

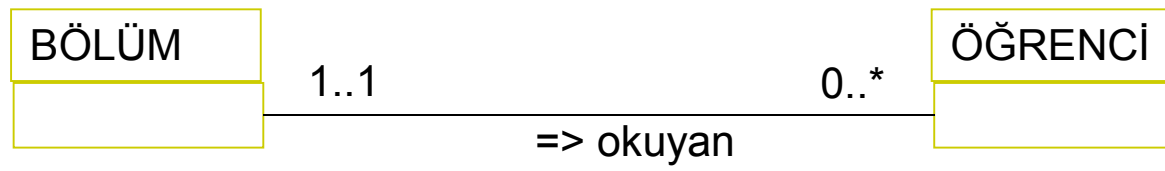
ER Basic Model

- Relationship: Association among two or more entities. E.g., Attishoo **works in** Pharmacy department.
- Relationship Set: Collection of similar relationships.
 - An n-ary relationship set R relates n entity sets $E_1 \dots E_n$;
 - each relationship in R involves entities e_1 in E_1 , ..., e_n in E_n

Participation Constraint

- Does every department have a manager?
 - this is a participation constraint: the participation of Departments in **Manages** is said to be total (vs. partial) (tam katılım yada kısmi katılım)
 - Every did value in Departments table must appear in a row of the Manages table (with a non-null ssn value!)
- *B varlık kümesindeki her bir 'b' varlığı, A varlık kümesindeki bir 'a' varlığı ile mutlaka bağıntılı olması gerekiyorsa, «B varlık kümesi A varlık kümesine var olma bağımlıdır» denir. Burada A birincil varlık, B ikincil varlık olarak adlandırılır.*
- *Bölüm, Çalışan'a var olma bağımlıdır.*

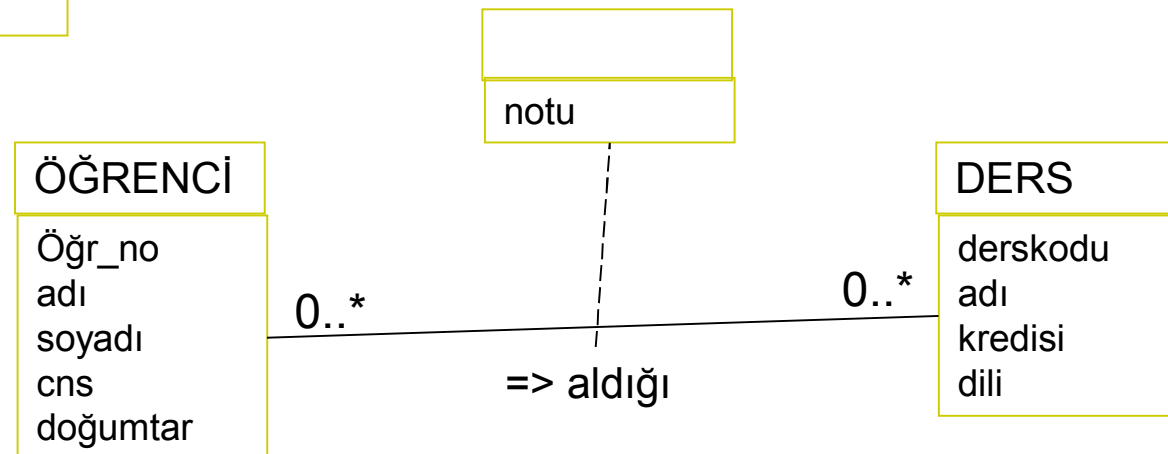
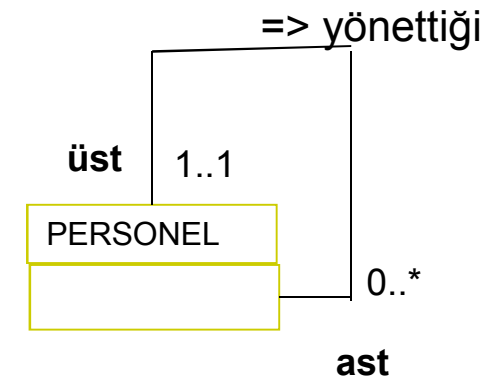
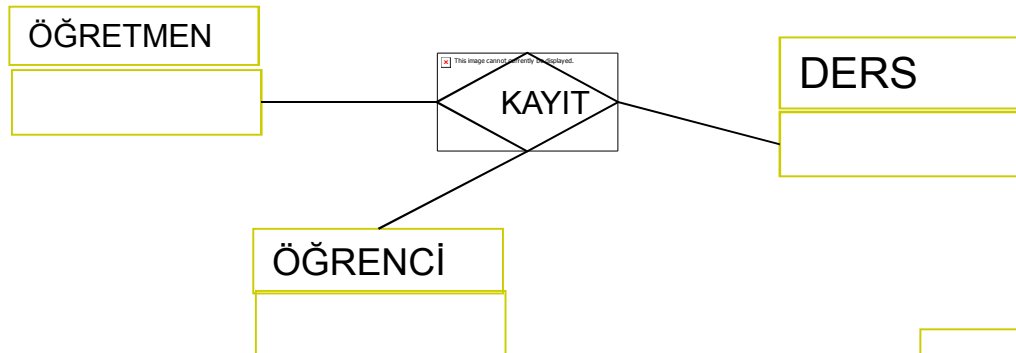
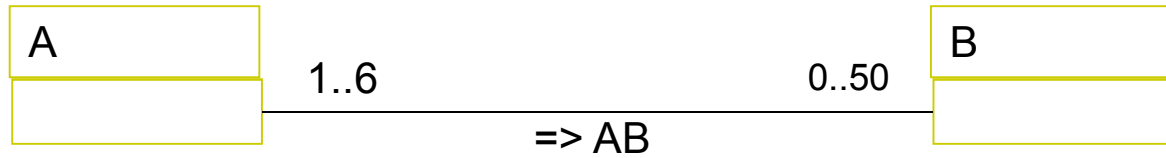
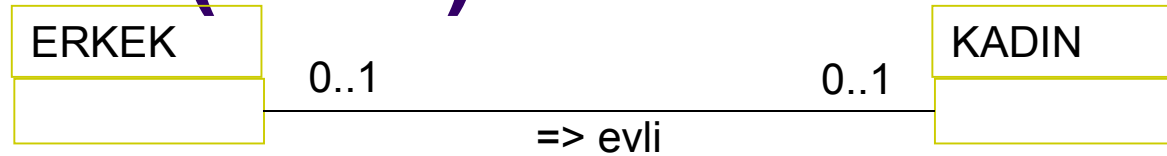
Örnek



Weak Entities

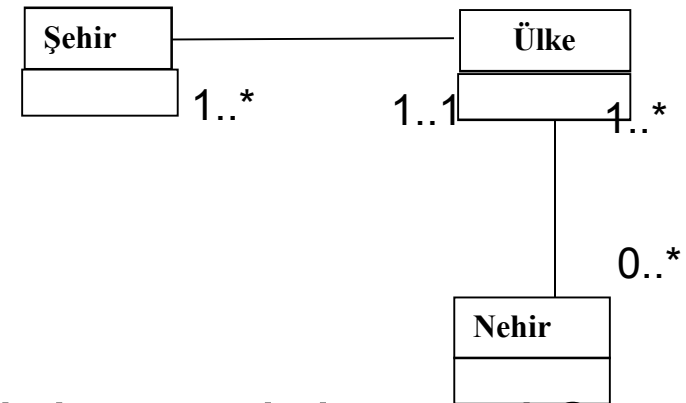
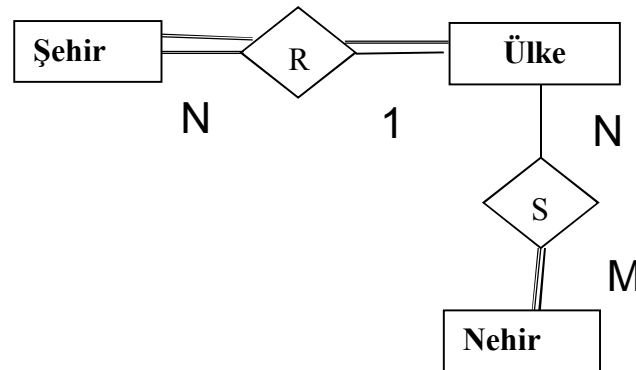
- A weak entity can be identified uniquely only by considering the primary key of another (owner) entity.
 - Owner entity set and weak entity set must participate in a one-to-many relationship set (one owner, many weak entities).
 - Weak entity set must have total participation in this identifying relationship set.

