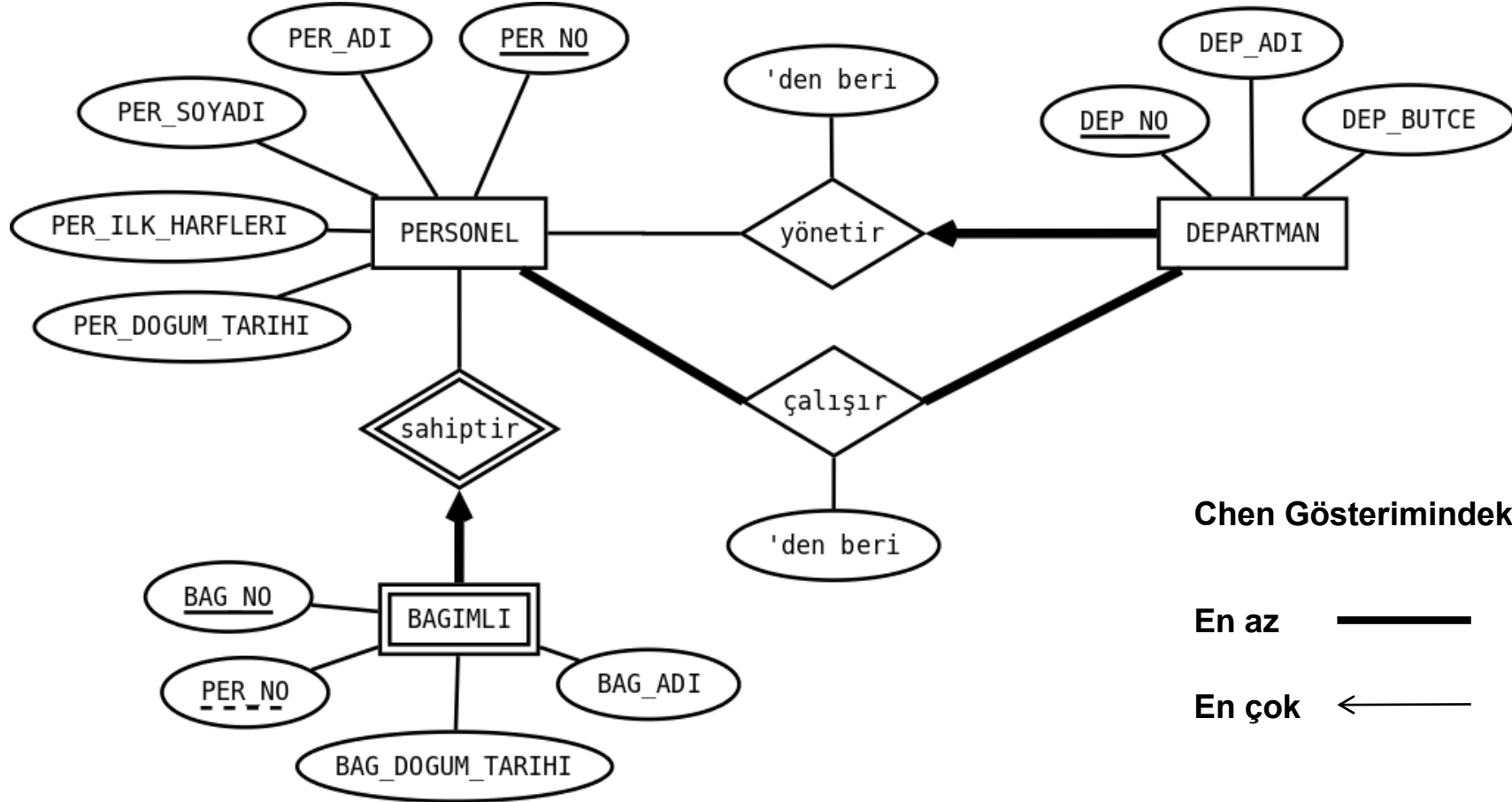
**CLASS is optional to COURSE**



**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.

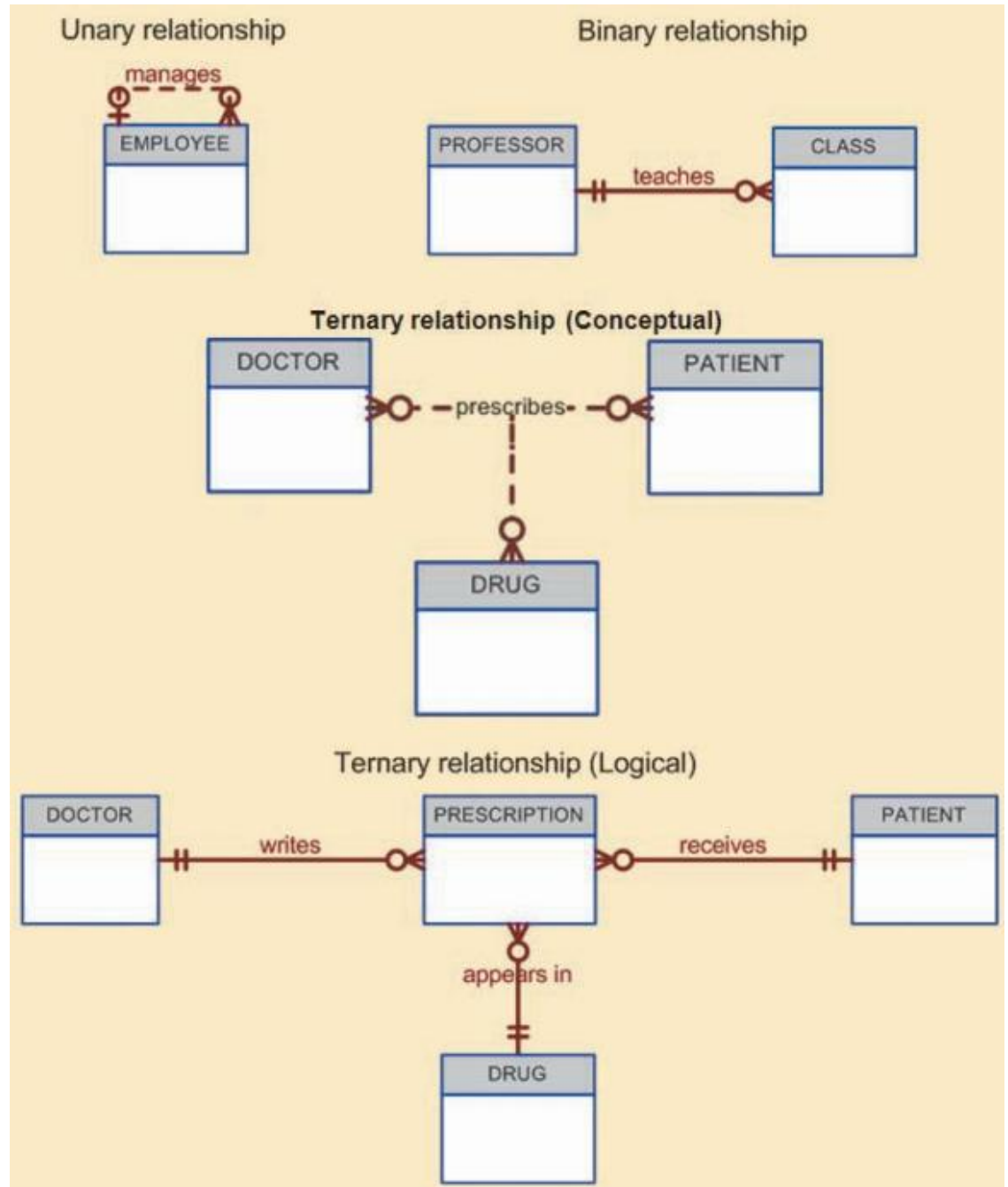
1 bölümde **çok** sayıda personel çalışabilir. **En az 1** personel çalışmak zorundadır.  
1 personel **çok** sayıda bölümde çalışabilir. **En az 1** bölümde çalışmak zorundadır.

