

Veritabanı Yönetim Sistemleri (335)

Dr. Öğr. Üyesi Ahmet Arif AYDIN

L6-

Varlık-İlişki Modeli

Entity Relationship (ER) Model

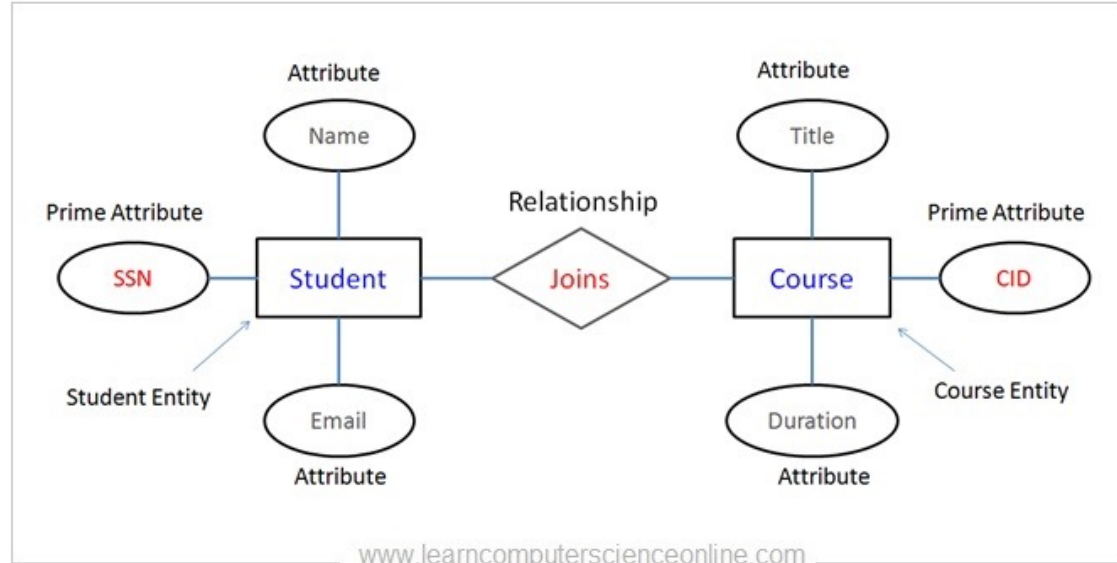
GÜZ -2022

Bölüm Soruları

- Veritabanı Yönetim Sistemleri(VTYS) kullanıcılarına örnek veriniz.
- VTYS'nin kuramsal seviyeleri nelerdir?
 - Harici, Kavramsal ve Fiziksel şemanın görevlerini açıklayınız .
- VTYS kullanmanın dezavantajları nelerdir?
- Veritabanı Tasarım aşamalarını açıklayınız?
- Metadata nedir ?
- Veri bağımsızlığı (data independence) nedir?
- Dosya sistemleri yerine DBMS kullanmanın avantajları nelerdir?
- Dosya sistemlerinin veri kaydedilmesindeki problemleri nelerdir?

Varlık-İlişki Modeli (Entity-Relationship Model)

Entity Relationship Diagram (ERD)



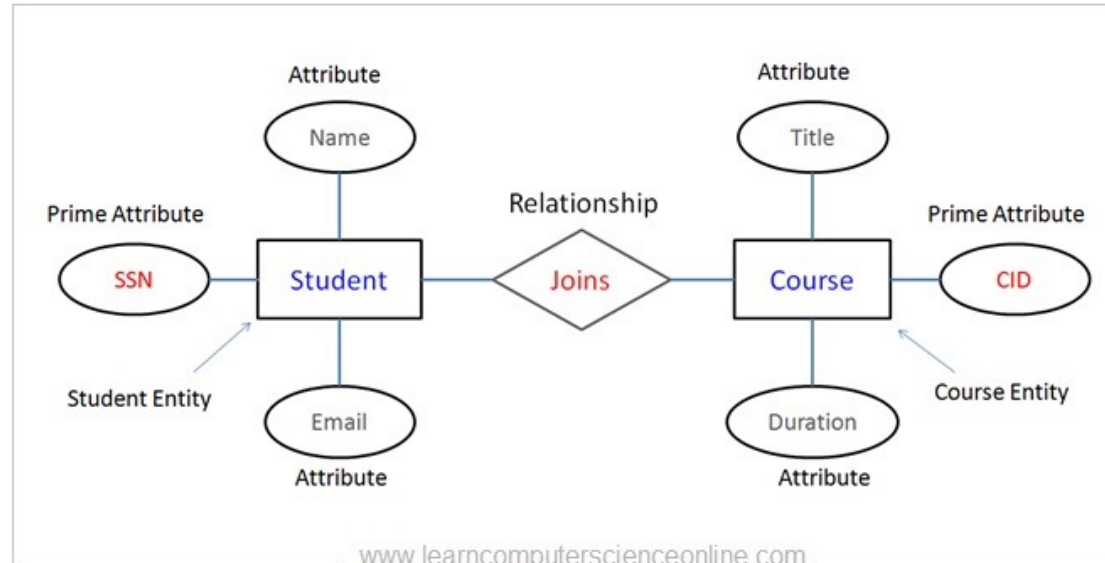
- Varlık (entity) ve varlıklar arasındaki ilişkiyi (relationship) diyagramlar yardımıyla görselleştirerek tanımlamayı sağlayan ve yaygın olarak kullanılan bir *kavramsal veri modeli* (conceptual data model) ER Model dir.
- ER modeli bir veritabanının kavramsal yapısının görsel olarak sunulmasıdır.

<https://www.learncomputerscienceonline.com/entity-relationship-diagram/>

Varlık-İlişki Modeli (Entity-Relationship Model)

- Bir veritabanının ihtiyaçlar belirlendikten sonra veritabanında depolanacak *veriyi modellemek*, *nesneleri tanımlamak* ve *nesneler arasındaki ilişkileri* belirtmek için kullanılır.
- ER modelini oluşturmak kullanıcıların istedikleri doğrultusunda bir veritabanını oluşturma sürecinde ki ilk somut aşamadır.

Entity Relationship Diagram (ERD)



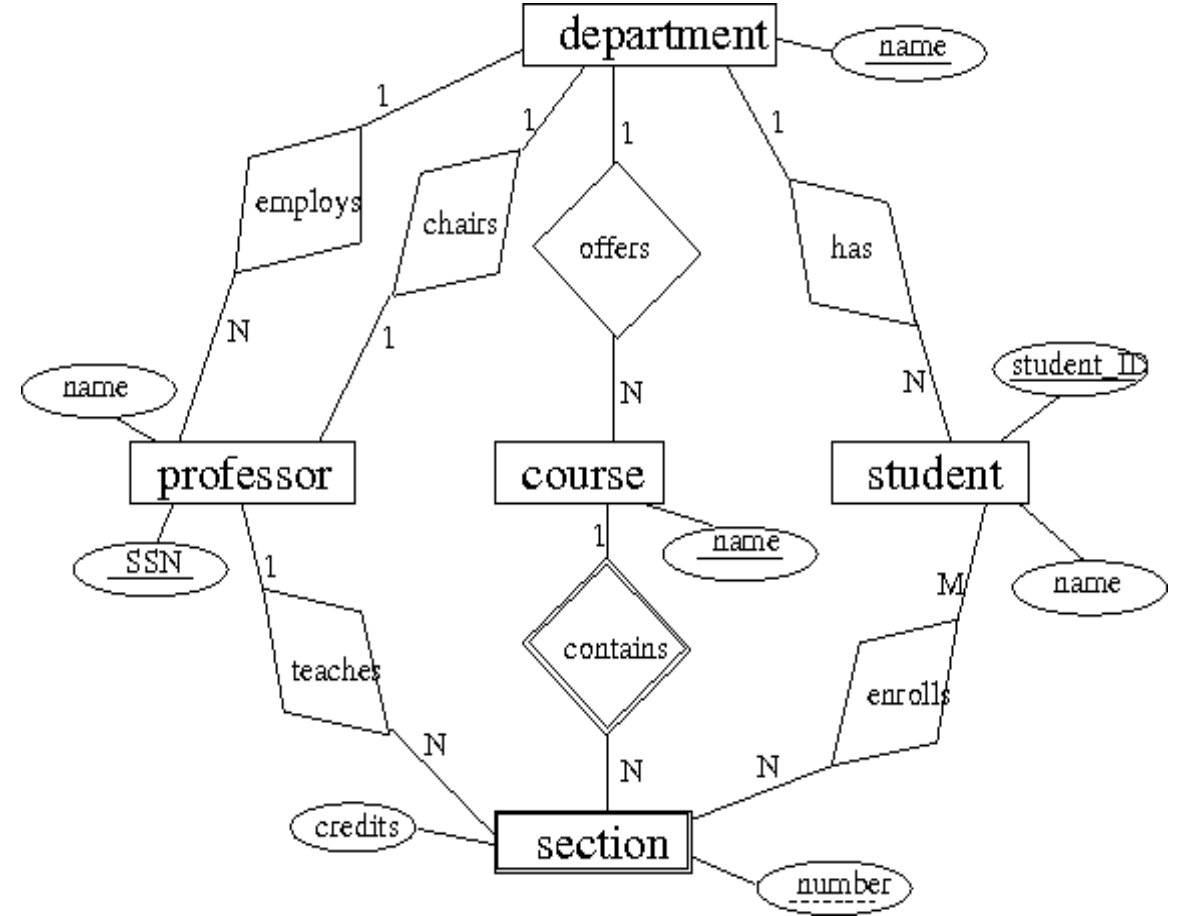
<https://www.learncomputerscienceonline.com/entity-relationship-diagram/>

Varlık-İlişki Modeli : Varlık

Gerçek hayatta bulunan ve diğer nesnelerden

Ayrt edilebilen (distinguishable) nesnelere **varlık** (**entity**) denir.