Örnek (UML)

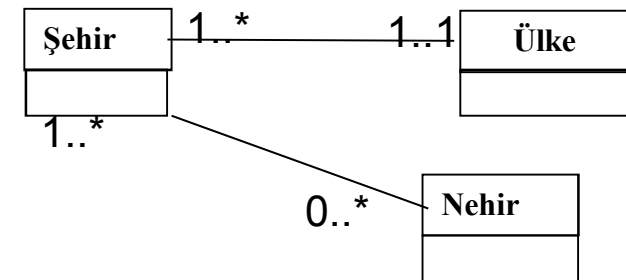
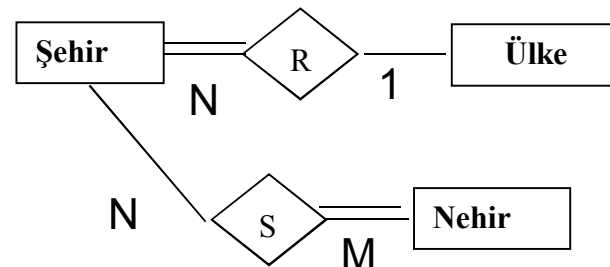


Örnek (ER,UML)

- Şehirler, ülkeler ve nehirleri gösteren bir veritabanı tasarımı.



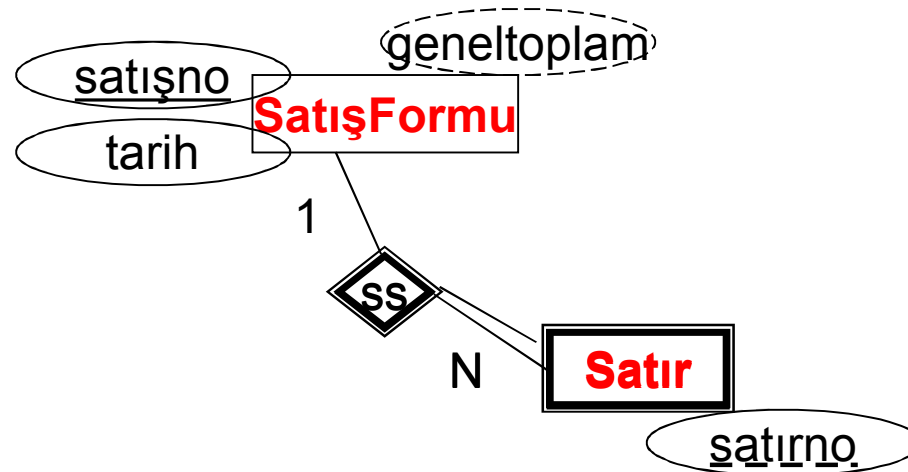
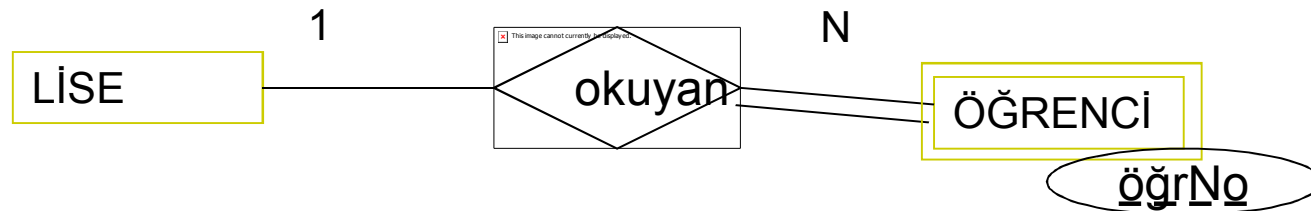
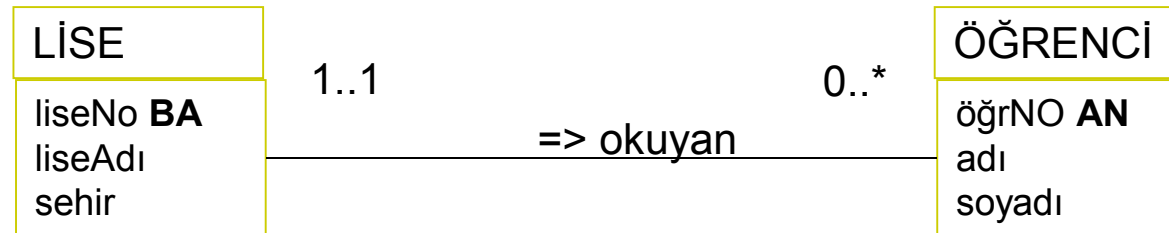
- Nehirlerin hangi şehirlerden geçtiğini tutmak istersek?



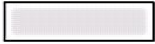
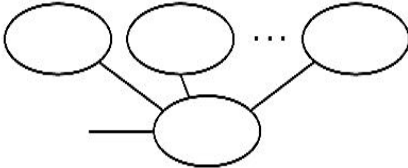
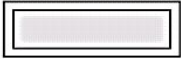
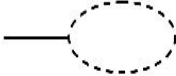
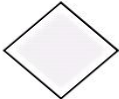
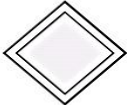
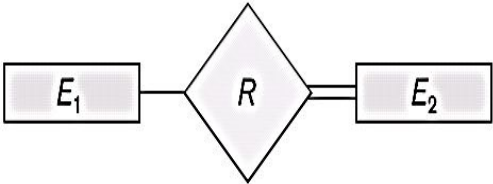
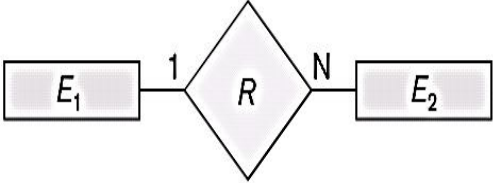
Güçlü/Zayıf Varlık Kümeleri

- Bir varlık kümesinde anahtar bulunamıyorsa (*bütün nitelikler birarada olsa da anahtar olmuyorsa*) bu varlık kümesine «**zayıf varlık kümesi**» denir.
- Anahtarı olan varlık kümesine «güçlü varlık kümesi» denilir.
- Zayıf varlık kümesi, güçlü bir varlık kümesi ile 1-1 veya N-1 tipinde var olma bağımlı bir bağıntı kurmalıdır.
- Zayıf varlık kümesindeki, aynı güçlü varlığa bağlı olan varlıkları ayırd edebilmek için ayırdedici nitelik(ler)-**AN** tanımlamak gerekir.

zayıf varlık kümesi örnekleri



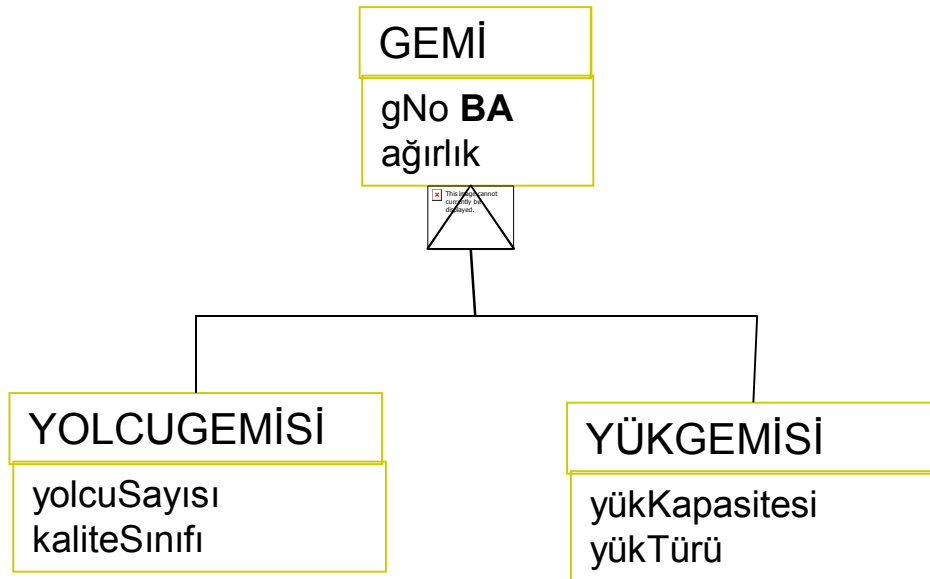
Chen notasyonundaki sembol ve anlamları

Symbol	Meaning		
	Entity		Composite Attribute
	Weak Entity		Derived Attribute
	Relationship		Identifying Relationship
	Attribute		Total Participation of E_2 in R
	Key Attribute		Cardinality Ratio 1: N for $E_1:E_2$ in R
	Multivalued Attribute		

ISA ('is a') Hierarchies

- As in C++, or other PLs, attributes are inherited.
- If we declare A ISA B, every A entity is also considered to be a B entity.
 - Overlap constraints(Ayrıklık kısıtlaması): Can Joe be an Hourly_Emps as well as a Contract_Emps entity?(Allowed/disallowed)
 - Covering constraints(Katılım kısıtlaması): Does every Employees entity also have to be an Hourly_Emps or a Contract_Emps entity?(yes:mandatory/no:optional)
- Reasons for using ISA:
 - To add descriptive attributes specific to a subclass.
 - To identify entities that participate in a relationship.