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, bir veya daha fazla çalışanın aynı zamanda yöneticisidir. Bir çalışanın, muhakkak bir yöneticisi olmalıdır.
- ✓ **İkili (Binary) İlişki:** İki varlığın bağıntısına (ilişisine) 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 mevcuttur.
  - ✓ Bir doktor bir veya daha fazla reçete yazar.
  - ✓ Bir hasta bir veya daha fazla reçete alabilir.
  - ✓ Bir ilaç, bir veya daha fazla 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 yada 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.



# Örnek Uygulama - Üniversite Bilgi Sistemi

## SENARYO

✓Üniversite bilgi sisteminde, bir üniversitenin fakültelerinin, bölümlerinin, bölümlerin sunduğu derslerin, öğretim elemanlarının, açılan derslerin, açılan derslerin dersliklerinin, öğrencilerin, öğrencilerin açılan derslere yaptıkları kayıtların, üniversitedeki binaların ve binalardaki dersliklerin bilgilerinin saklanması istenmektedir.

## İŞ KURALLARI

- ✓Üniversite fakültelerden oluşur.
- ✓Fakülte bilgileri içerisinde fakültenin adı ve dekanı bilgileri yer almalıdır.
- ✓Bölüm bilgileri içerisinde bölüm adı, bağlı oldu fakülte ve bölüm başkanı bilgileri yer almalıdır.
- ✓Öğretim elemanı bilgileri içerisinde kimlik numarası, adı, soyadı, isminin ilk harfleri ve bağlı bulunduğu (görev yaptığı) bölüm bilgileri bulunmalıdır.
- ✓Bölümlerin sunduğu dersler (katalog dersler) bilgileri içerisinde ders kodu, ders adı, kredisi ve dersi sunan bölümün bilgisi olmalıdır.
- ✓Açılan dersler bilgileri içerisinde açılan dersin kodu, katalog ders bilgisi (kodu), bolumu, şube bilgisi, dersi veren öğretim elemanın ve dersin yapıldığı dersliğin bilgisi olmalıdır.
- ✓Öğrenci bilgileri içerisinde öğrenci numarası, bölüm numarası, adı, soyadı ve danışman hocasının bilgisi yer almalıdır.
- ✓Öğrencilerin açılan derslere yaptıkları kayıtlar, kayıt tarihleri ve o derslerden aldıkları notlar da saklanmalıdır.
- ✓Bina bilgileri içerisinde bina adı ve yeri bilgileri bulunmalıdır.
- ✓Derslik bilgileri içerisinde derslik kodu ve dersliğin hangi binada bulunduğuna ilişkin bilgiler bulunmalıdır.

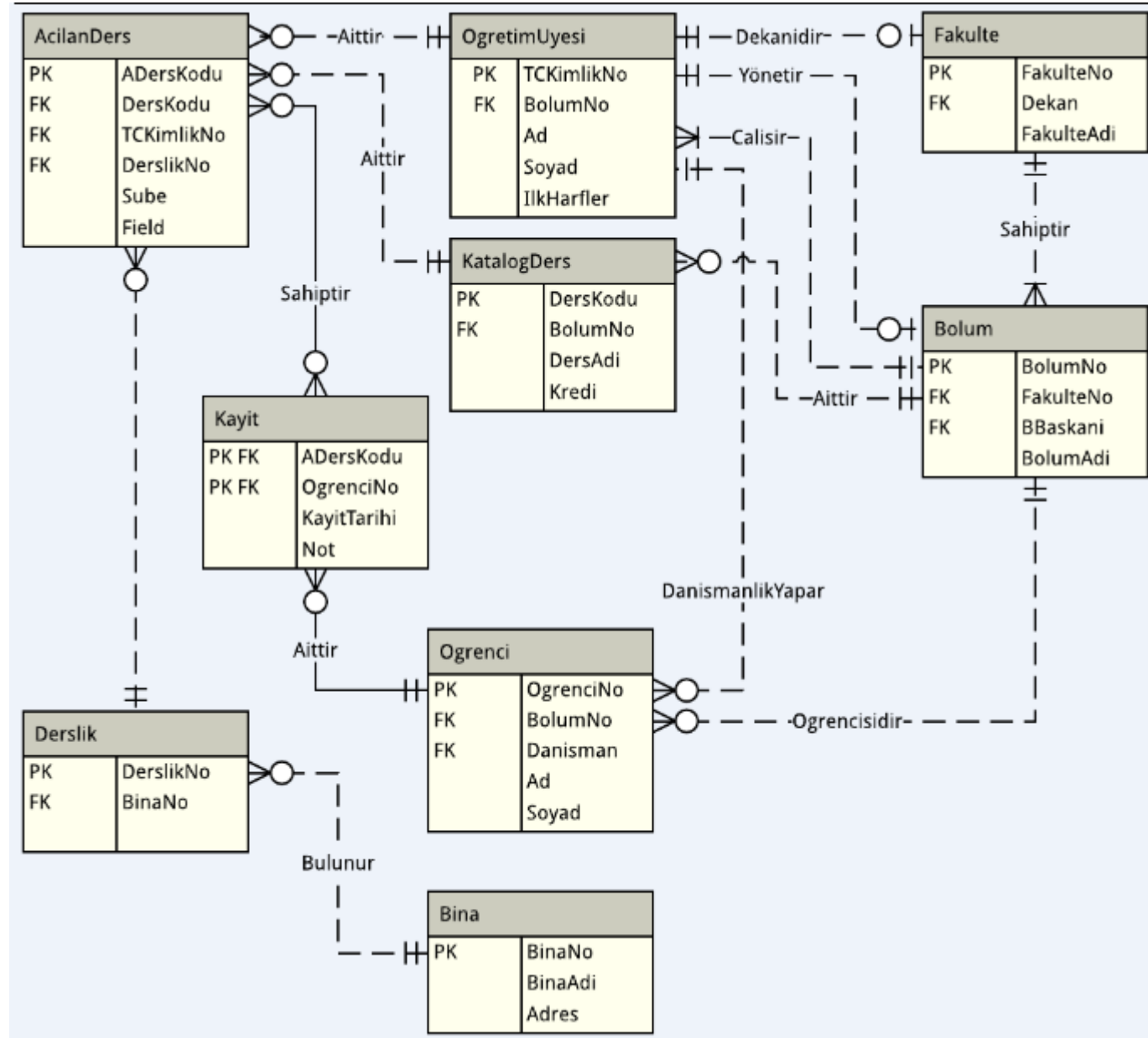
# Örnek Uygulama - Üniversite Bilgi Sistemi