- 335 VTYS Dersi
- 1212 nolu öğrenci
- Bitirme Projesi I
- Prof. Dr. hocalarımız

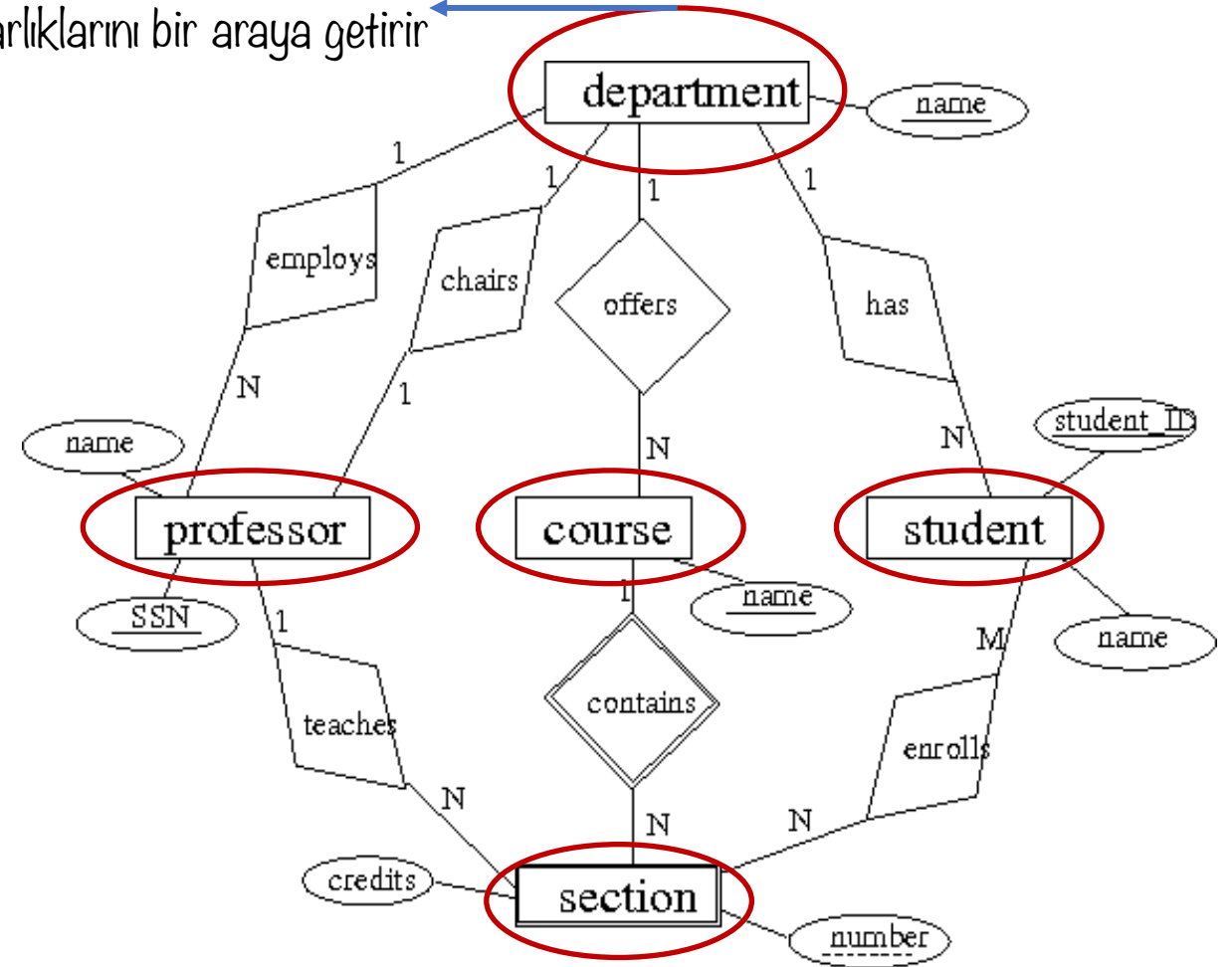


Varlık-İlişki Modeli : Varlık Seti

department bir varlık setidir.
Bölüm varlıklarını bir araya getirir

Ortak özellikleri bulunan aynı tipteki nesnelerin oluşturduğu kolleksiyona varlık seti (entity set) denir.

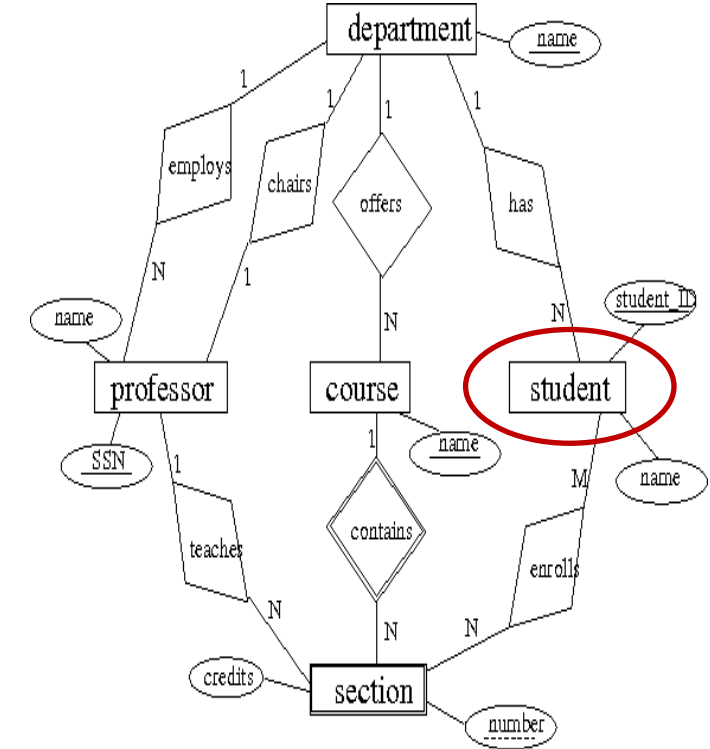
- department
- Professor
- Course
- student



Varlık-İlişki Modeli : Varlık Seti (entity set)

Varlık seti ayık olmayabilir

- Öğrenciler varlık setinde 1. ve 4. sınıftan öğrenciler bulunabilir.
- İnönü üniversitesi bünyesinde bulunan öğretim üyeleri farklı bölümlerde olmalarına rağmen Akademik Personel varlık seti içerisinde değerlendirilebilir



76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor

98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

student

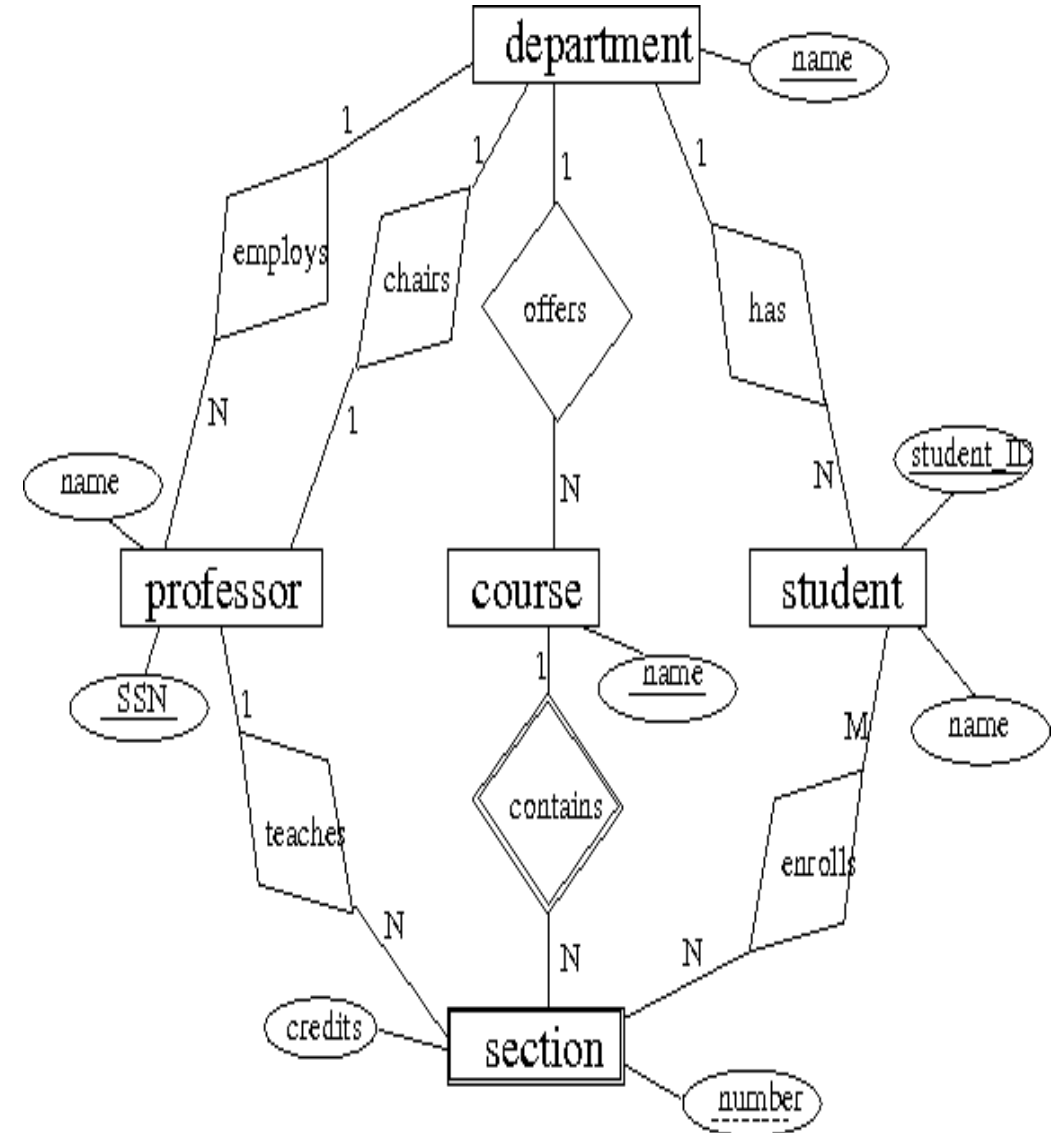
Varlık-İlişki Modeli : İlişki (relationship)

Varlıklar arasındaki etkileşim

ilişki (relationship) olarak tanımlanır.

- Bölüm ders açar (offers)
- öğrenciler derslere kayıt olur (enrolls)
- Prof. dersleri okutur (teaches)
- İdari personel bölüm de çalışır
- Prof. Bölümü yönetir (chairs)