Genelleme («/S A», «Ait olma» bağıntısı)



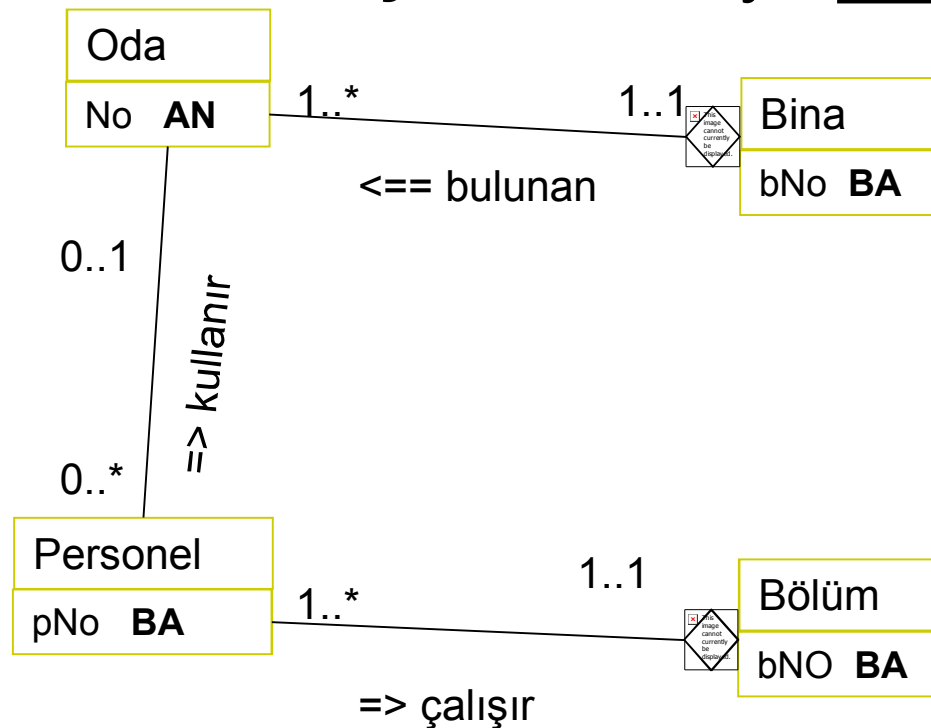
- Kalıtım: nitelik üst varlıktan alt varlığa geçer
- Katılım kısıtlaması:
 - Zorunlu: üst sınıfın her ögesi alt sınıfta yer alır
 - Seçimli: yer almayabilir
- Ayırıklık kısıtlaması:
 - OR: üst sınıfın her ögesi en çok bir alt sınıfta var (alt sınıflar: disjoint-ayrık)
 - AND: birden çok alt sınıf
- 4 farklı kısıtlama tanımlanabilir, ihtiyaca göre kullanılabilir.

Aggregation

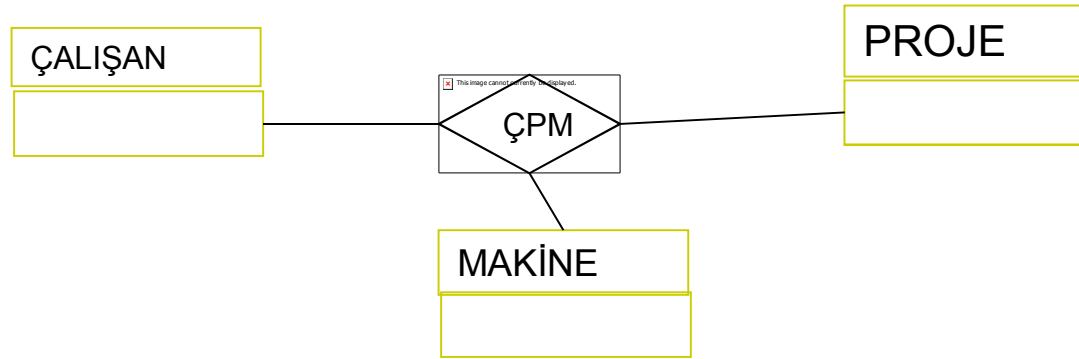
- Used when we have to model a relationship involving (entity sets and) a relationship set.
- Aggregation allows us to treat a relationship set as an entity set for purposes of participation in (other) relationships.

Kümeleme-1 (*aggregation*)

- «has a», «bulunur/sahiptir» bağıntısı
- ‘Küme’ ile ‘Parçaları’ arasındaki bağıntılar için kullanılır. Parçalar kümeye **var olma bağımlıdır**.



Kümeleme (çok dereceli bağıntılar)



- Yukarıdaki 3'lü bağıntıda:
 - İki varlık arasındaki bağıntı, 3. varlıktan bağımsız bir şekilde ifade edilemiyor.
 - (Kavramsal seviyede de olsak) Bilgi tekrarı var.
 - Bilgi tekrarıdan dolayı aykırılık/anormallikler ortaya çıka(bili)r.
- **Kümeleme**, varlık kümeleri ile aralarındaki bağıntının kümelenmesi ile yeni bir varlık kümesi tanımlamaya olanak sağlar.

Conceptual Design Using the ER Model

- Design choices:
 - Should a concept be modeled as an entity or an attribute?
 - Should a concept be modeled as an entity or a relationship?
 - Identifying relationships: Binary or ternary? Aggregation?
- Constraints in the ER Model:
 - A lot of data semantics can (and should) be captured.
 - But some constraints cannot be captured in ER diagrams.

Entity vs. Attribute

- Should address be an attribute of Employees or an entity (connected to Employees by a relationship)?
- Depends on the use we want to make of address information, and the semantics of the data:
 - If we have several addresses per employee, address must be an entity (since attributes cannot be set valued).
 - If the structure (city, street, etc.) is important, e.g., we want to retrieve employees in a given city, address must be modeled as an entity (since attribute values are atomic).

Summary of Conceptual Design

- Conceptual design follows requirements analysis,
- Yields a high-level description of data to be stored
 - ER model popular for conceptual design
 - Constructs are expressive, close to the way people think about their applications.
 - Basic constructs: entities, relationships, and attributes (of entities and relationships).
- Some additional constructs: weak entities, ISA hierarchies, and aggregation.
- Note: There are many variations on ER model.

Summary of ER (Contd.)

- Several kinds of integrity constraints can be expressed in the ER model: key constraints, participation constraints. Some foreign key constraints are also implicit in the definition of a relationship set.
- Some constraints (notably, functional dependencies) cannot be expressed in the ER model.
- Constraints play an important role in determining the best database design for an enterprise.

Summary of ER (Contd.)

- ER design is subjective. There are often many ways to model a given scenario! Analyzing alternatives can be tricky, especially for a large enterprise.

Common choices include:

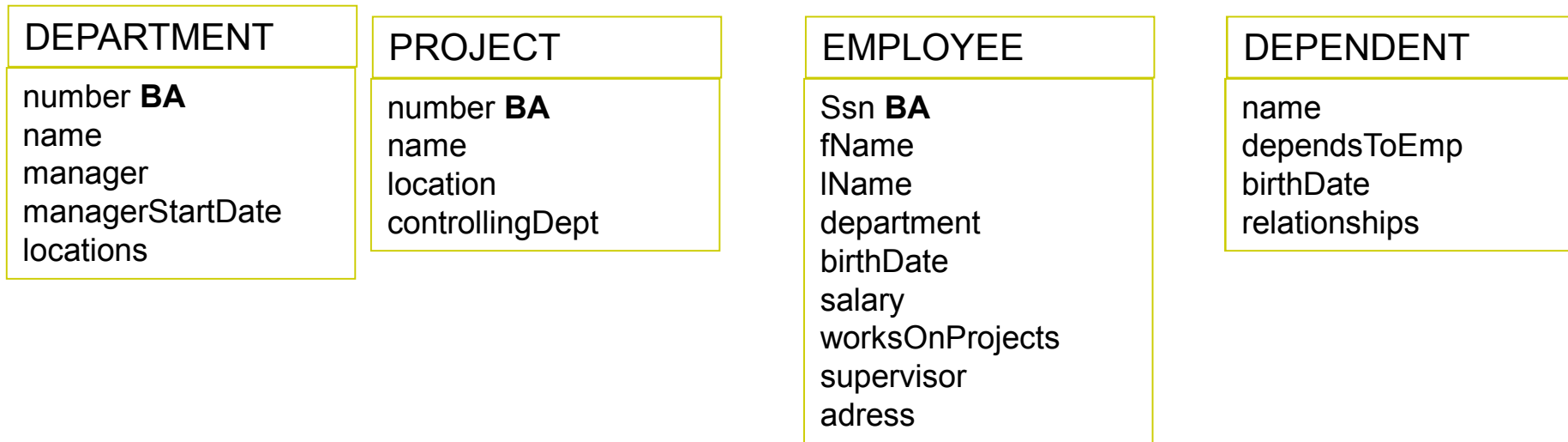
- Entity vs. attribute, entity vs. relationship, binary or n-ary relationship, whether or not to use ISA hierarchies, and whether or not to use aggregation.
- Ensuring good database design: resulting relational schema should be analyzed and refined further. FD information and normalization techniques are especially useful.