## İŞ KURALLARI

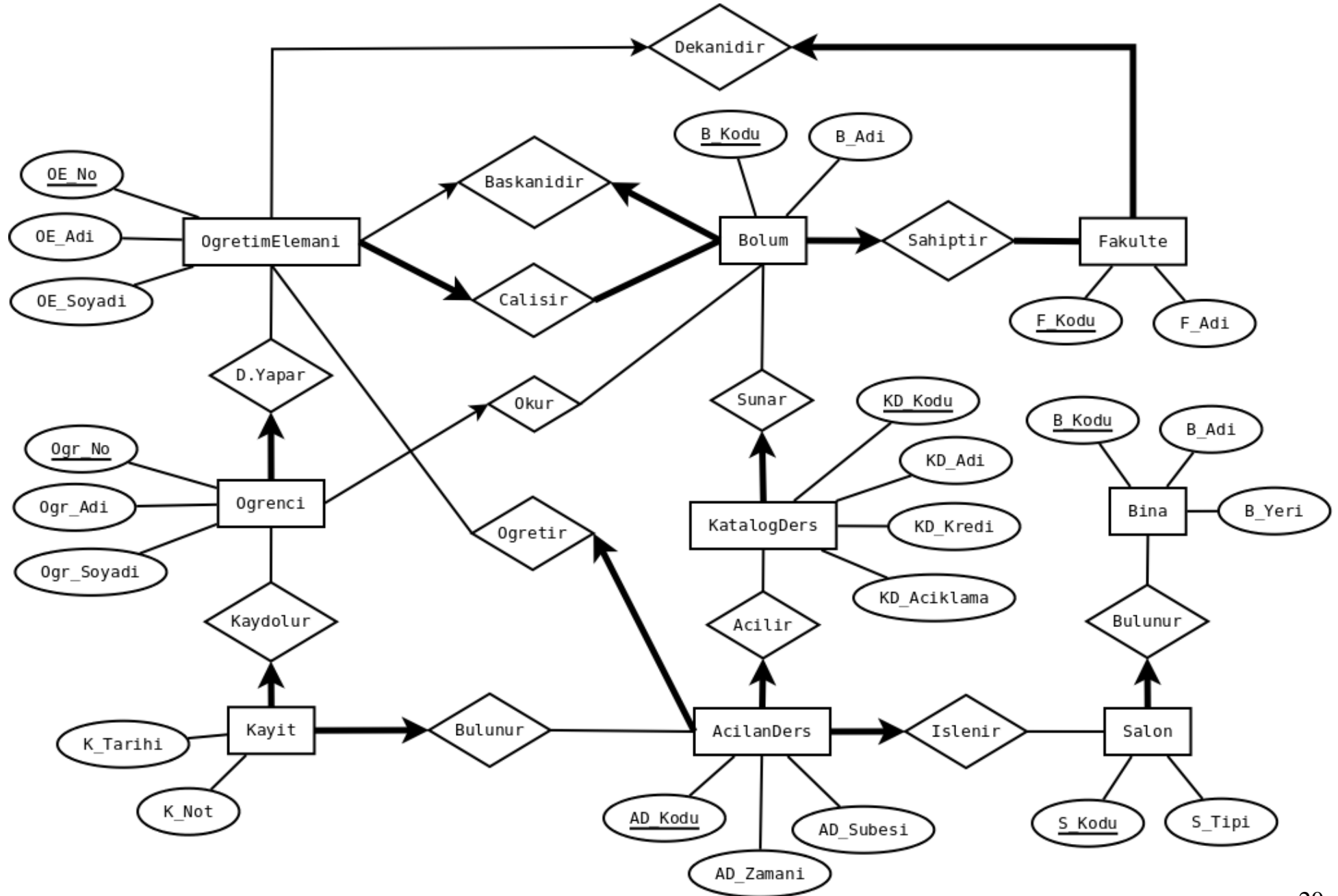
- ✓ Bir fakültenin yalnızca bir tane dekanı (öğretim elemanları içerisinde) olmak zorundadır. Bir öğretim elemanı en fazla bir fakültenin dekanı olabilir.
- ✓ Bir fakültenin en az bir bölümü olmak zorundadır ancak birden fazla bölümü de olabilir. Bir bölüm ise yalnızca bir fakültenin altında yer alabilir.
- ✓ Bir öğretim elemanı yalnızca bir bölüme bağlıdır. Bir bölümün en az bir öğretim elemanı olmak zorundadır ancak birden fazla öğretim elemanı da olabilir.
- ✓ Bir bölüm yalnızca bir öğretim elemanı tarafından yönetilmek zorundadır. Bir öğretim elemanı ise en fazla bir bölümün yöneticiliğini yapabilir.
- ✓ Bir öğretim elemanı çok sayıda öğrencinin danışman hocası olabilir. Bir öğrencinin danışmanlığını yalnızca bir tane öğretim elemanı yapabilir.
- ✓ Bir bölümün çok sayıda öğrencisi mevcuttur. Bir öğrenci, en fazla bir tane bölümün öğrencisidir.
- ✓ Bir öğretim elemanı birden fazla (açılan) ders verebilir. Bir (açılan) ders yalnızca bir öğretim elemanı tarafından verilmelidir.
- ✓ Bir bölüm çok sayıda katalog ders sunabilir. Bir katalog ders yalnızca bir bölüm tarafından sunulur.
- ✓ Bir katalog dersin çok sayıda açılan dersi (şubesi) olabilir. Bir açılan dersin yalnızca bir tane katalog dersi olmalıdır.
- ✓ Bir öğrenci birden fazla derse kayıt yaptırabilir. Bir derse birden fazla öğrenci kayıt olabilir.
- ✓ Bir açılan ders yalnızca bir derslikte işlenir. Bir derslikte çok sayıda açılan ders işlenebilir.
- ✓ Bir derslik yalnızca bir binada bulunur. Bir binada çok sayıda derslik bulunabilir.