İlişkileri bulunmaktadır



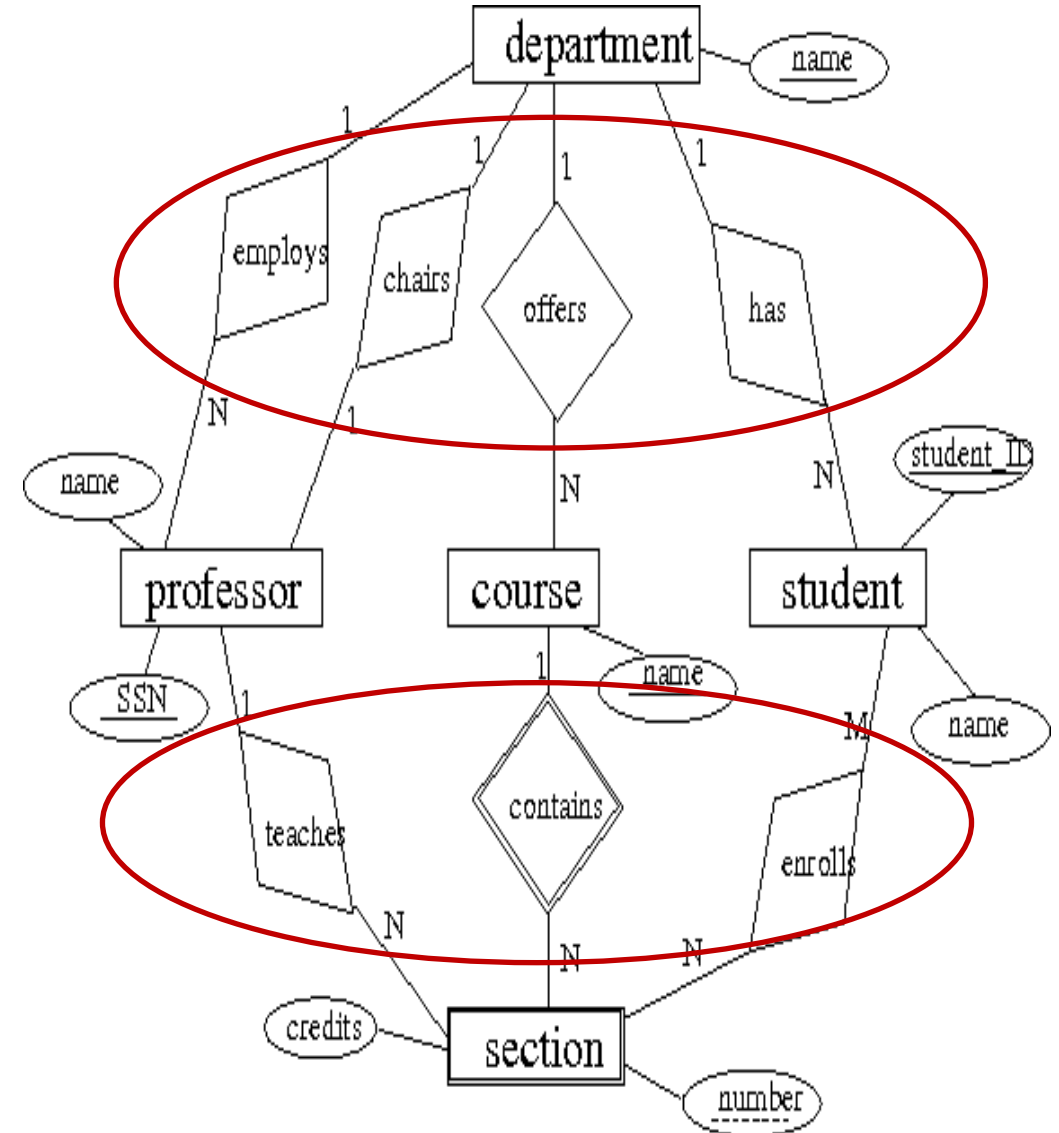
Varlık-İlişki Modeli : İlişki (relationship)

Varlıklar arasındaki etkileşim

ilişki (relationship) olarak tanımlanır.

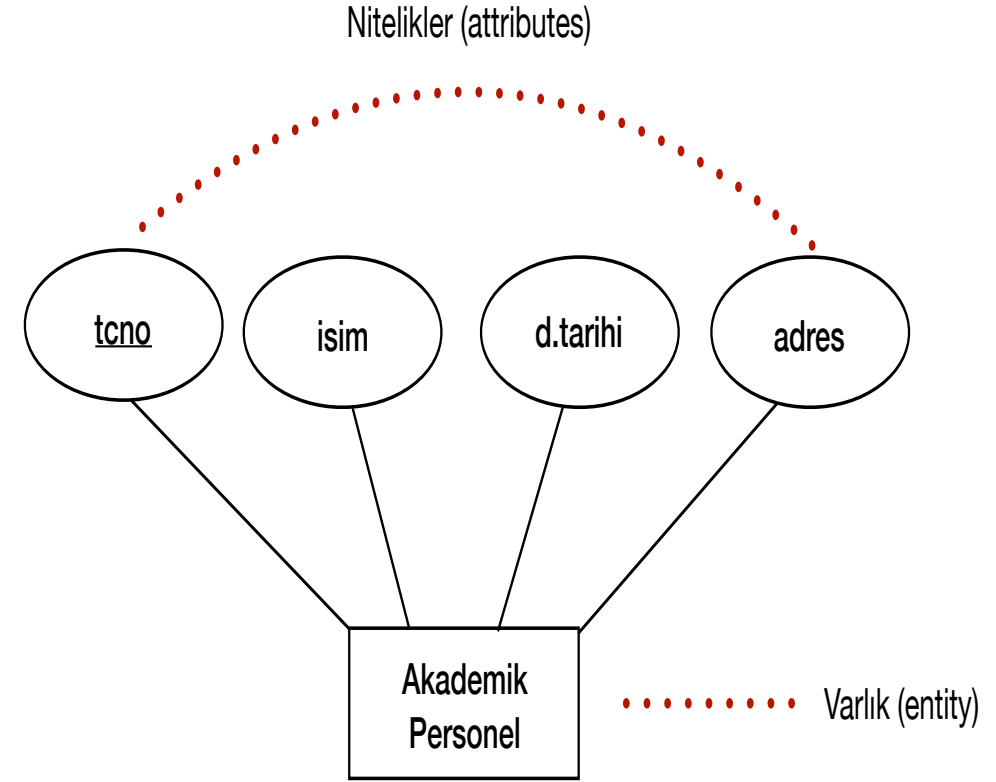
- Bölüm ders açar (offers)
- Öğrenciler derslere kayıt olur (enrolls)
- Prof. dersleri okutur (teaches)
- İdari personel bölüm de çalışır
- Prof. Bölümü yönetir (chairs)

İlişkileri bulunmaktadır



Varlık-İlişki Modeli : nitelikler (attributes)

- Varlık (entity) nitelikler (attributes) ile tanımlanır
- Bir varlık seti içinde bulunan nesnelerin ortak nitelikleri bulunmaktadır
(isim varchar (30), TC No (int), d.tarihi, address)
- Varlık seti içersinde her bir varlığı diğer varlıktan ayırt edici bir nitelik bulunur: **anahtar (key)**
- Birden fazla anahtar adayı varsa bir tanesi primary key olarak seçilir

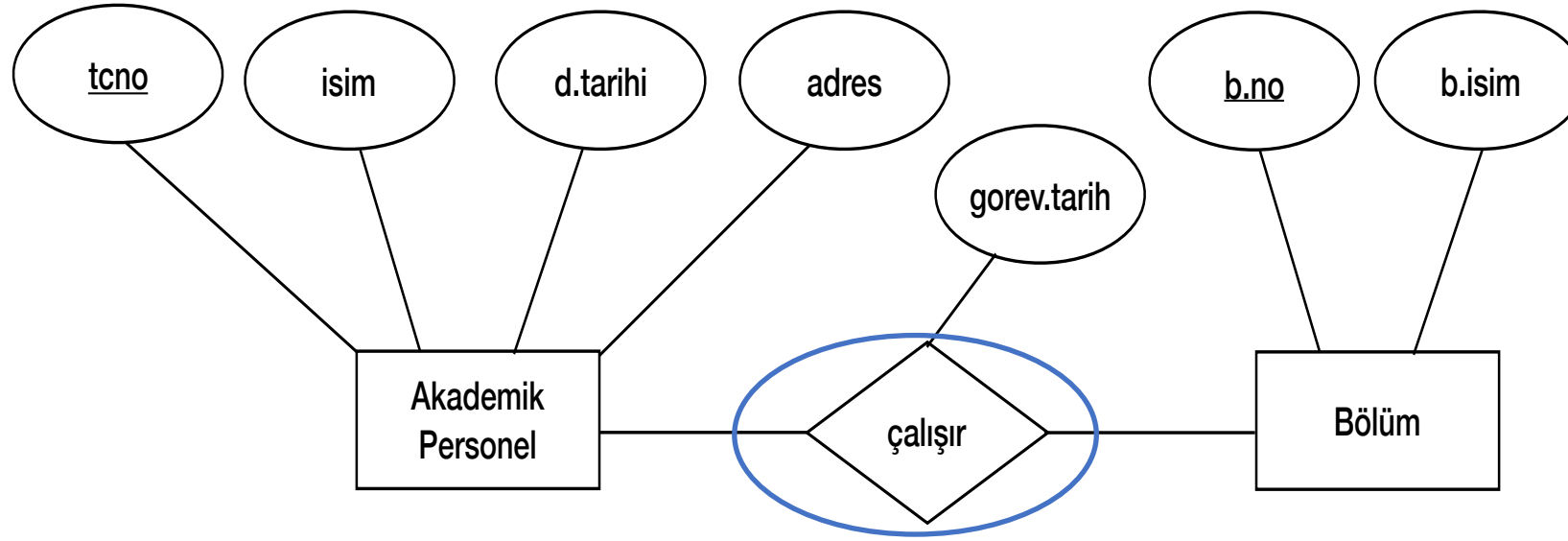


Varlık-İlişki Modeli : ilişki seti (relationship set)

- İki veya daha fazla varlığın bağlantılı olmasına **ilişki** (*relationship*) denir
- Benzer ilişkiler **ilişki seti** (relationship set) içerisinde tanımlanır
- (e_1, e_2, \dots, e_n) varlıkları farklı varlık setleri içerisinde yer alır.
- Örneğin, e_3 , ve e_9 , akademik personeldir. E_m varlık seti içinde yer alabilir.
- dolayısıyla $e_3 \in E_m, e_9 \in E_m$ biçiminde ifade edilir.

$$\{ (e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n \}$$

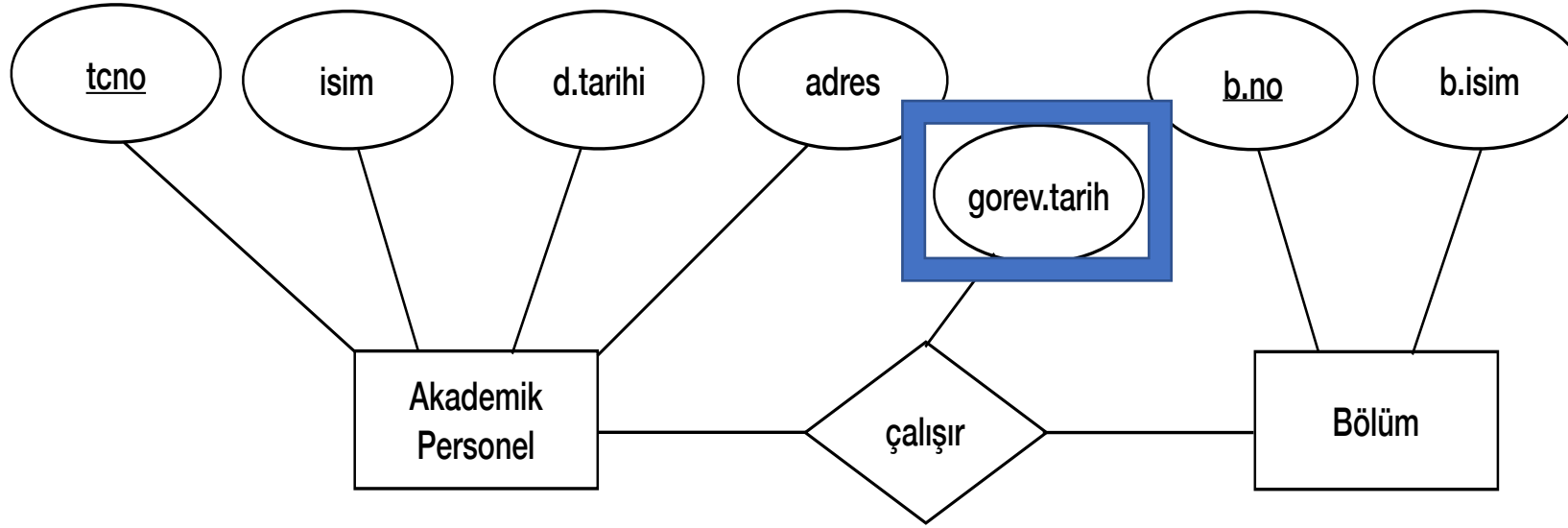
Varlık-İlişki Modeli : ilişki seti (relationship set)



Akademik Personel varlık seti ile Bölüm varlık seti arasında

çalışır ilişkisi tanımlanmıştır.