Örnek: COMPANY Database

- We need to design a database schema based on the following (simplified) **requirements** of the COMPANY Database:
 - The company is organized into **DEPARTMENTS**. Each department has a name, number and an employee **who manages the department**. We keep track of the start date of the department manager. A department may have several locations.
 - Each department **controls** a number of **PROJECTs**. Each project has a unique name, unique number and is located at a single location.
 - We store each **EMPLOYEE's** social security number, address, salary, and birthdate.
 - Each employee **works for** one department but may **work on** several projects.
 - We keep track of the number of hours per week that an employee currently works on each project.
 - We also keep track of the **direct supervisor** of each employee.
 - Each employee may **have** a number of **DEPENDENTs**.
 - For each dependent, we keep track of their name, sex, birthdate, and relationship to the employee.

Initial Design of COMPANY DB

- Based on the requirements, we can identify four initial entity types in the COMPANY database:
 - DEPARTMENT, PROJECT, EMPLOYEE and DEPENDENT
- The initial attributes shown are derived from the requirements description



İlk tasarım sonrası iyileştirmeler

- *Eğer mevcut nitelik başka bir varlığa işaret ediyorsa:*
 - Nitelik → Bağıntı haline dönüşmeli.
- *Eğer mevcut nitelik çok değer alabiliyorsa:*
 - Nitelik → Varlık kümesi haline dönüşmeli.
- *Eğer mevcut varlık kümesi sadece 1 niteliğe sahip ve bir bağıntısı varsa:*
 - Varlık kümesi → nitelik haline dönüşmeli.
- Varlık kümeleri arasında «**genelleme**» mümkünse yapılmalı. Yukardan-aşağıya (top-down), aşağıdan-yukarıya (bottom-up) yaklaşımlar kullanılmalı.
- Bağıntıların dereceleri tekrar değerlendirilmeli, gerekirse «**kümeleme**» ile daha açık yapılanmalar kullanılmalı.

COMPANY DB (devam)

- In the refined design, some attributes from the initial entity types are refined into relationships:
 - Manager of DEPARTMENT -> MANAGES
 - Works_on of EMPLOYEE -> WORKS_ON
 - Department of EMPLOYEE -> WORKS_FOR
 - Controlling_Department of PROJECT → CONTROLS
 - Supervisor of EMPLOYEE → SUPERVISION
 - Dependent_name of DEPENDENT → DEPENDENTS_OF
- In general, more than one relationship type can exist between the same participating entity types(MANAGES and WORKS_FOR are distinct relationship types between EMPLOYEE and DEPARTMENT (Different meanings and different relationship instances.)

COMPANY DB ER DIAGRAM

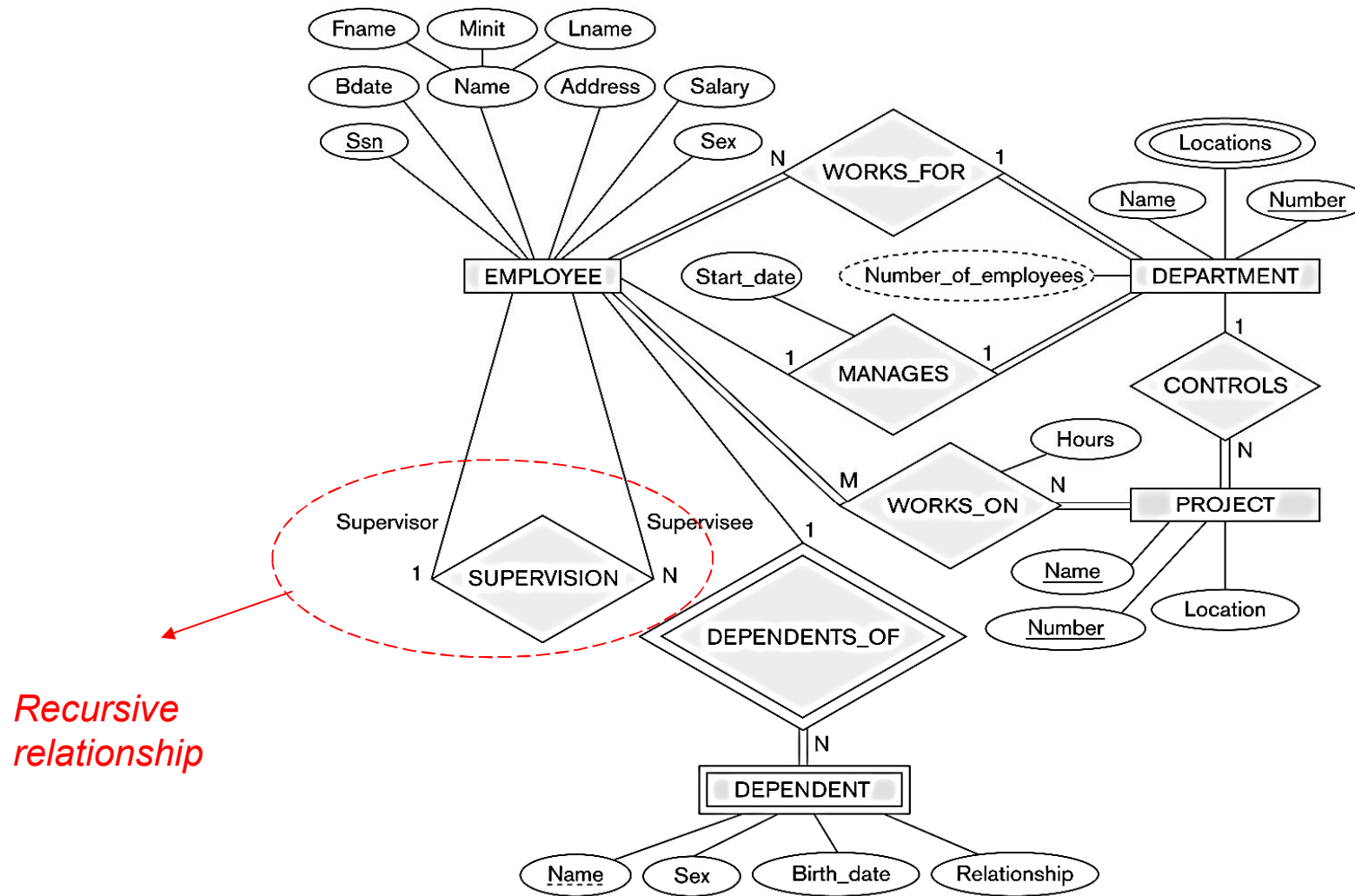
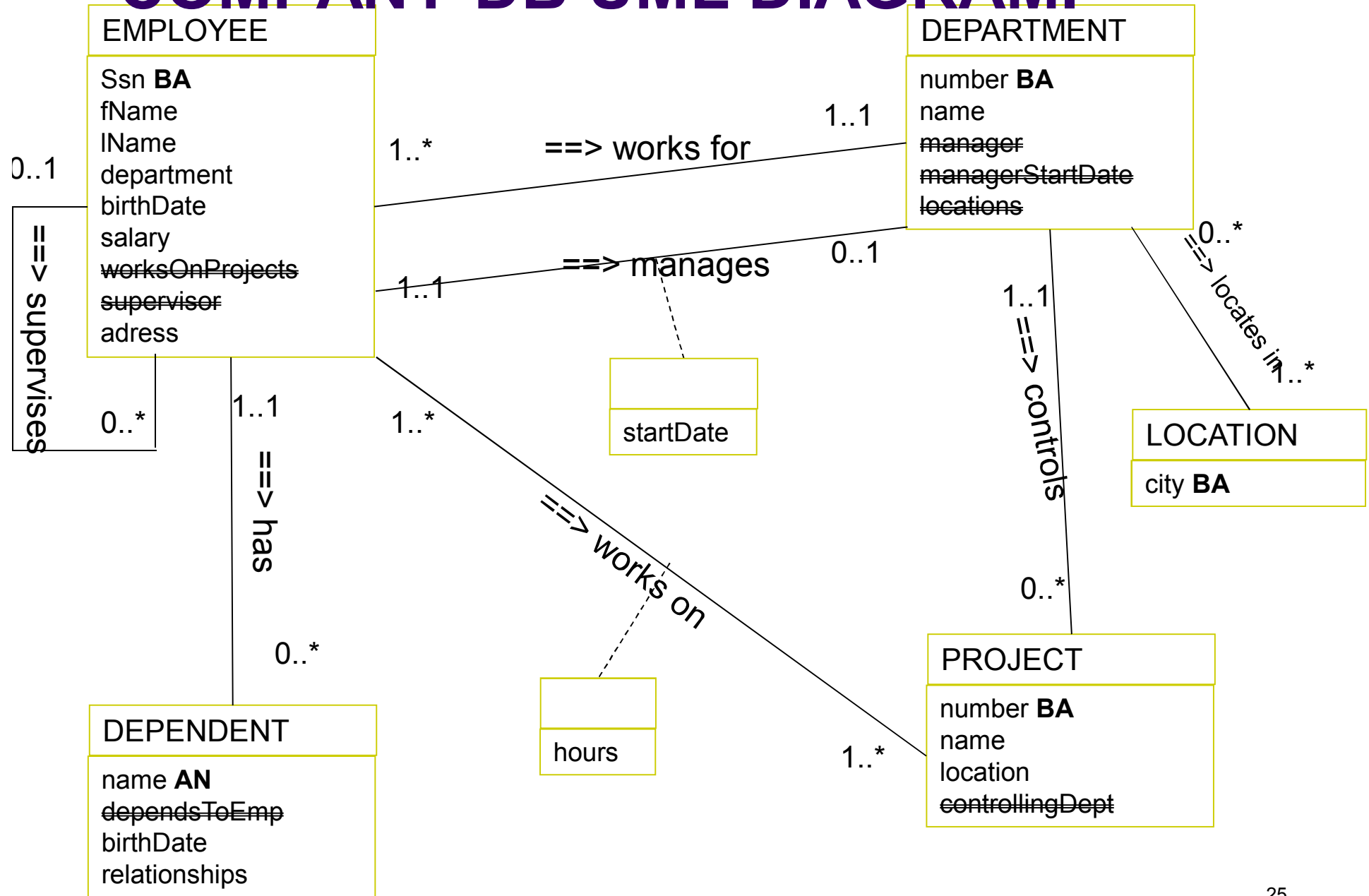