# Örnek Uygulama

## Üniversite Bilgi Sist.



# Örnek Uygulama - Üniversite Bilgi Sistemi



# Örnek Uygulama - Kitabevi Veritabanı

## SENARYO

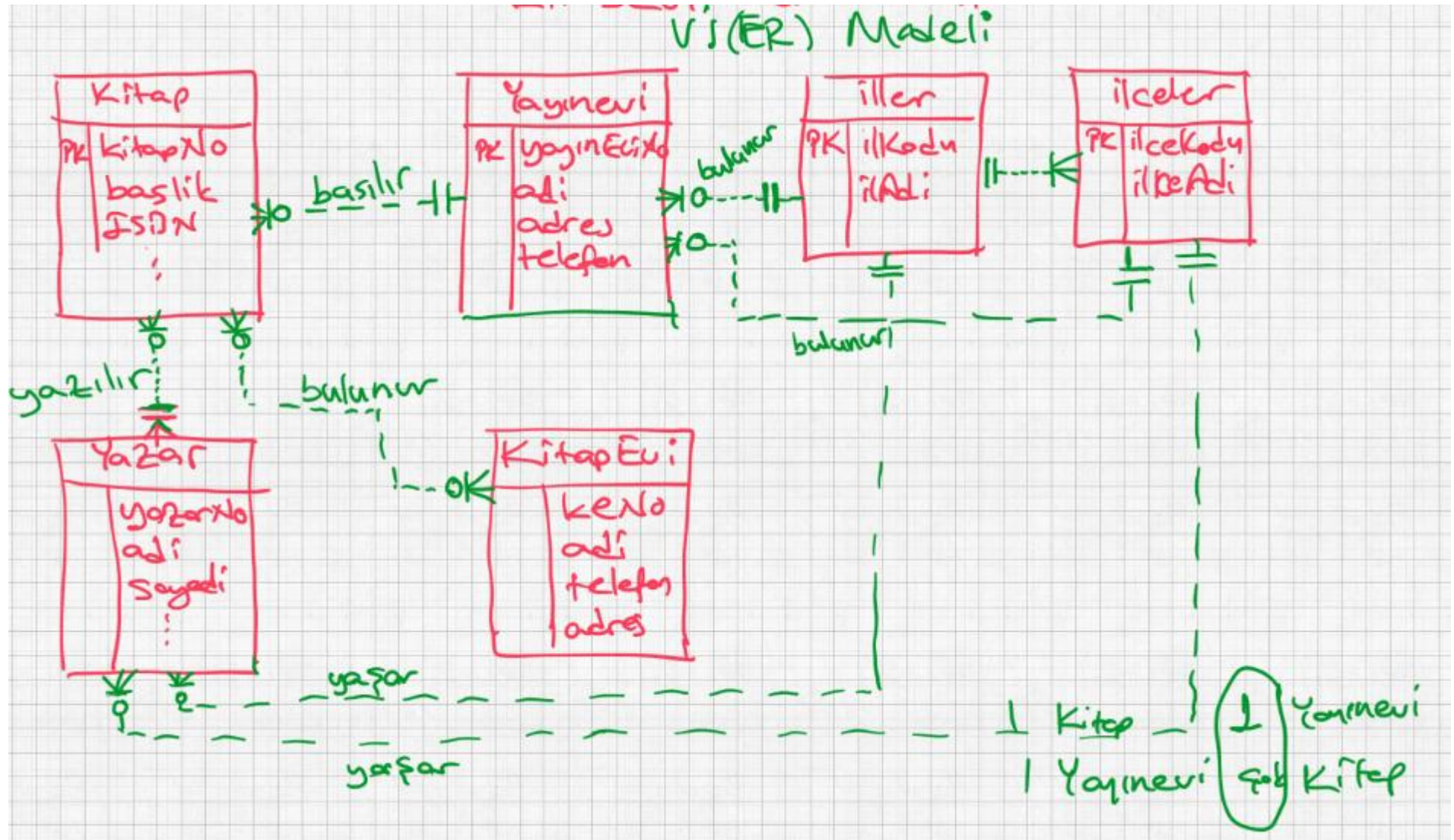
- ✓ Bir kitabevi için veritabanı tasarlanması istenmektedir.

## İŞ KURALLARI

- ✓ Kitabevinde kitaplar mevcuttur.
- ✓ Her kitabın eşsiz bir ISBN numarası mevcuttur. Bunun yanı sıra kitaplar, başlık ve fiyat bilgilerine sahiptir.
- ✓ Kitaplar yayınevleri tarafından basılır.
- ✓ Yayınevinin kodu, adı ve telefon numarası bilgileri mevcuttur.
- ✓ Kitaplar yazarlar tarafından yazılır.
- ✓ Yazarın kodu, adı, soyadı ve telefon numarası bilgileri mevcuttur.
  
- ✓ Bir kitap yalnızca bir yayınevi tarafından basılır. Bir yayınevinin hiçbir kitabı olmayabilir ancak birden fazla kitabı da olabilir.
- ✓ Bir kitabın en az bir yazarı olmalıdır; birden fazla yazarı da olabilir. Bir yazarın hiçbir kitabı olmayabilir ancak çok sayıda kitabı da mevcut olabilir.