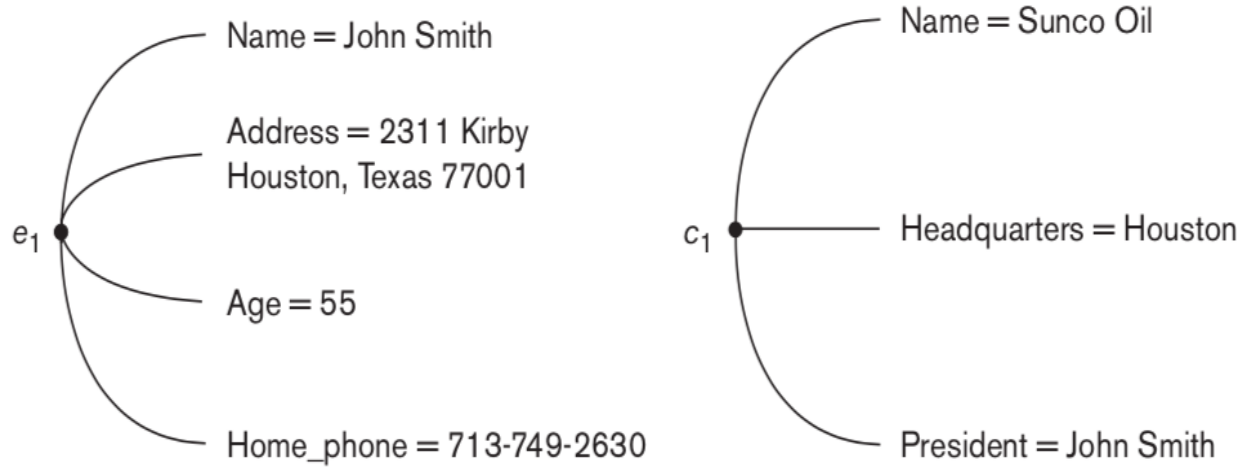
Varlık-İlişki Modeli : ilişkilerin nitelikleri



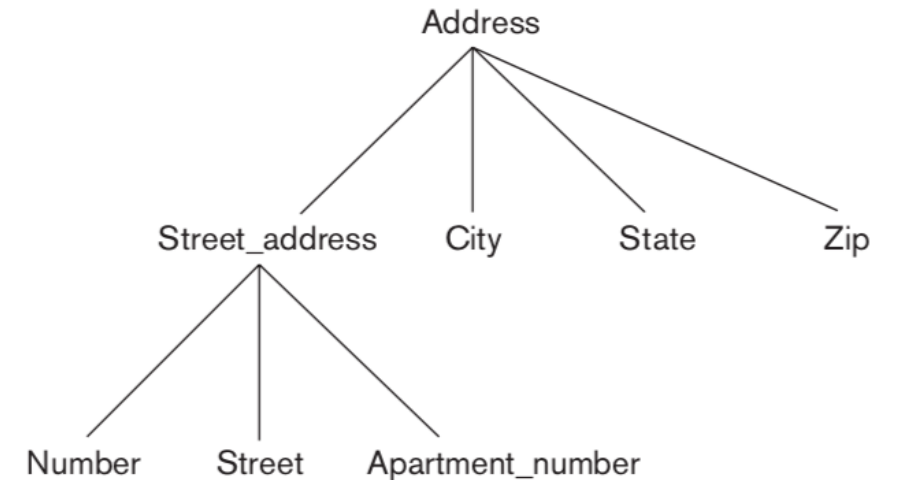
- İlişkilerin ilişki setleri içerisinde *açıklayıcı nitelikleri* (*descriptive attributes*) bulunabilir. Örneğin çalışır ilişkisinin **gorev.tarih** niteliği bulunmaktadır
- Varlıklar arasındaki ilişki tek (**unique**) olarak tanımlanmalıdır. Akademik personel ve bölüm varlıklarının ilişkisi içerisinde bulunan gorev.tarih niteliği belirtilen ilişkiyi unique(tek) olmasını sağlar.

Varlık-İlişki Modeli : Nitelik Çeşitleri

Bölünemeyen (simple & atomic)

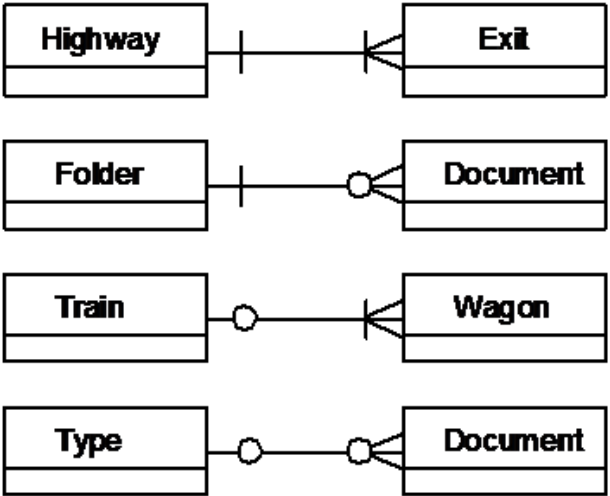
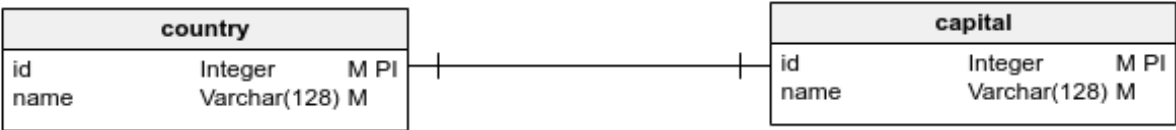
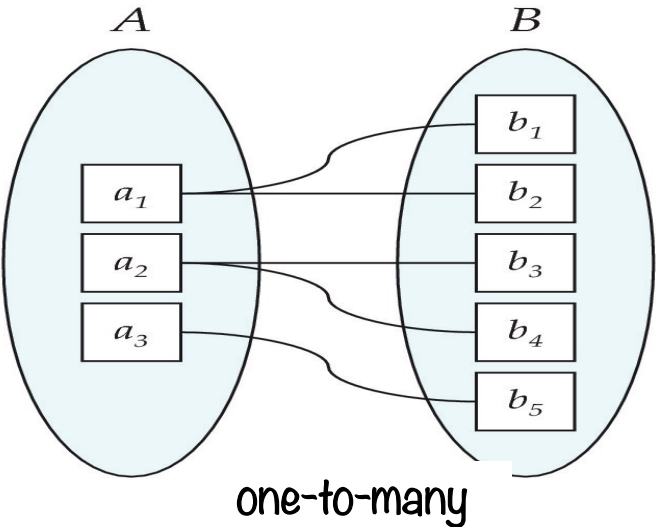
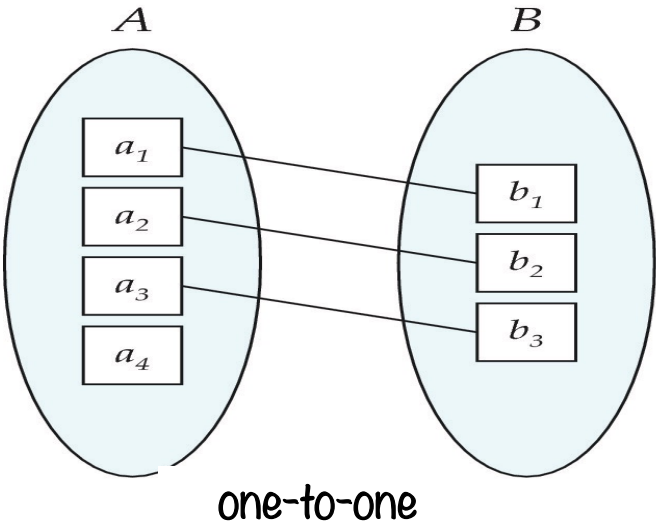


Birleştirilmiş (composite) nitelikler



- Tek değer (single value): yaş
- Birden çok değerli (multivalued):
 - Color (Kırmızı, Turuncu, Sarı , Yeşil ,)
 - Mezuniyet (Lisans, MSs, PhD, Post doc)
- Kayıtlı (Stored): Doğum Tarihi
- Elde Edilen Değer (derived): yaş (bu günün tarihi – dogum tarihi)

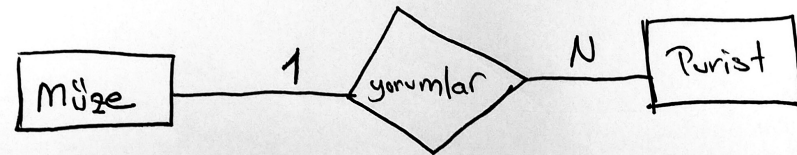
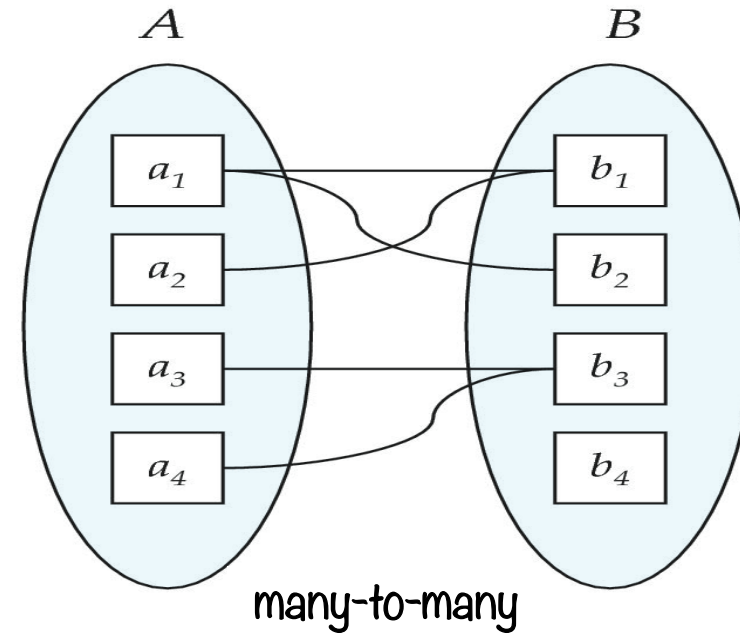
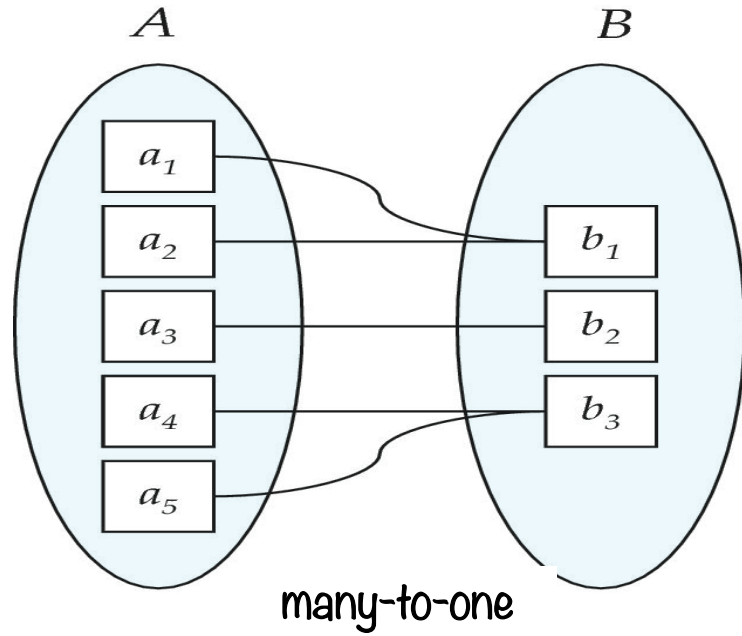
ER Model: one-to-one one-to-many