Figure 3.2

An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

COMPANY DB UML DIAGRAMI



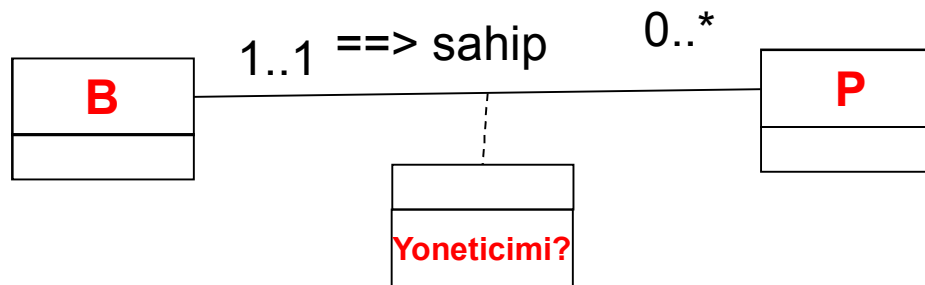
Örnek 2

Bir şirketteki bölümler ve personel bilgilerini tutan Bölüm-Personel veri tabanını düşünelim.

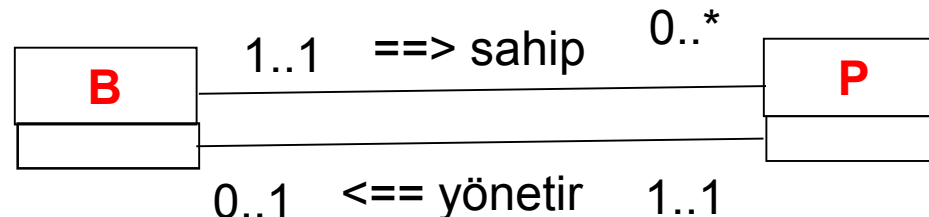
- Her **bölüm** çok sayıda **personele** sahiptir ve her personel mutlaka bir bölüme bağlıdır.
- Bir bölümün sadece bir yöneticisi (*mutlaka*) vardır.

Buna göre aşağıdaki durumları *birbirinden bağımsız olarak* cevaplayınız...

- Bir bölümün yöneticisi ancak o bölüme bağlı bir personel olabiliyor ise, Nasıl bir ER modeli çizilebilir?



- Bir bölümün yöneticisi herhangi bir personel olabiliyor ise, Nasıl bir ER modeli çizilebilir?

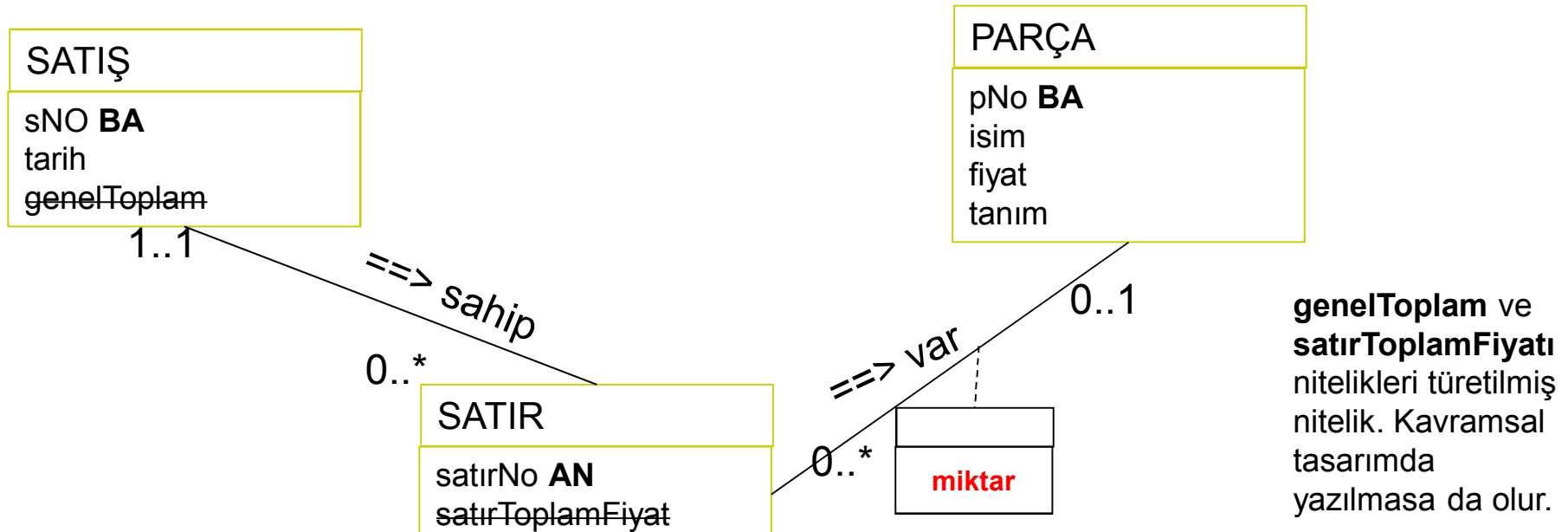


Örnek 3

SATIŞ FORMU						
SatışNO: 12						Tarih: 11.06.2009
SatırNo	ParçaNo	Parçaİsim	Tanım	Miktar	Birim Fiyat	Toplam
1	23645	ayakkabı	Spor,basketbo 1	1	110	110
2	12674	gomlek	klasik tarz	2	80	160
3	...		...			
					GenelTopl am:	880

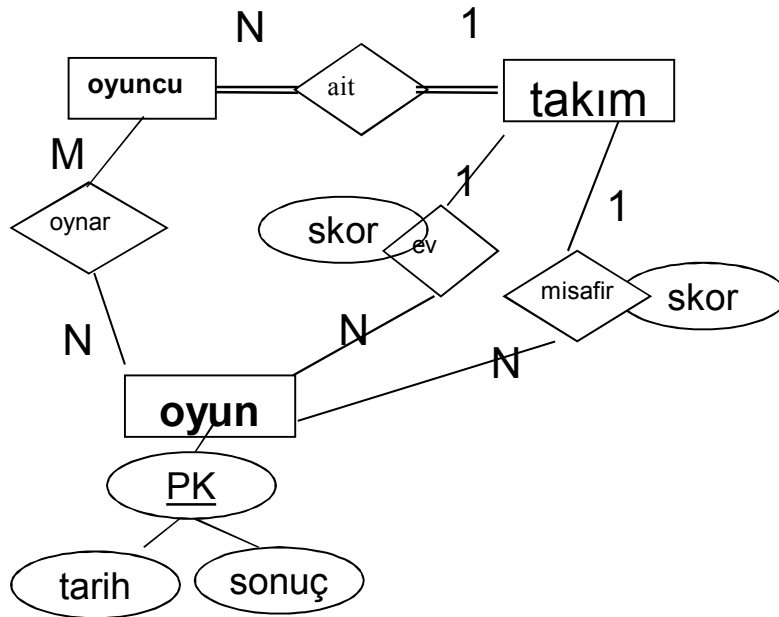
Yukarıda satış bilgileri tutan örnek bir satış formu içeriği görülmektedir. Buna karşılık gelen veri tabanı, **SATIŞ**, **PARÇA** ve **SATIR** olmak üzere 3 varlık setinden oluşmaktadır. (Aynı ParçaNolu parçalar farklı satırlarda tekrar edebilir.)