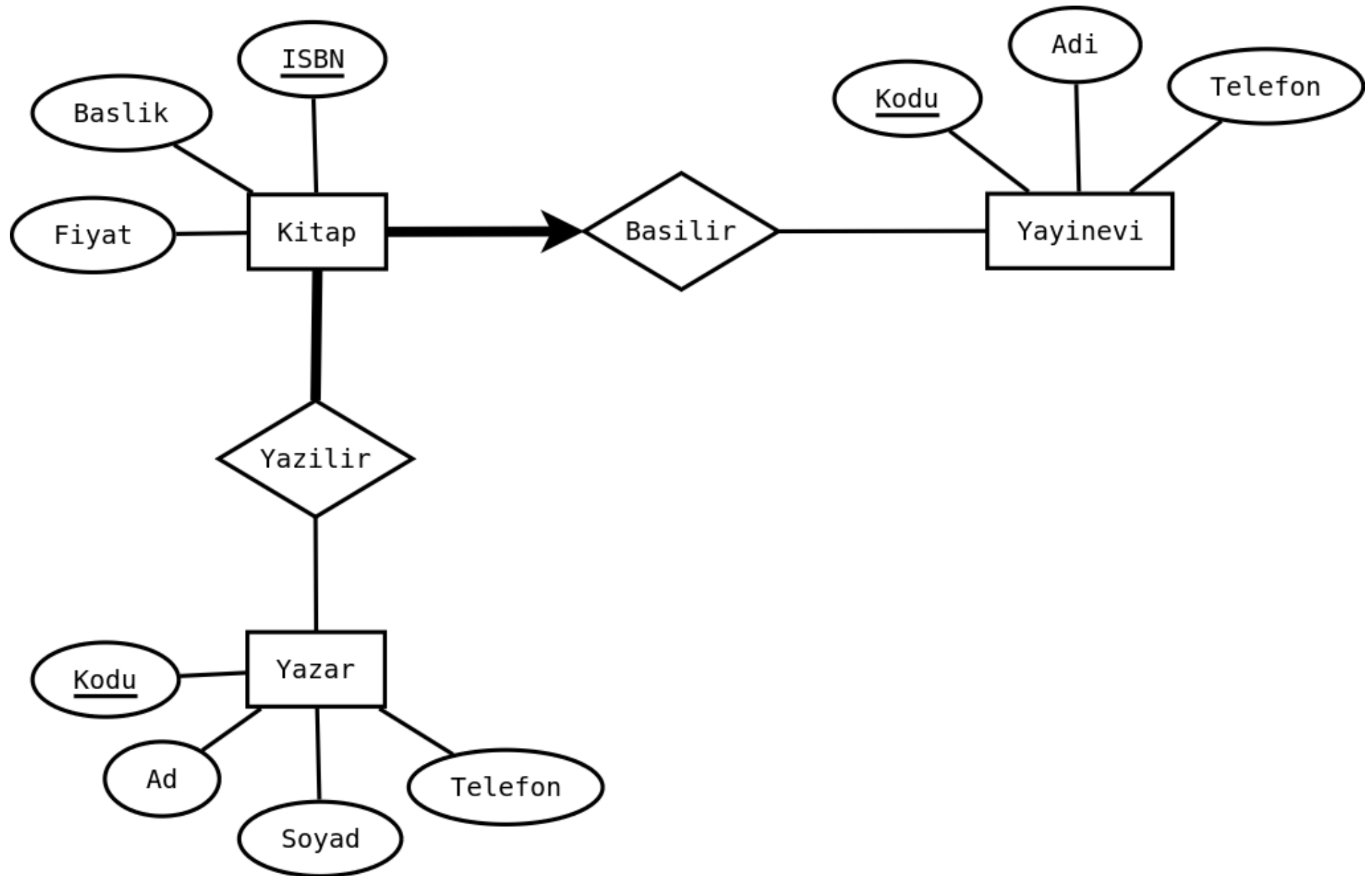
# Örnek Uygulama - Kitabevi Veritabanı

Vİ(ER) Modeli





## Örnek Uygulama - Kitabevi Veritabanı



# Örnek Uygulama - Kütüphane Veritabanı

## SENARYO

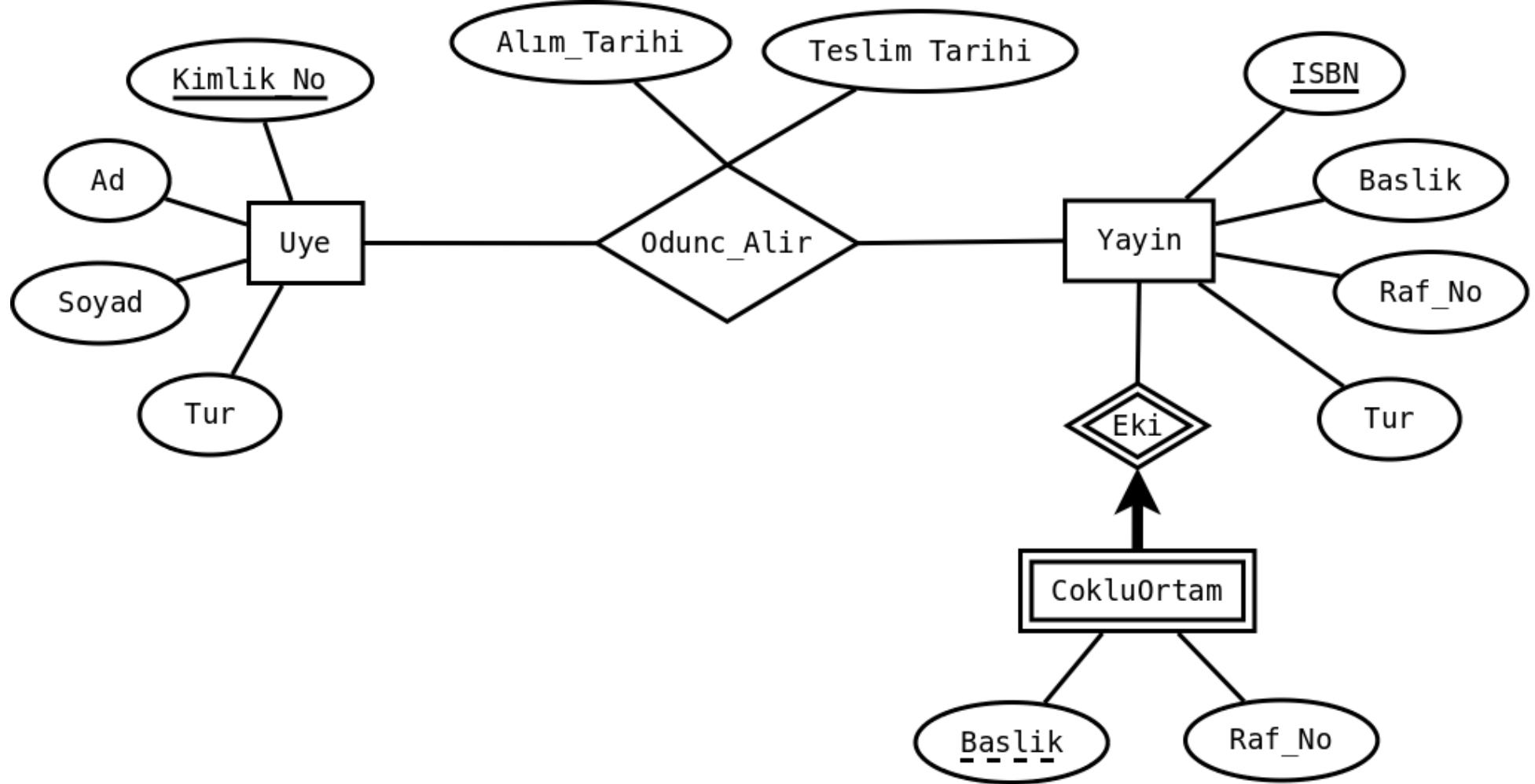
✓ Bir üniversite kütüphanesi için veritabanı tasarlanması istenmektedir. Bu veritabanında, kütüphanedeki bütün basılı yayınlar ve bunların ekleri olan çoklu ortam öğelerinin kayıtlarının bulunması talep edilmektedir. Aynı zamanda kütüphaneye üye olan kişilerin ve üyelerin ödünç aldıkları yayınların ödünç bilgilerinin de kaydedilmesi gerekmektedir.

## İŞ KURALLARI

- ✓ Her üyenin kimlik numarası, adı, soyadı ve türü bilgileri mevcuttur. Üyenin türü, akademisyen, personel veya öğrenci olabilir.
- ✓ Her yayının ISBN numarası, başlığı, raf numarası ve türü bilgileri mevcuttur. Yayının türü süreli yayın yahut kitap olabilir.
- ✓ Bazı yayınların çoklu ortam ekleri olabilir. Çoklu ortam eklerinin başlığı ve raf numarası mevcuttur. Çoklu ortam ekleri birbirinden başlık ve ait oldukları yayının ISBN numarası bilgileri ile ayırt edilebilir.
- ✓ Üyelerin ödünç aldıkları yayınların, ödünç alım ve teslim tarihleri mevcut olmalıdır.
- ✓ Bir çoklu ortam öğesi yalnızca bir yayının eki olmak zorundadır. Yayını olmayan çoklu ortam öğesi mevcut olamaz.
- ✓ Bir yayının hiçbir çoklu ortam eki olmayabilir. Ancak birden fazla çoklu ortam eki de mevcut olabilir.
- ✓ Bir üye hiçbir yayın ödünç almayabileceği gibi birden fazla yayın da ödünç alabilir.
- ✓ Bir yayın hiçbir üye tarafından ödünç alınmayabilir. Ancak birden fazla üye tarafından da ödünç alınabilir.