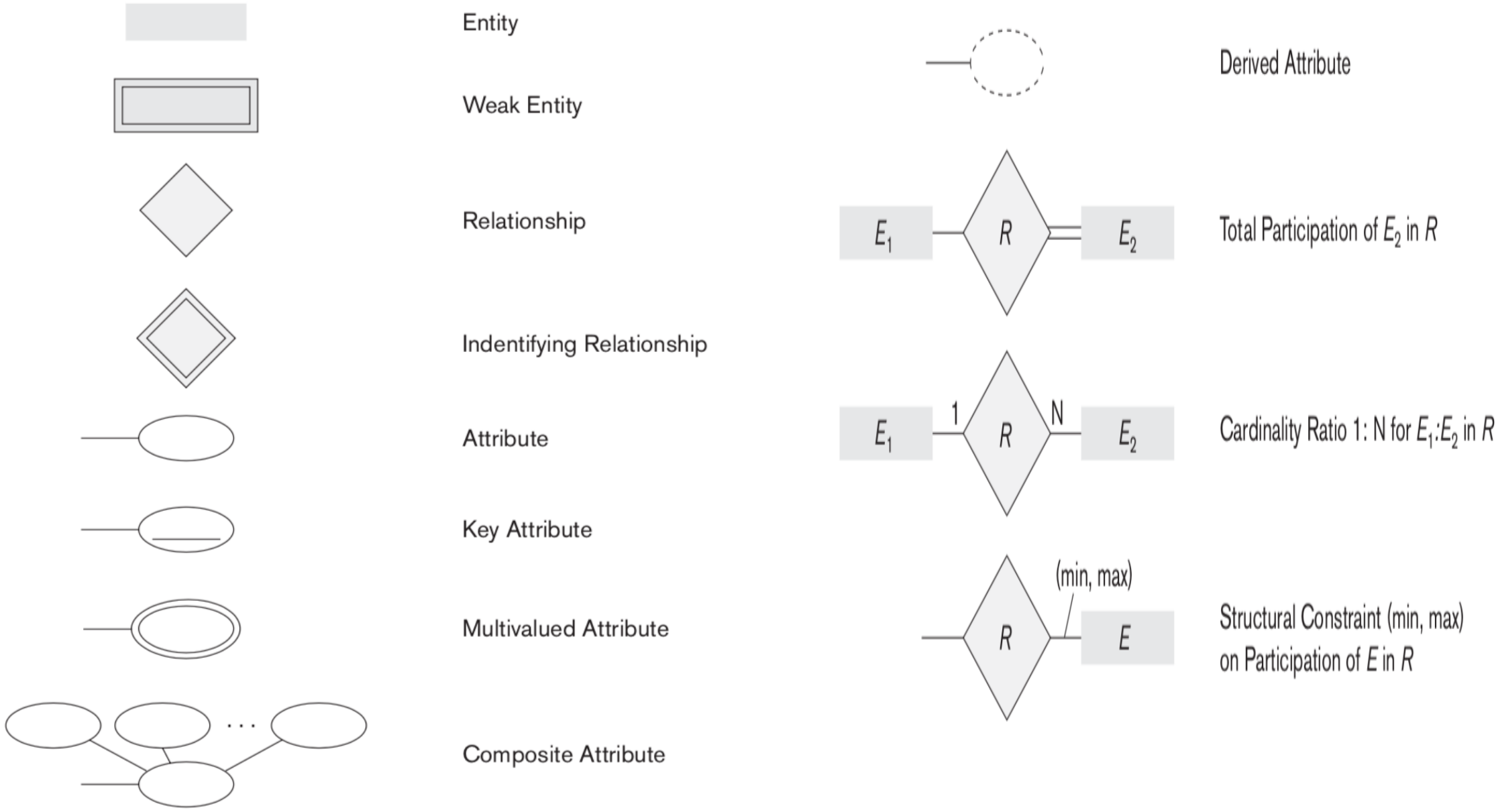
<https://vertabelo.com/blog/one-to-one-relationship-in-database/>

http://www.risetobloome.com/Page_1_S.aspx?ITEM=616

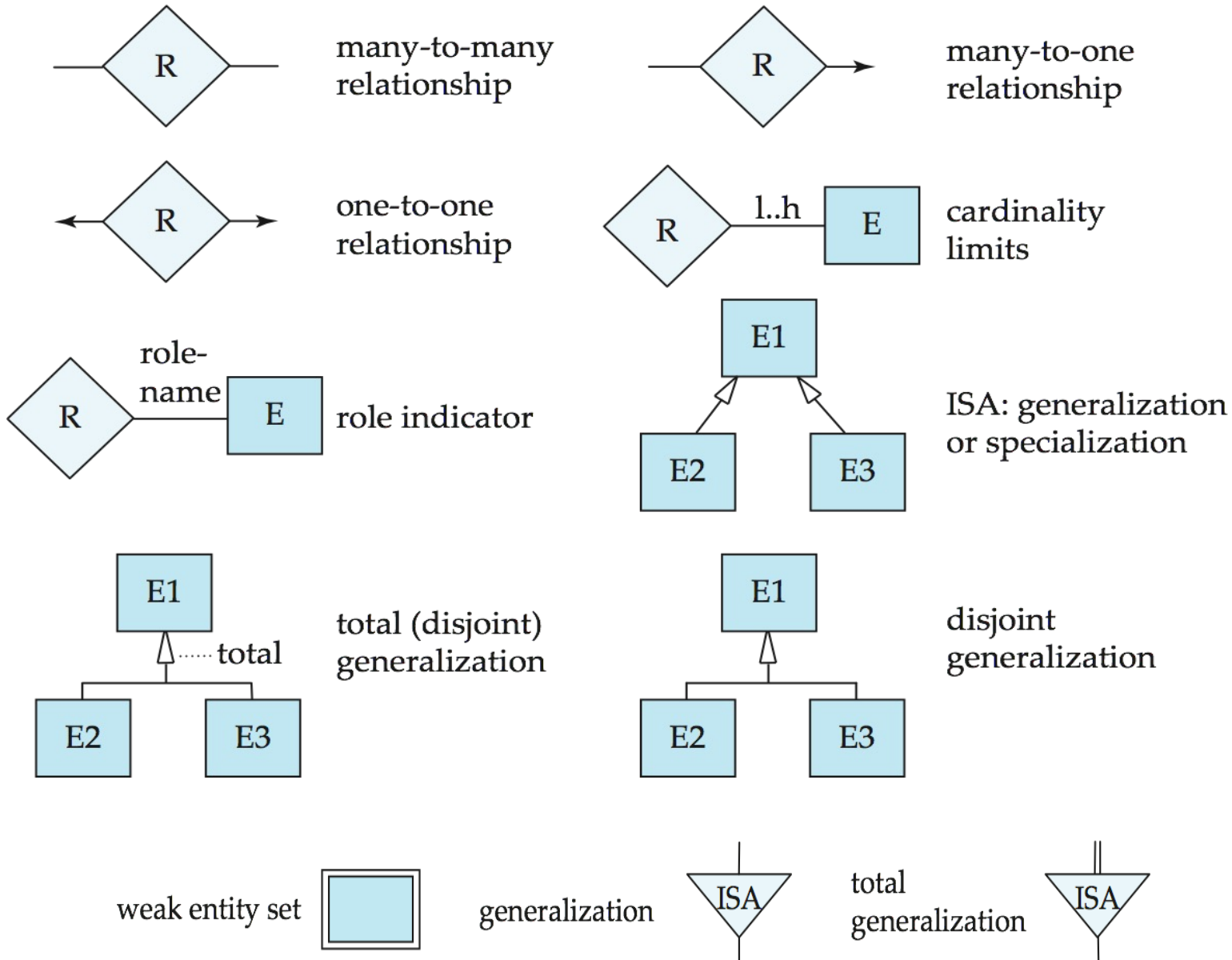
ER Modeli: many-to-one many-to-many



ER Modelinde Kullanılan Semboller

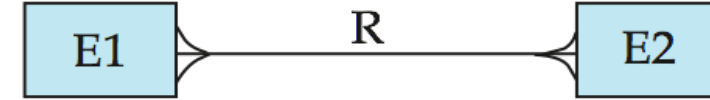
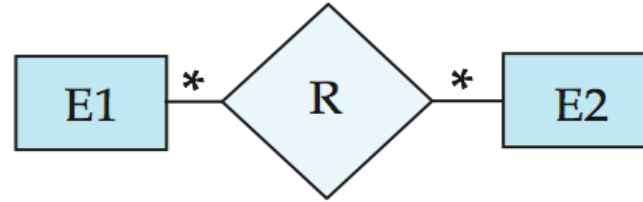


ER Modelinde Kullanılan Semboller

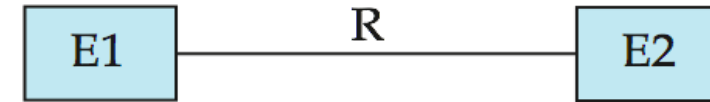
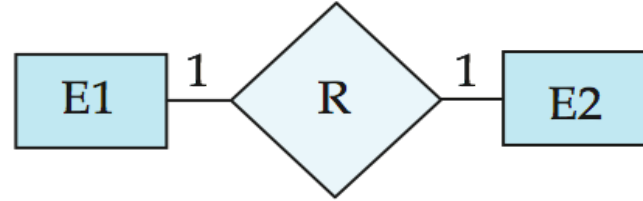