Bu varlık setlerini, özelliklerini ve aralarındaki ilişkileri gösteren ER diyagramını çiziniz.



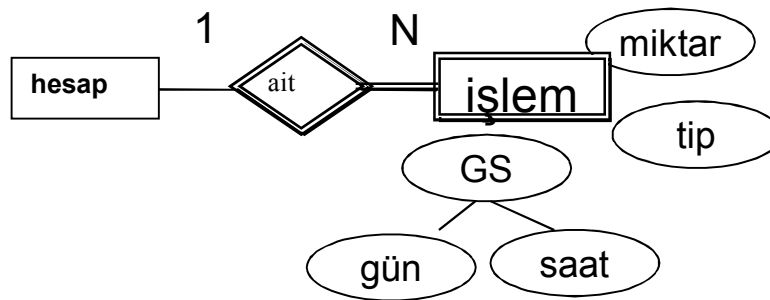
Örnek 4

- Birden çok oyuncu içeren takımlar arasında, iki takım içeren oyunlar ile ilgili bilgiler tutulmak isteniyor. (*Oyuncu en fazla bir takımın elemanı olabilir.*)
- her oyunda hangi takımların yer aldığı (hangi takım ev sahibi hangisi misafir olduğu) ve oyunun tarihi ve sonucu gibi bilgiler tutuluyor.
- her oyunda takımın hangi oyuncularının yer aldığı bilgisi takip ediliyor.



Örnek 5

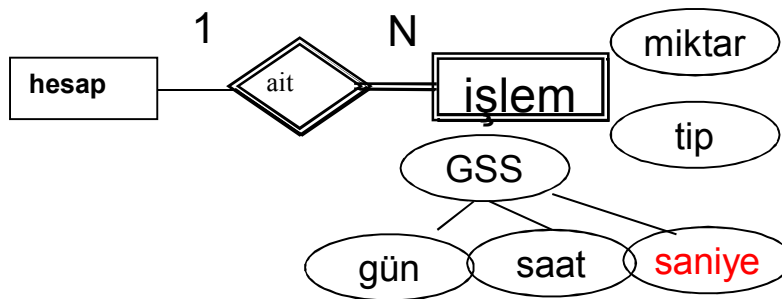
- Bir banka hesabına ait işlemler hakkında işlem zamanı(*gün, saat*), işlem tipi ve miktar bilgileri tutuluyor. bu olayı takip eden ER diyagramı:



Not: *gün, saat*
partial key
oluyor.

**Farklı hesaplarda aynı saat içinde
aynı miktar ve tip işlem olabilir!!!**

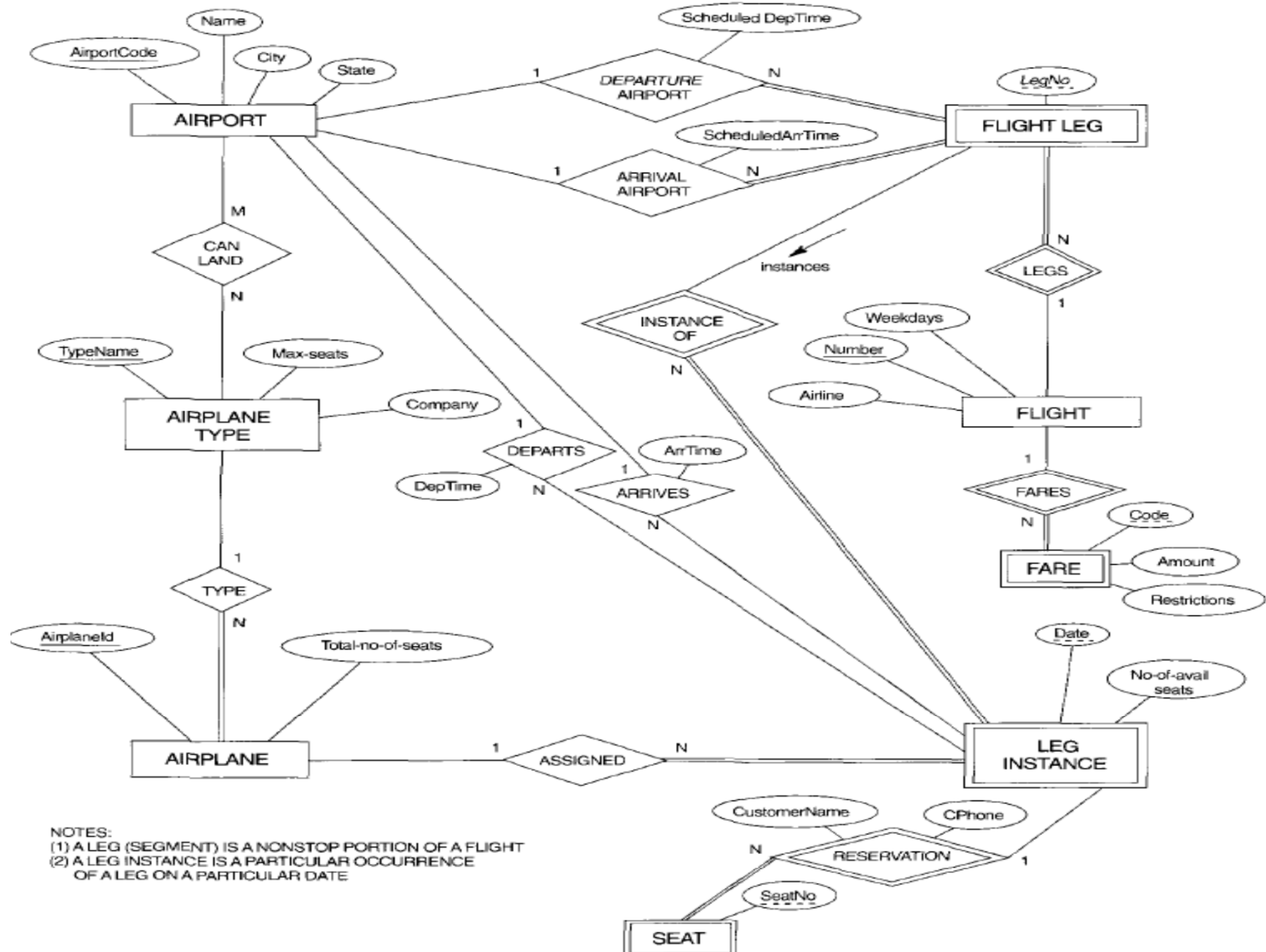
- Yukarıdaki çözüm, aynı hesap üzerinde aynı saat içinde işlem yapılmaması durumunda geçerlidir. Bu koşul kalktığı zaman, tasarım işlem zamanı olarak sistem zamanını saklamalıdır;



Not:
gün, saat, saniye
partial key
oluyor.

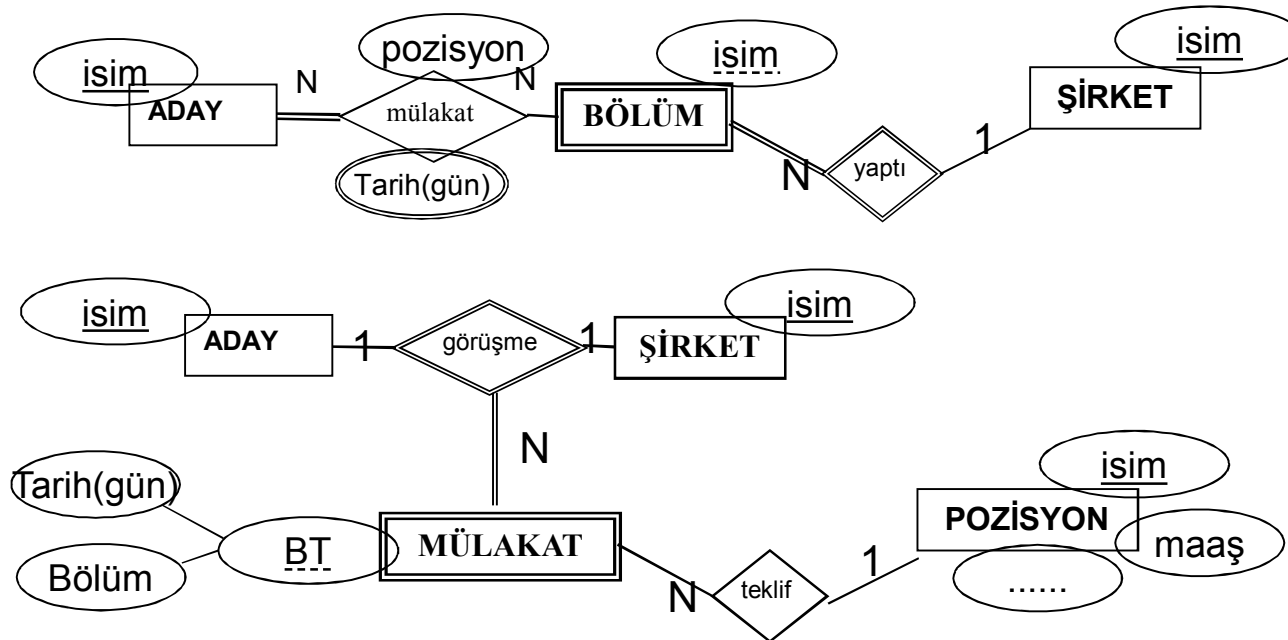
Farklı hesaplarda aynı **saniye içinde
aynı miktar ve tip işlem olabilir!!!**