# Örnek Uygulama - Kütüphane Veritabanı



# Örnek Uygulama - Ayakkabı İmalathanesi Veritabanı

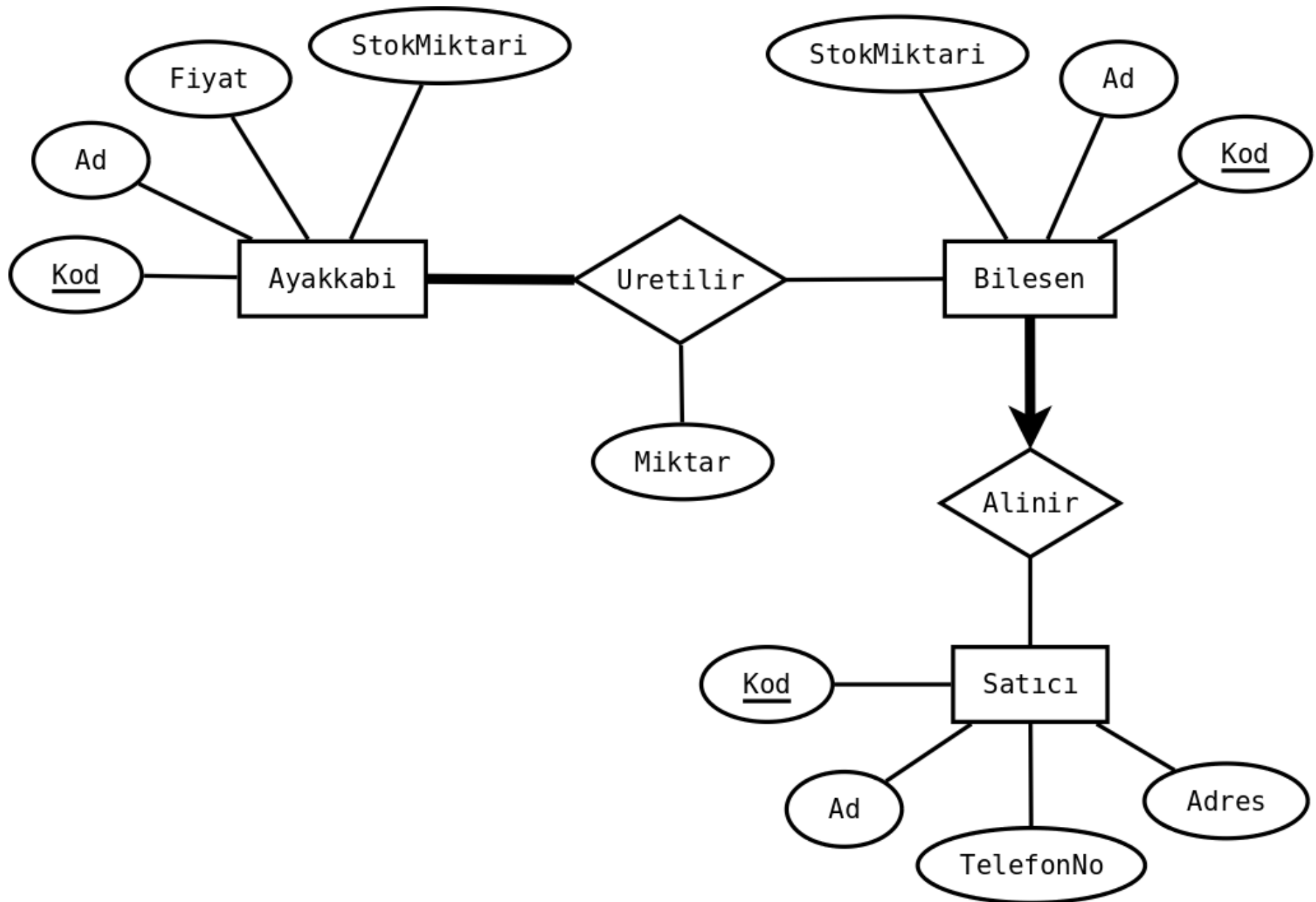
## SENARYO

✓ Bir ayakkabı imalat firması için geliştirilen uygulamada firmanın ürettiği ayakkabılar, bu ayakkabılarda kullanılan bileşenler ve bileşenlerin satıcıları hakkındaki bilgilerin saklandığı bir veritabanına ihtiyaç bulunmaktadır.

## İŞ KURALLARI

- ✓ Firma ayakkabı üretir.
- ✓ Ayakkabıların kodu, adı, stok miktarı ve fiyatı bilgileri mevcuttur.
- ✓ Ayakkabılar bileşenler kullanılarak üretilir.
- ✓ Bileşenlerin kodu, adı ve stok miktarı bilgileri mevcuttur.
- ✓ Bileşenler satıcılardan tedarik edilir.
- ✓ Satıcıların kodu, adı, adresi ve telefon numarası bilgileri mevcuttur.
- ✓ Her bir ayakkabı için, bu ayakkabının üretiminde kullanılan bileşenler, adetleri ile birlikte kaydedilmelidir.
- ✓ Bir ayakkabı, birden fazla bileşenden oluşmak zorundadır. Bir bileşen, birden fazla ayakkabıda kullanılabilir. Ancak hiçbir ayakkabıda da kullanılmayabilir.
- ✓ Bir bileşen yalnızca bir satıcıdan temin edilir. Bir satıcı hiçbir bileşen satmayabilir ancak birden fazla bileşen de satabilir.

## Örnek Uygulama - Ayakkabı İmalathanesi Veritabanı



# Örnek Uygulama - Müzik Şirketi Veritabanı

## SENARYO

✓ Bir müzik şirketi, yayınladığı albümlerin, bu albümlerde yer alan şarkıların, albümleri gerçekleştiren müzisyenlerin ve müzisyenlerin kullandığı müzik aletlerinin kaydedilip saklanacağı bir uygulama talep etmiştir.

## İŞ KURALLARI

✓ Şirkette kayıt yapan her müzisyenin, kimlik numarası, adı, adresi ve telefon numarası bilgileri bulunmaktadır. Müzisyenlerin çoğunun adresi aynıdır ve yalnızca bir telefon bilgisi bulunmaktadır.

✓ Şarkı kayıtlarında kullanılan her müzik aletinin eşsiz bir kodu, adı (Örn. gitar, flüt vb.) ve müziksel anahtarı (Örn. C, B-Flat, E-Flat vb.) mevcuttur.

✓ Kaydedilen her albümün eşsiz bir kodu, başlığı, telif hakkı tarihi ve formatı (cd, dvd vb.) mevcuttur.

✓ Kaydedilen her şarkının eşsiz bir kodu, adı ve yazarı mevcuttur.

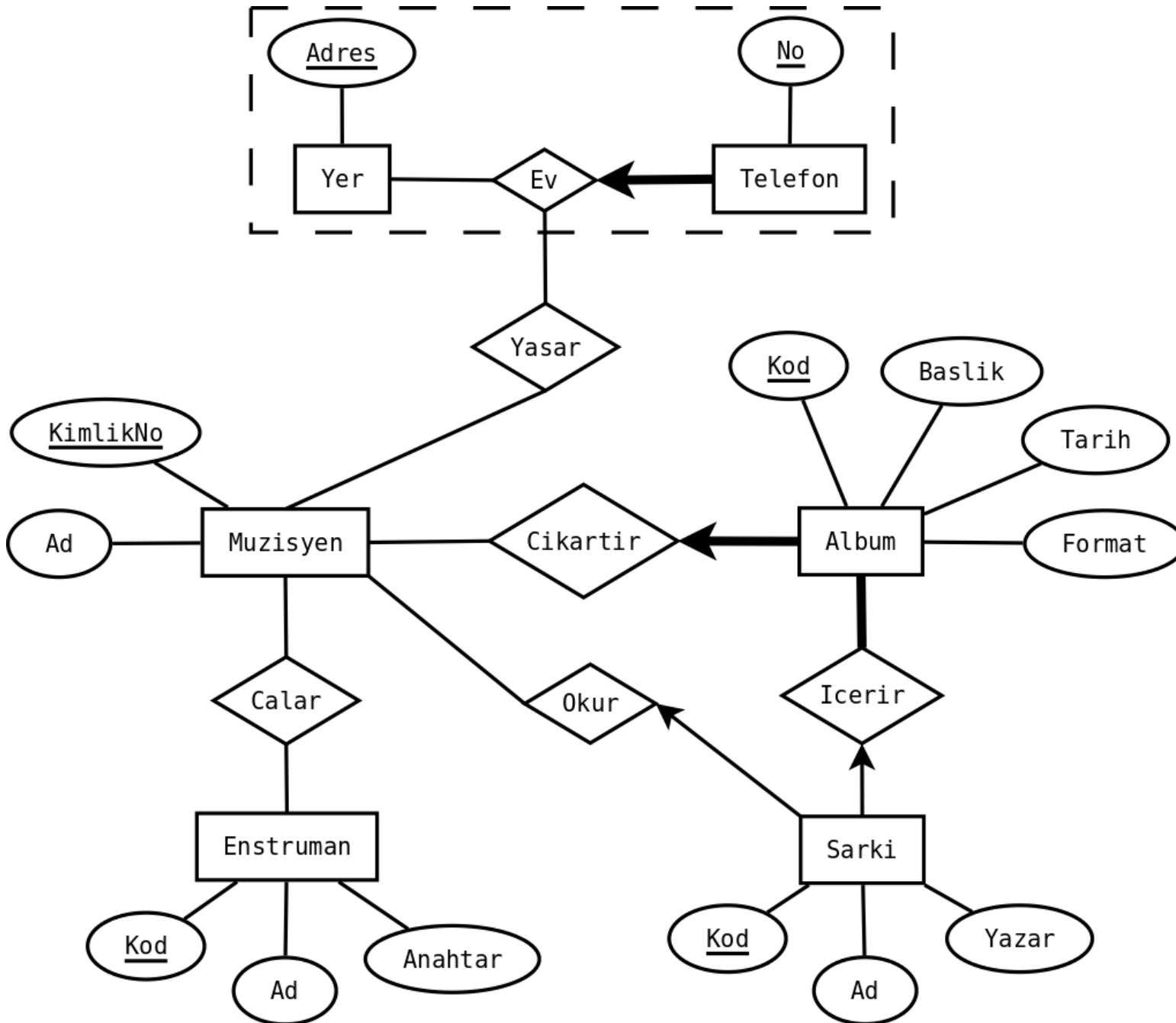
✓ Bir albümde en az bir adet şarkı bulunmak zorundadır; birden fazla şarkı da bulunabilir. Bir şarkı birden fazla albümde yer alamaz. Bir şarkı hiçbir albümde de yer almayabilir.