ER Modelinde Kullanılan Semboller

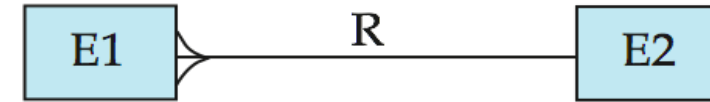
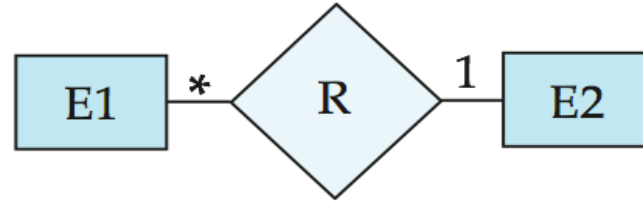
many-to-many
relationship



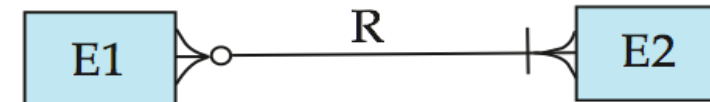
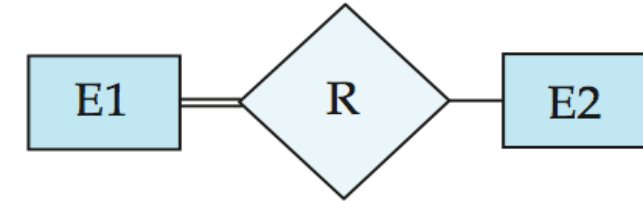
one-to-one
relationship



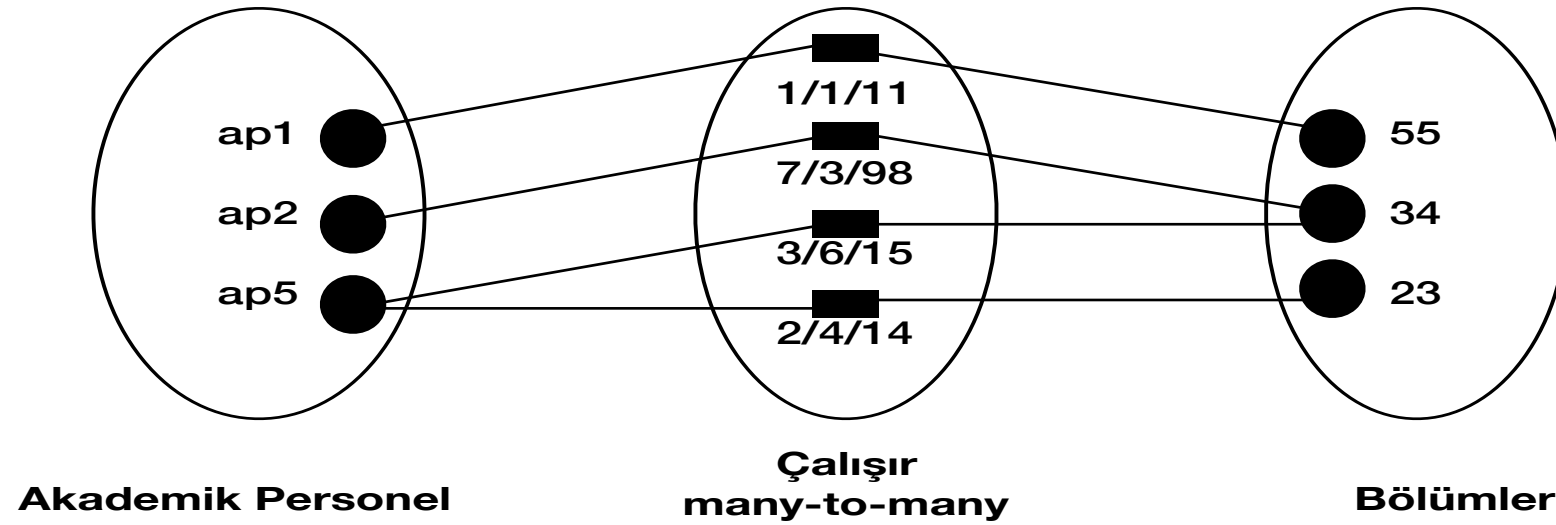
many-to-one
relationship



participation
in R: total (E1)
and partial (E2)

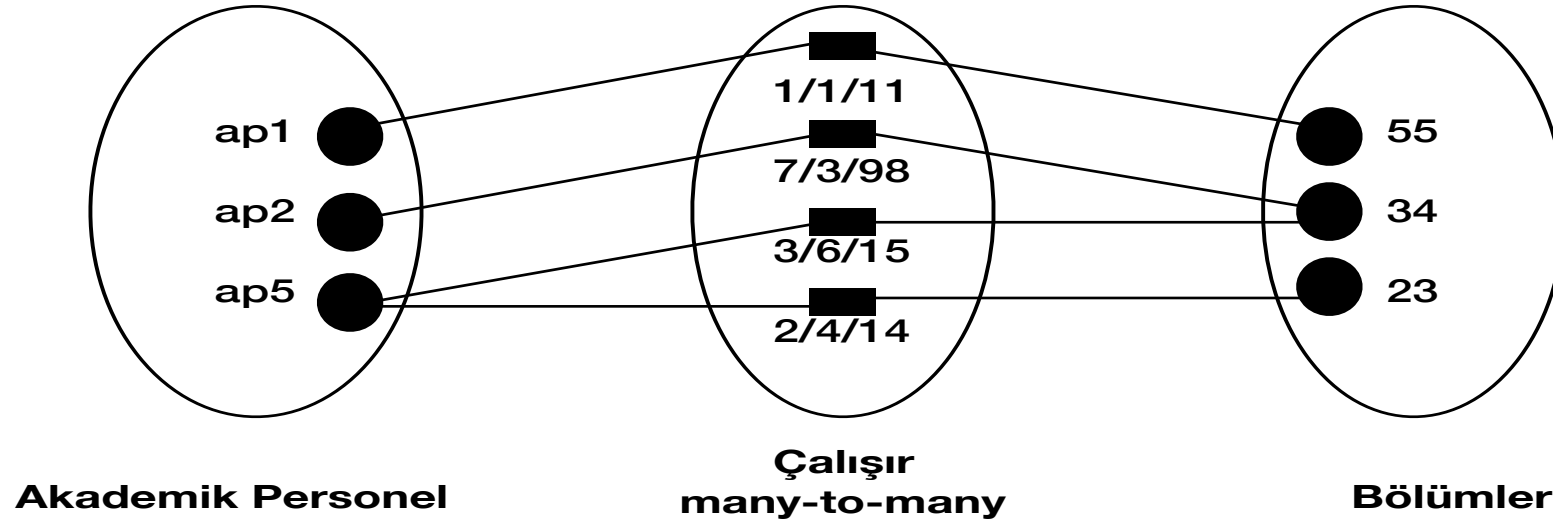


ER Modeli: örnek set (instance)



İlişki setinin bir örneği (*instance*) de bir ilişki setidir.

ER Modeli: örnek set (instance)



Akademik personel ve bölümler arasında **many-to-many** ilişkisi bulunur.

- Bir akademik personel birden fazla bölümde çalışabilir (ap5).
- Bir bölümde birden fazla akademik personel çalışabilir. (34 nolu bölüm)

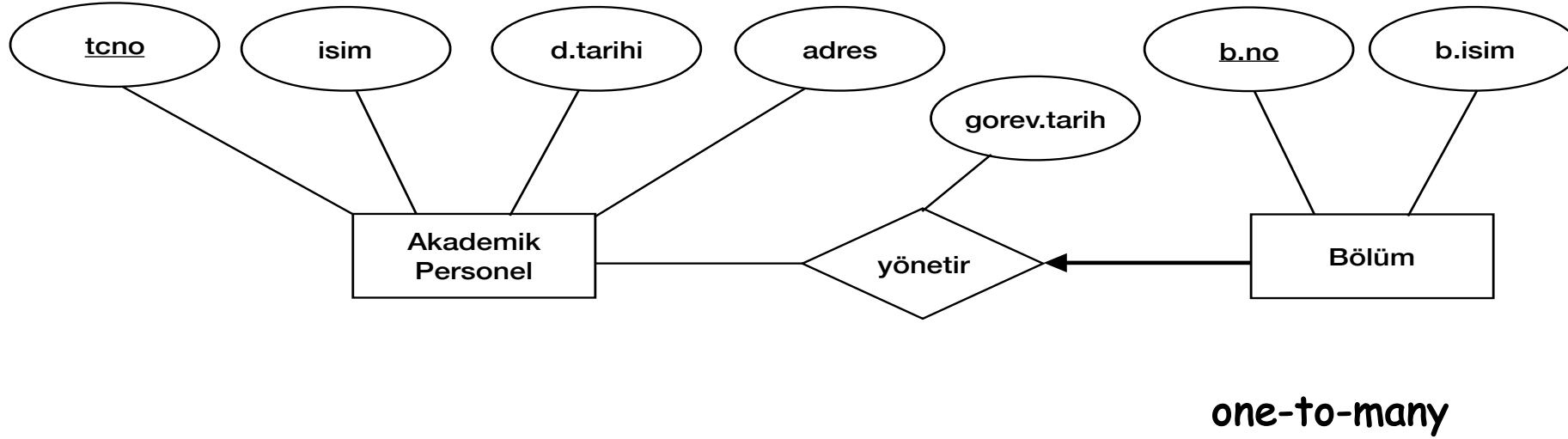
ER Modeli: kısıtlamalar (key constraints)

ER modelinde

bazı semboller (ok, kalın çizgi) kullanılarak

anahtar kısıtlamalar (key constraints) tanımlanmaktadır.

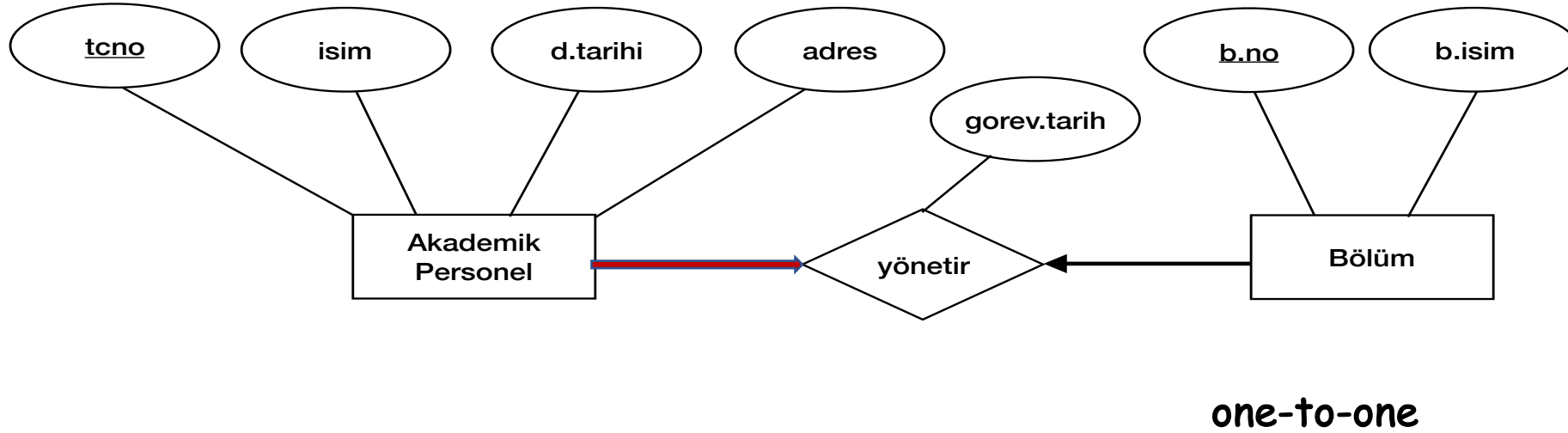
ER Modeli: kısıtlamalar (key constraints)