Örnek 6: Figure 3.17, Elmasri/Navathe book



Örnek 7

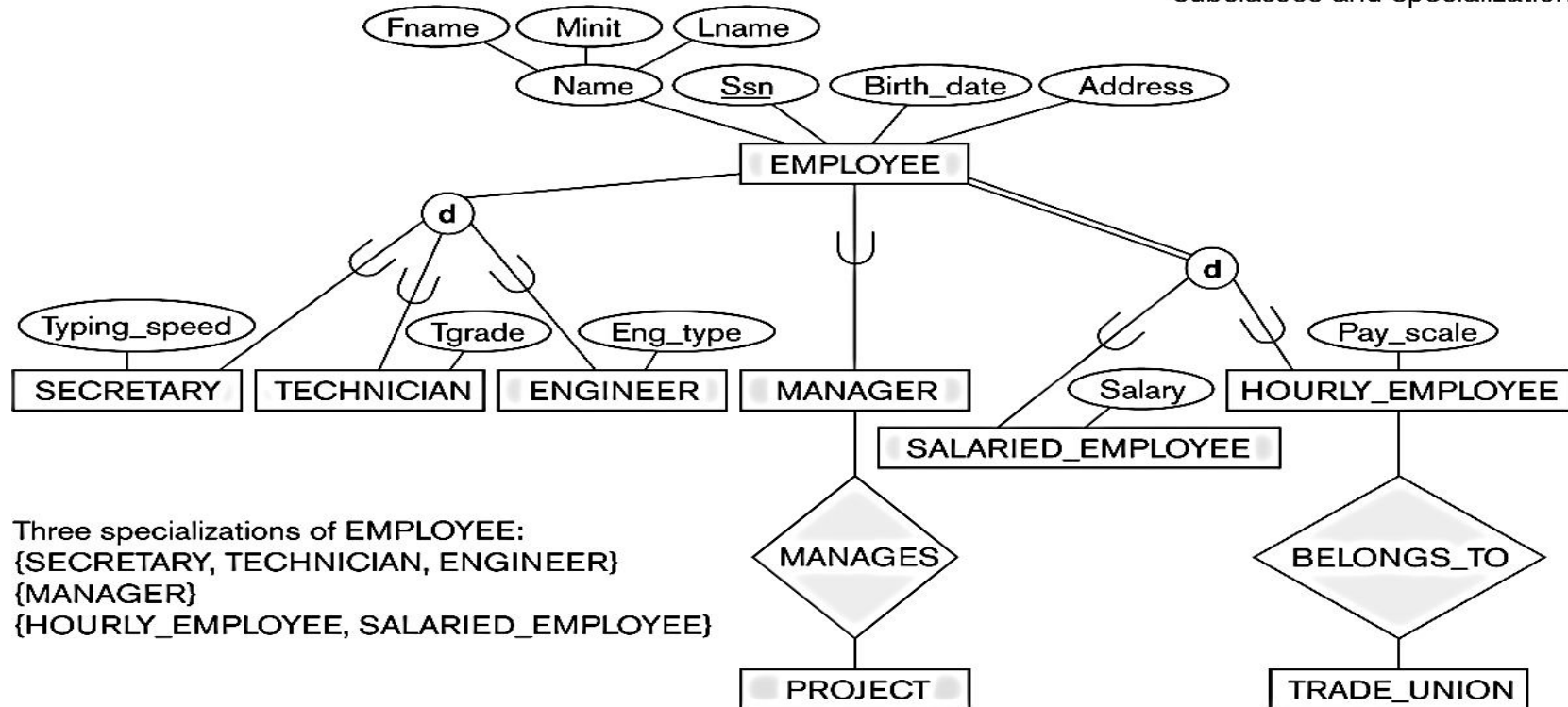
- Adayların şirketler ile yaptıkları görüşme ve teklif edilen iş pozisyonunu takip etmek istiyoruz.
 - Aday bir şirketin birden çok bölümü ile görüşme yapabilir ve her bölümden farklı pozisyon teklifler alabilir; bir bölümden ancak 1 pozisyon teklifi alabilir.
 - Teklif edilen pozisyonun maası, özellikleri gibi bir takım özellikler daha tutulmak isteniyor..
 - Görüşmenin yapıldığı gün (yıl/ay/gün) bilgisi tutluyor. Aynı bölüm ile görüşme ancak farklı bir günde olabiliyor.
 - Gerek aday isimleri gerek şirket isimleri sistemde biricik olduğunu varsayabiliriz..
 - Bölüm hakkında bir bilgi tutulmuyor.



Örnek 8 (genelleme)

Figure 4.1

EER diagram notation to represent subclasses and specialization.



- ER diyagramında IS A (AİT OLMA) bağıntısı
 - Katılım, zorunlu ise çift çizgi ile, zorunlu değilse tekçizgi ile
 - Ayırıklık, *OR(disjoint)* ise yuvarlak içinde **d** ile, AND (*notdisjoint*) ise yuvarlak içinde **o** ile ifade ediliyor.

Örnek 9 (genelleme)

(b)

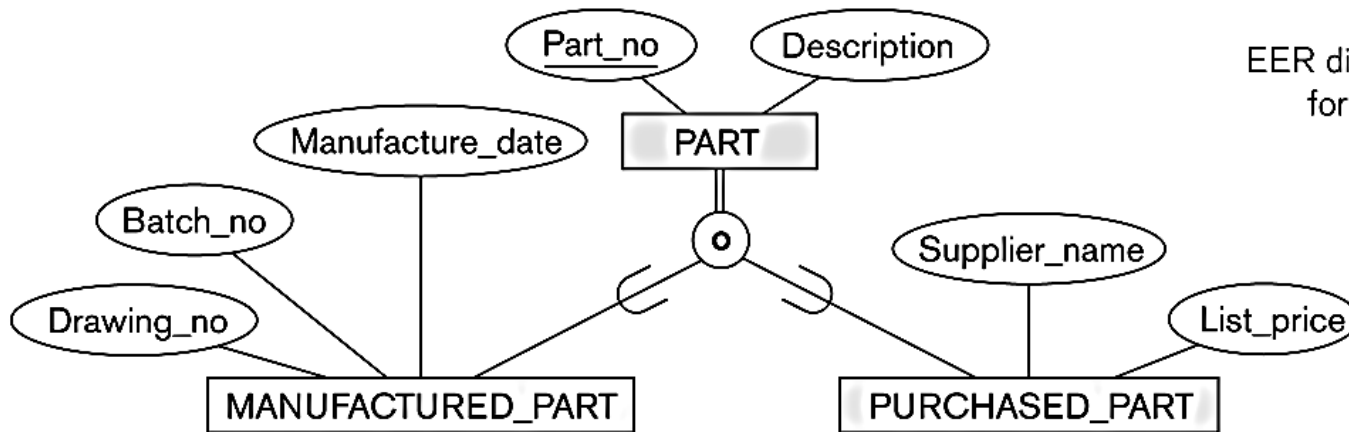
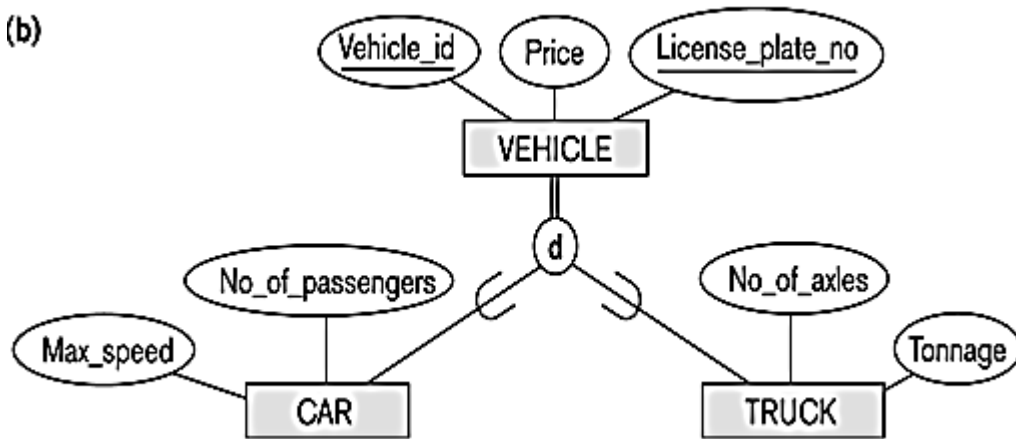


Figure 4.5
EER diagram notation
for an overlapping
(nondisjoint)
specialization.