✓ Bir müzisyen birden çok müzik aleti çalabilir; hiçbir müzik aleti çalmayabilir de. Bir müzik aleti hiçbir müzisyen tarafından çalınmayabilir ancak birden fazla müzisyen tarafından da çalınabilir.

✓ Bir şarkı en fazla bir müzisyen tarafından okunabilir. Bir müzisyen hiçbir şarkı okumayabileceği gibi çok sayıda şarkı da okuyabilir.

✓ Bir albüm yalnızca bir müzisyen tarafından çıkartılabilir ve en az bir müzisyen tarafından çıkartılmalıdır. Bir müzisyenin hiçbir albümü olmayabilir ancak birden çok albümü de olabilir.

# Örnek Uygulama - Müzik Şirketi Veritabanı



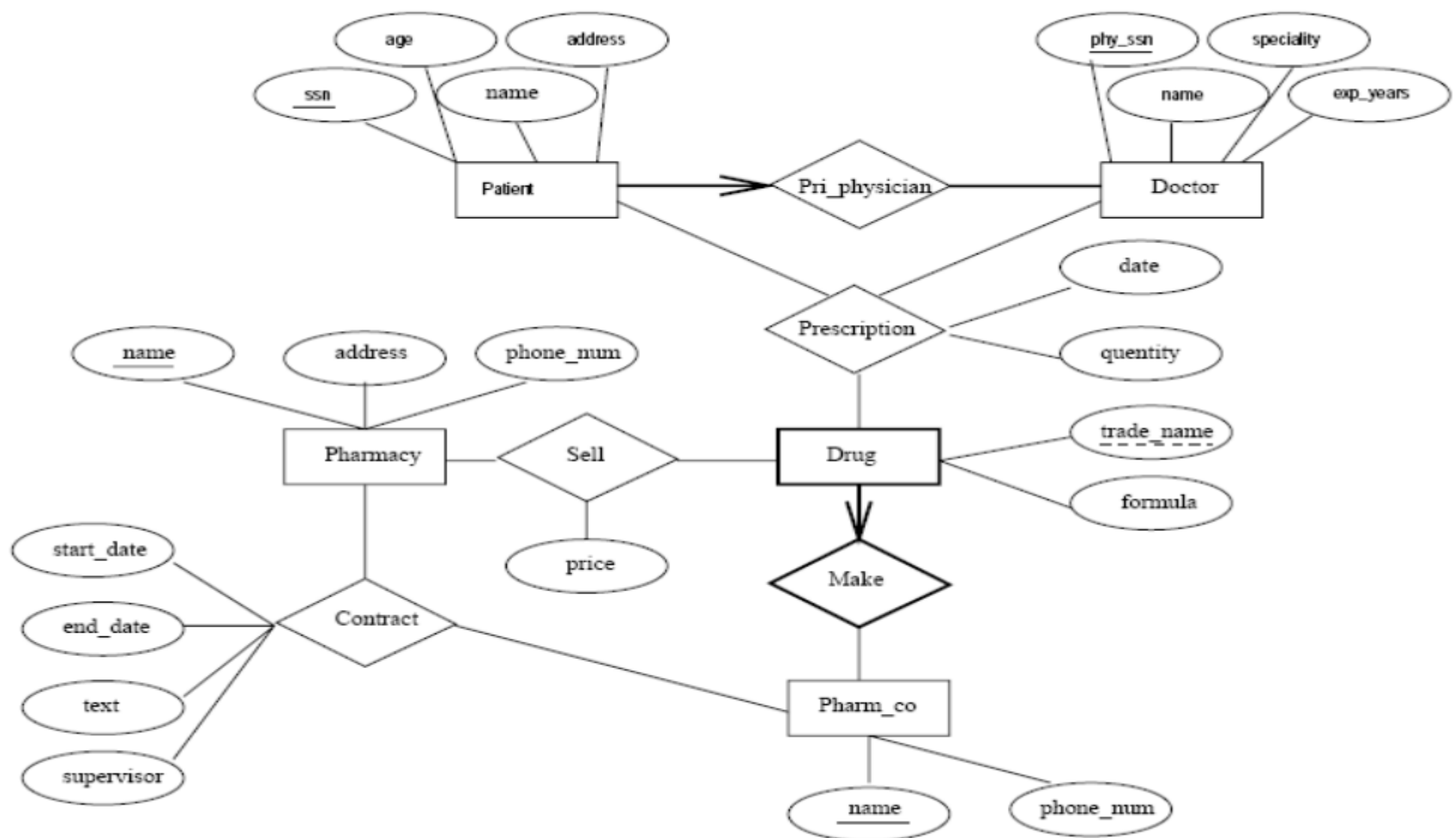
## Örnek Uygulama - Eczane

- Eczane için bir veritabanı modellenecektir. Sistemler istenenler:
- 1) hastaların SSN, adı, adresi ve yaşı tutulmalıdır.
- 2) doktorların SSN, adı, uzmanlığı ve deneyim süresi (yıl olarak) tutulmalıdır.
- 3) her ilaç firmasının kendine özel adı ve telefon numarası vardır.
- 4) her ilacın satış adı ve formülü tutulmalıdır. Her ilaç sadece bir ilaç firması tarafından satılmaktadır ve satış adı her ilaç için şirket içinde tekildir. Eğer bir ilaç şirketi silinirse o şirkete ait olan ilaçlar da silinmelidir.
- 5) her eczanenin adı, adresi ve telefon numarası vardır.
- 6) her hastanın bir aile doktoru vardır. Her doktorun en az 1 hastası vardır.
- 7) her eczane birçok ilacı farklı fiyatlardan satmaktadır. Aynı ilaç farklı eczanelerde farklı fiyattan satılabilir.
- 8) doktorlar hastalara ilaç yazarlar. Bir hasta birçok hastaya reçete yazabildiği gibi bir hasta birçok doktordan reçete alabilir. Her reçetenin bir tarihi ve miktarı vardır. Eğer bir doktor aynı hastaya aynı ilaçları birden fazla kez yazıyorsa son reçetenin veritabanında tutulması yeterlidir.
- 9) ilaç şirketleri, eczanelerle uzun süreli anlaşmalar yaparlar. Bir ilaç şirketi birçok eczaneye anlaşma sağlayabildiği gibi bir eczane birçok ilaç şirketiyle anlaşma sağlayabilir.
- 10) eczaneler her bir anlaşmalar için supervisor tutarlar. Supervisor anlaşma süresi bitmeden değiştirilebilir.