Bir akademik personel birden fazla bölümü yönetebilir fakat *aynı konum birden fazla kişi tarafından paylaşılamaz.*

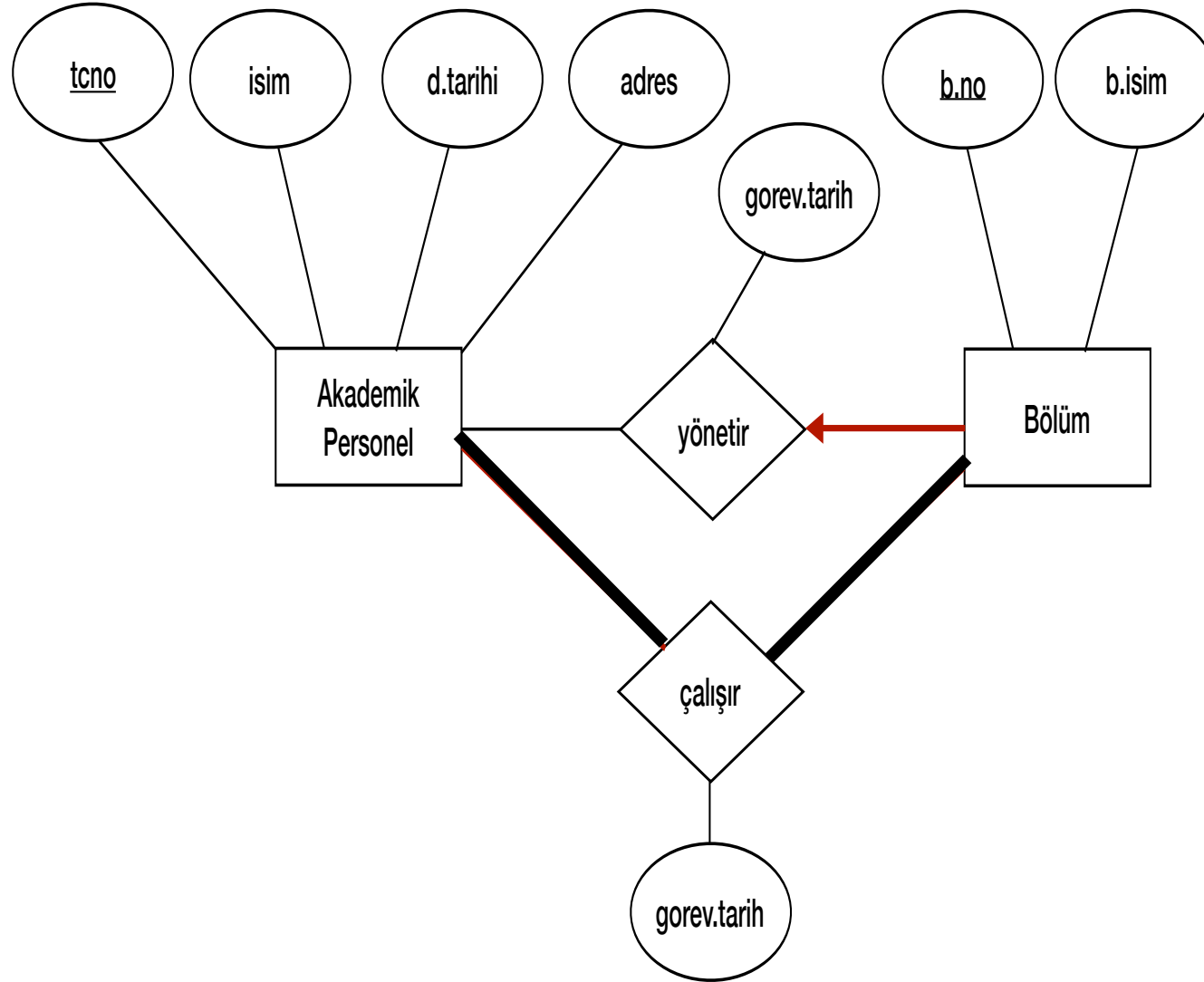
Mühendislik fakültesinin **sadece bir dekanı** olur fakat aynı kişi başka bir fakültede de dekanlık yapabilir.

ER Modeli: kısıtlamalar (key constraints)



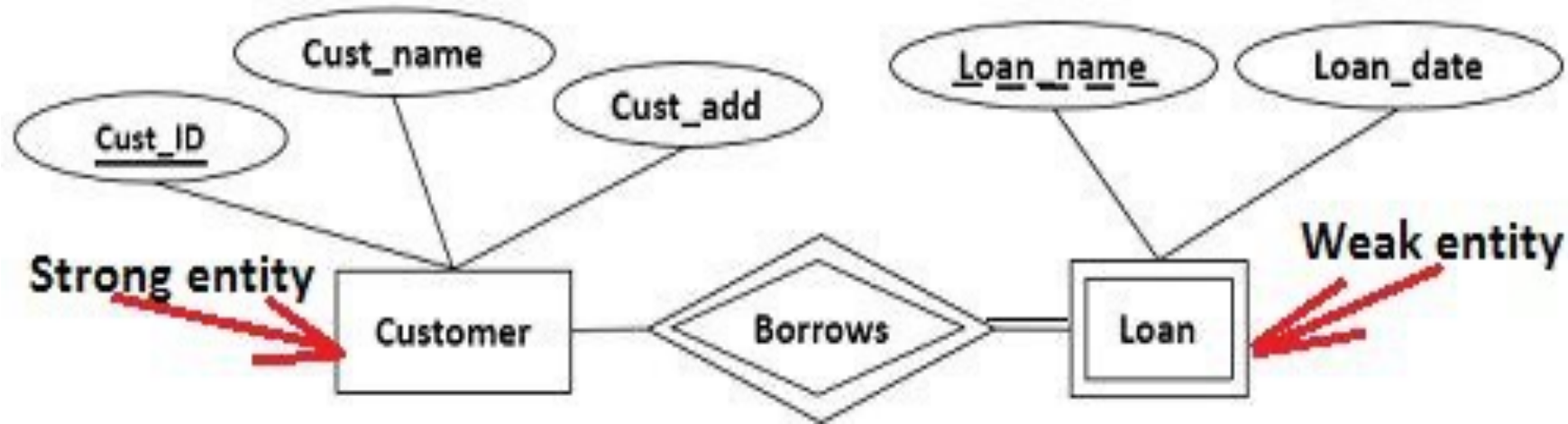
- Birden fazla bölümün yönetilmesi kısıtlanabilir
- Bir bölümün sadece bir akademik personel tarafından yönetilmesi isteniyorsa (**one-to-one**)
- Akademik personelden yönetir ilişkisine doğru bir ok ile bu kısıtlama eklenebilir

ER Modeli: toplam katılım (total participation)



Akademik personelin hepsi bölümde çalışır ilişkisinde yer alırsa akademik personel ile çalışır arasında **toplam katılım** (*total participation*) özelliği bulunduğu kalın bir çizgi ile gösterilir.

ER Modeli: zayıf varlık (weak entity)

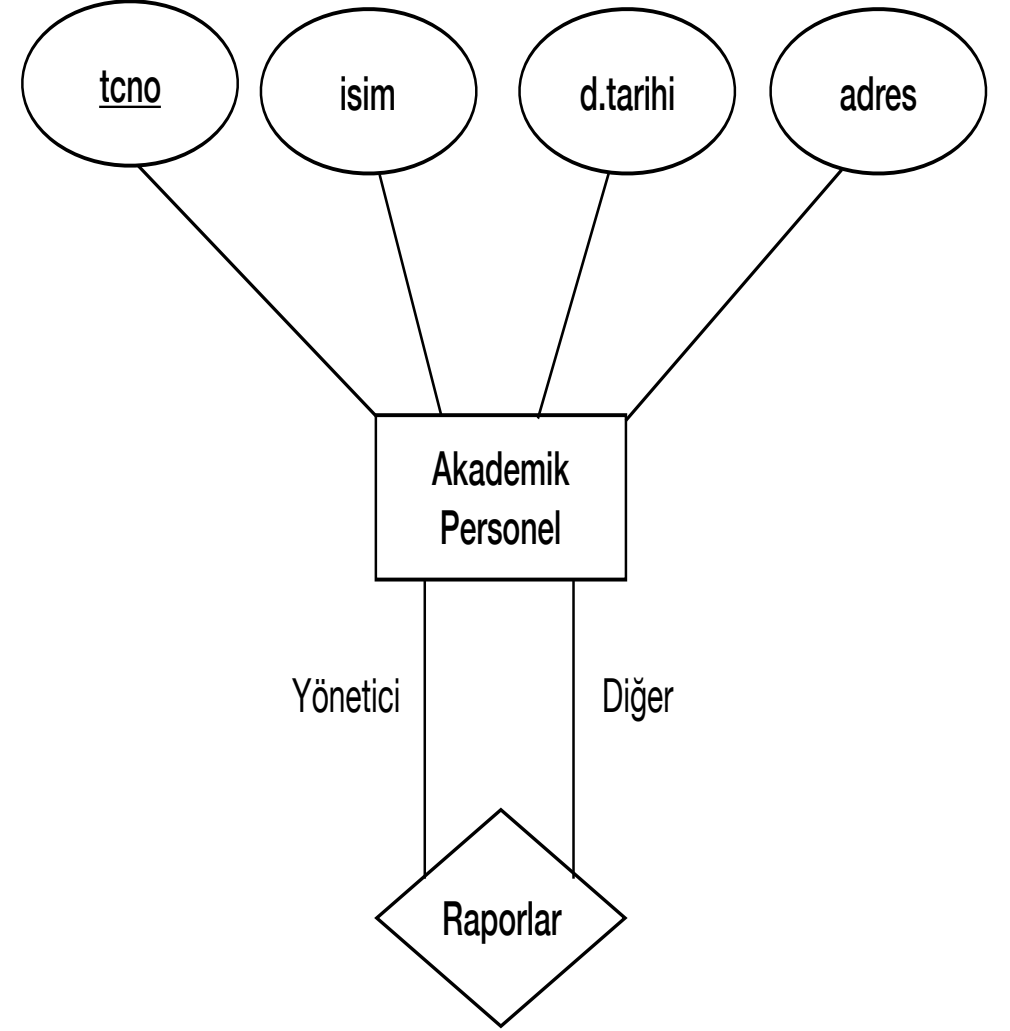


<https://techdifferences.com/difference-between-strong-and-weak-entity.html>

- Bir varlık setinde bulunan varlıklar geçici olabilen **zayıf varlıklar** (weak entities) ile ilişkilendirilebilir
- akademik personel varlık kümesine sorumludur ilişkisi ile bakmakla yükümlü olduğu kişiler de zayıf varlık olarak eklenebilir.