Örnek 10 (genelleme)

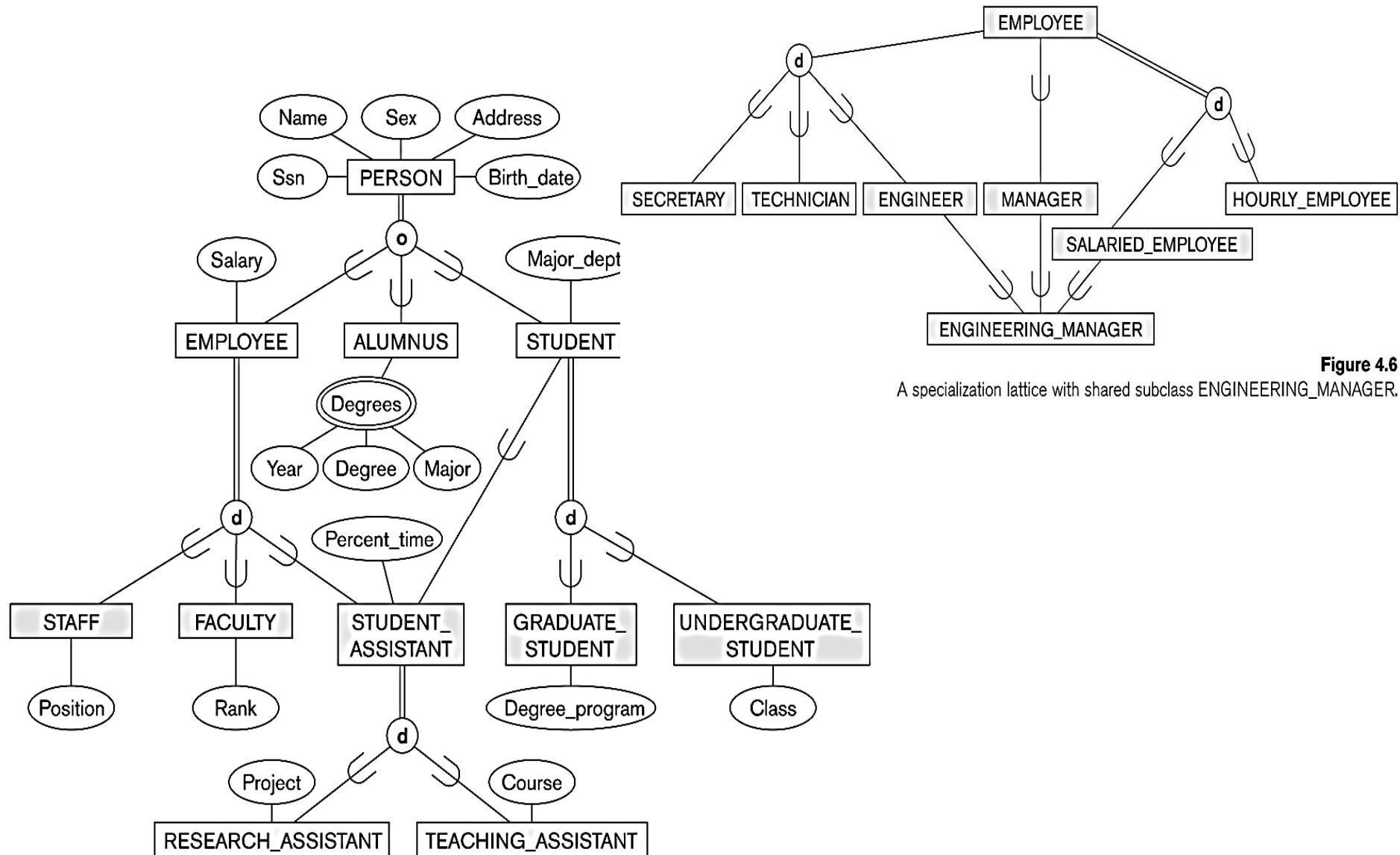


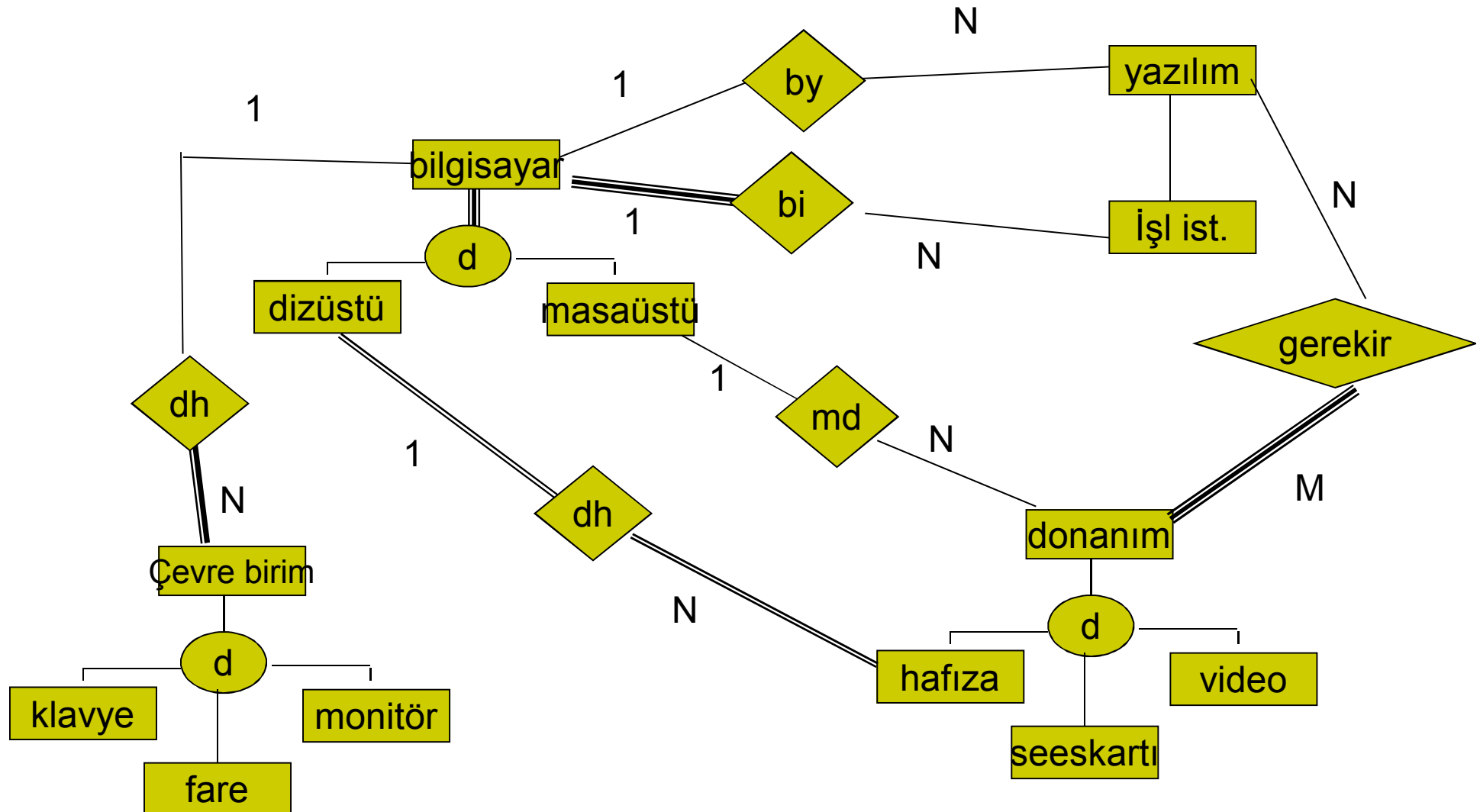
Figure 4.7

A specialization lattice with multiple inheritance for a UNIVERSITY database.

Figure 4.6

A specialization lattice with shared subclass ENGINEERING_MANAGER.

Örnek 11: *bilgisayar satis VT*



Uygun nitelikler belirleyebilirsiniz...