## Örnek Uygulama - Eczane

- İsteneler:
- 1) sistemin ER diagramının çizilmesi
- 2) eğer her bir ilaç sabit bir fiyattan satılsaydı sistem modeli nasıl değiştirilmesi gerekirdi
- 3) eğer bir doktorun bir hastaya aynı ilacı farklı zamanlarda yazması durumunda her reçetenin tutulması isteniyorsa nasıl değişiklik yapılmalıdır.

# Örnek Uygulama - Eczane





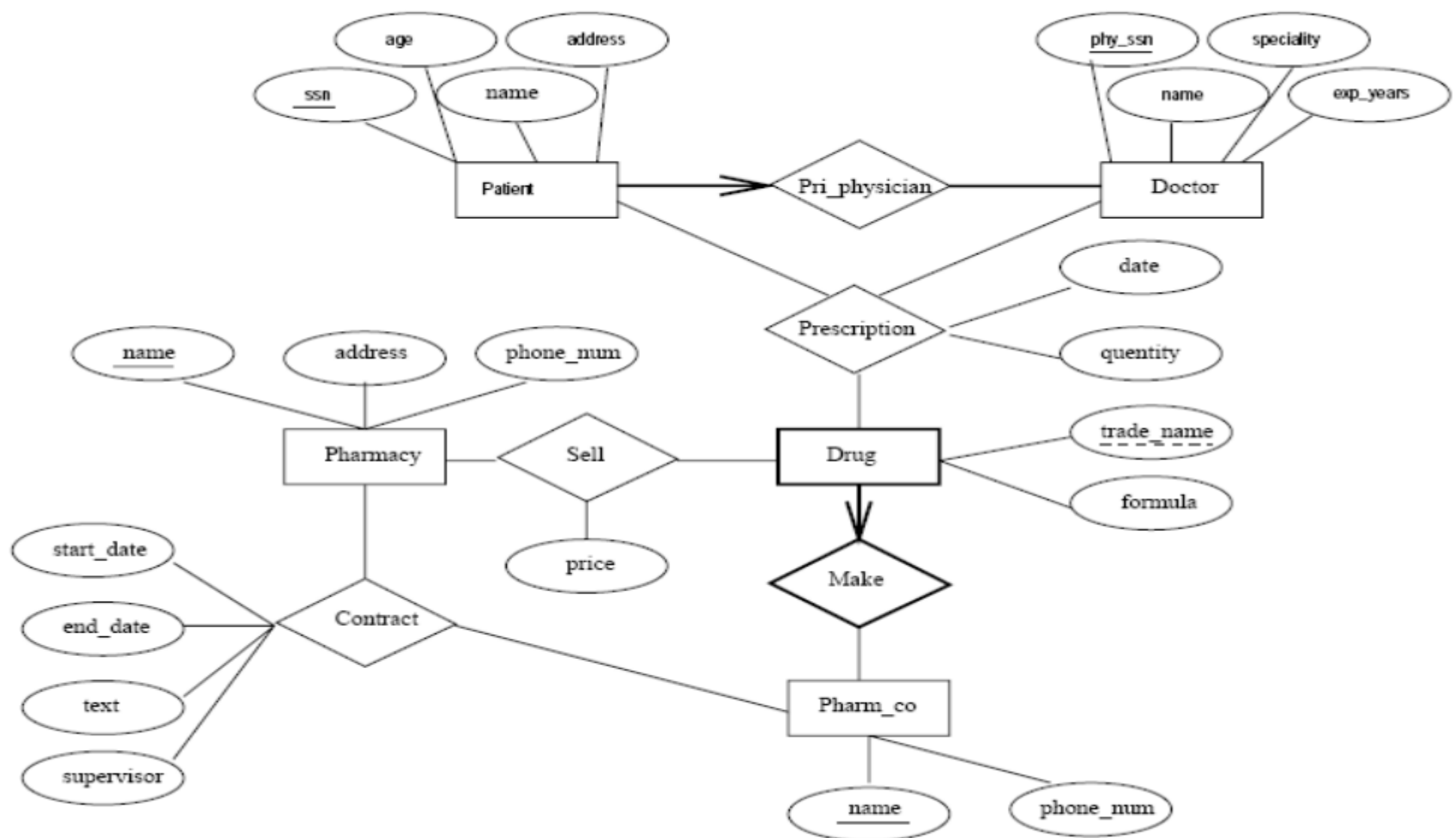
# Örnek Uygulama - Eczane

- Eczane için bir veritabanı modelleneyecektir. Sistemler istenenler:
- 1) hastaların SSN, adı, adresi ve yaşı tutulmalıdır.
- 2) doktorların SSN, adı, uzmanlığı ve deneyim süresi (yıl olarak) tutulmalıdır.
- 3) her ilaç firmasının kendine özel adı ve telefon numarası vardır.
- 4) her ilacın satış adı ve formülü tutulmalıdır. Her ilaç sadece bir ilaç firması tarafından satılmaktadır ve satış adı her ilaç için şirket içinde tekildir. Eğer bir ilaç şirketi silinirse o şirkete ait olan ilaçlar da silinmelidir.
- 5) her eczanenin adı, adresi ve telefon numarası vardır.
- 6) her hastanın bir aile doktoru vardır. Her doktorun en az 1 hastası vardır.
- 7) her eczane birçok ilacı farklı fiyatlardan satmaktadır. Aynı ilaç farklı eczanelerde farklı fiyattan satılabilir.
- 8) doktorlar hastalara ilaç yazarlar. Bir hasta birçok hastaya reçete yazabildiği gibi bir hasta birçok doktordan reçete alabilir. Her reçetenin bir tarihi ve miktarı vardır. Eğer bir doktor aynı hastaya aynı ilaçları birden fazla kez yazıyorsa son reçetenin veritabanında tutulması yeterlidir.
- 9) ilaç şirketleri, eczanelerle uzun süreli anlaşmalar yaparlar. Bir ilaç şirketi birçok eczaneyle anlaşma sağlayabildiği gibi bir eczane birçok ilaç şirketiyle anlaşma sağlayabilir.
- 10) eczaneler her bir anlaşmalar için supervisor tutarlar. Supervisor anlaşma süresi bitmeden değiştirilebilir.

## Örnek Uygulama - Eczane

- İsteneler:
- 1) sistemin ER diagramının çizilmesi
- 2) eğer her bir ilaç sabit bir fiyattan satılsaydı sistem modeli nasıl değiştirilmesi gerekirdi
- 3) eğer bir doktorun bir hastaya aynı ilacı farklı zamanlarda yazması durumunda her reçetenin tutulması isteniyorsa nasıl değişiklik yapılmalıdır.

## Örnek Uygulama - Eczane



## Kaynaklar

- ✓ Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management, Cengage Learning.
- ✓ Ümit Kocabıçak, Ders Notları, Sakarya Üniversitesi Bilgisayar ve Bilişim Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü.
- ✓ Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, Mc Graw Hill