ER Modeli: ilişki seviyeleri

- Bir varlık setinde ilişki sorumluluğu farklı olan varlıklar varsa **sorumluluk seviyesi** ilişkide belirtilir.
- Yönetici ve diğer (yönetilen) personel ile raporlar arasında bulunan farklı ilişki seviyesi belirlenebilir.



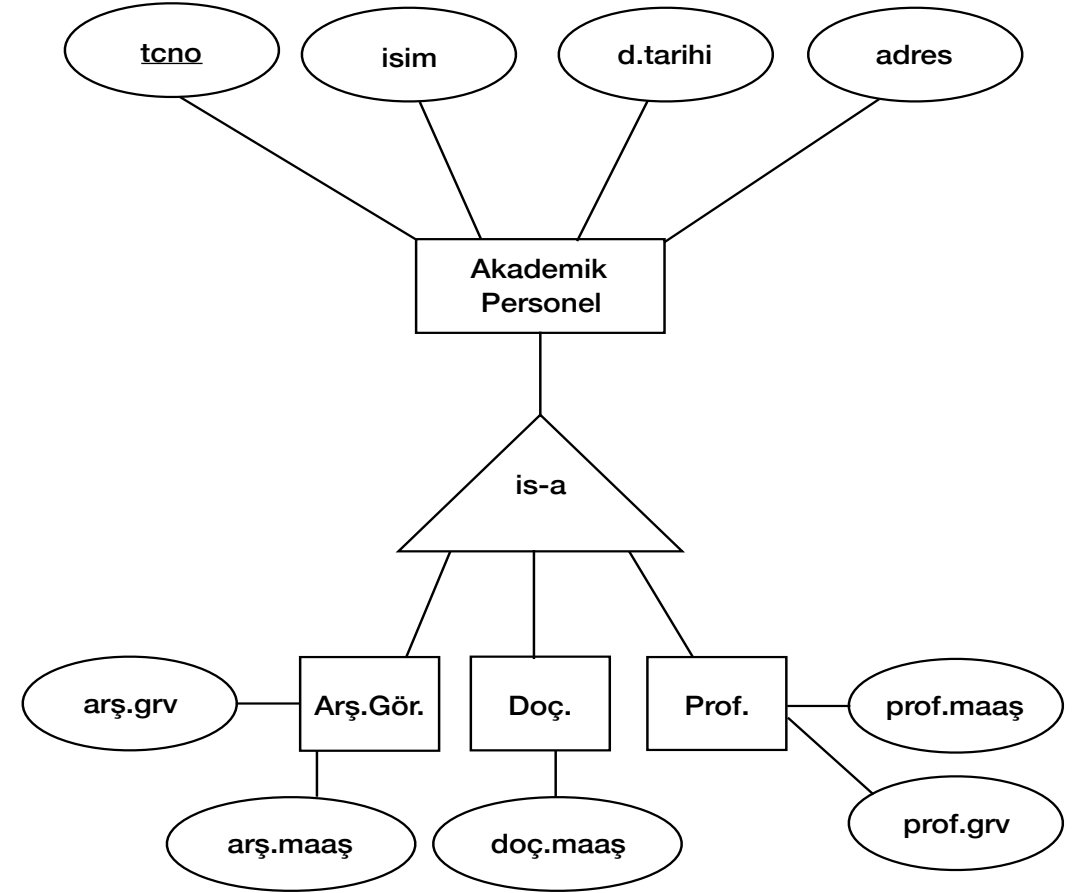
Sınıf Hiyerarşileri (Class Hierarchies)

- Bir varlık setinin kendi içerisinde **alt sınıflar (sub class)** bulunabilir
- Akademik personel de kendi içerisinde alt sınıfları bulunur
 - Uzman , Okutman
 - Araştırma Görevlisi
 - Dr.Öğr. Üyesi
 - Doçent
 - Professor

Sınıf Hiyerarşileri: kalıtım (inheritance)

Akademik Personel varlık setinde bulunan alt sınıflar, akademik personel varlık setinde tanımlı olan tüm nitelikleri kalıtsal olarak devralır (*inheritance*)

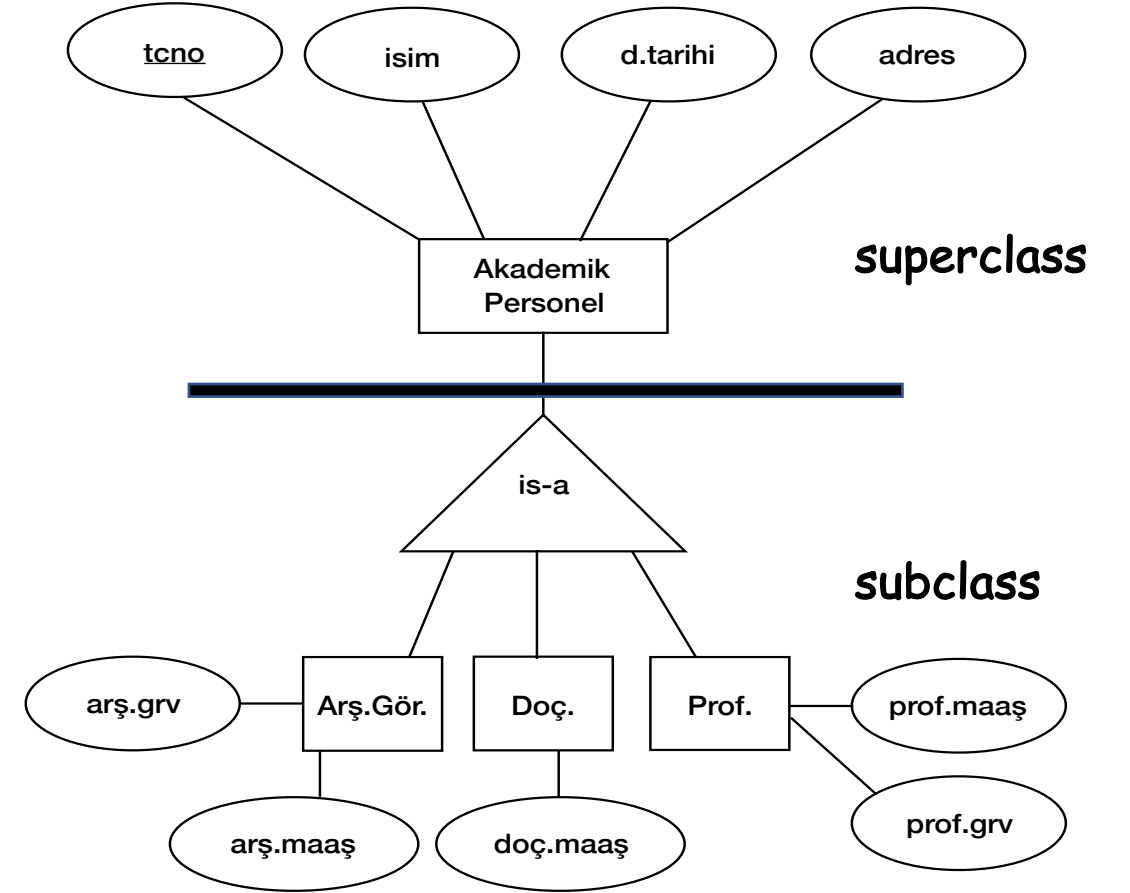
Akademik Personel varlık setinde tanımlı olmayan **yeni nitelikler** ise alt sınıf (subclass) oluşturularak eklenebilir.



Sınıf Hiyerarşileri: Kalıtım (inheritance)

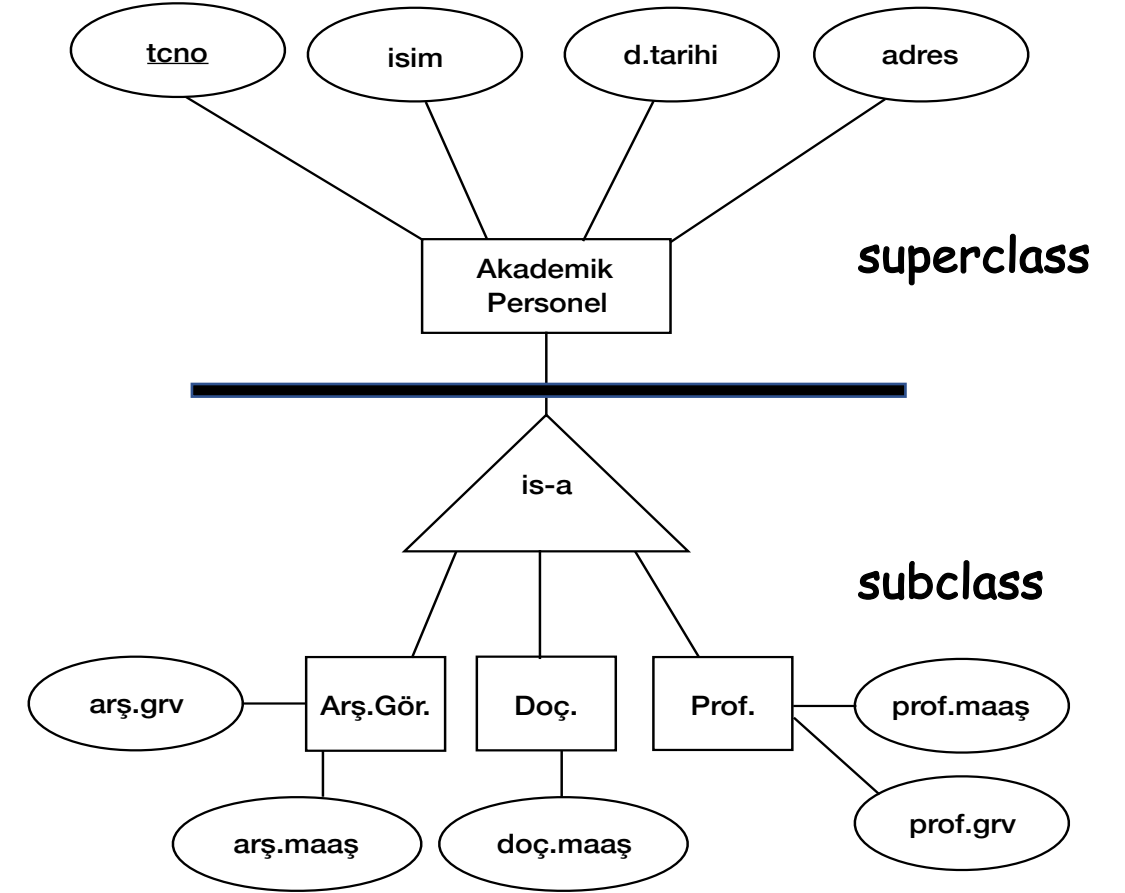
Hiyerarşik sınıf tanımlamalarında

1. üst sınıf (superclass) tanımlanır
2. üst sınıfı tanımlayan nitelikler belirlenir
3. alt sınıf (subclass) lar tanımlanır
4. alt sınıfa ait nitelikler belirlenir



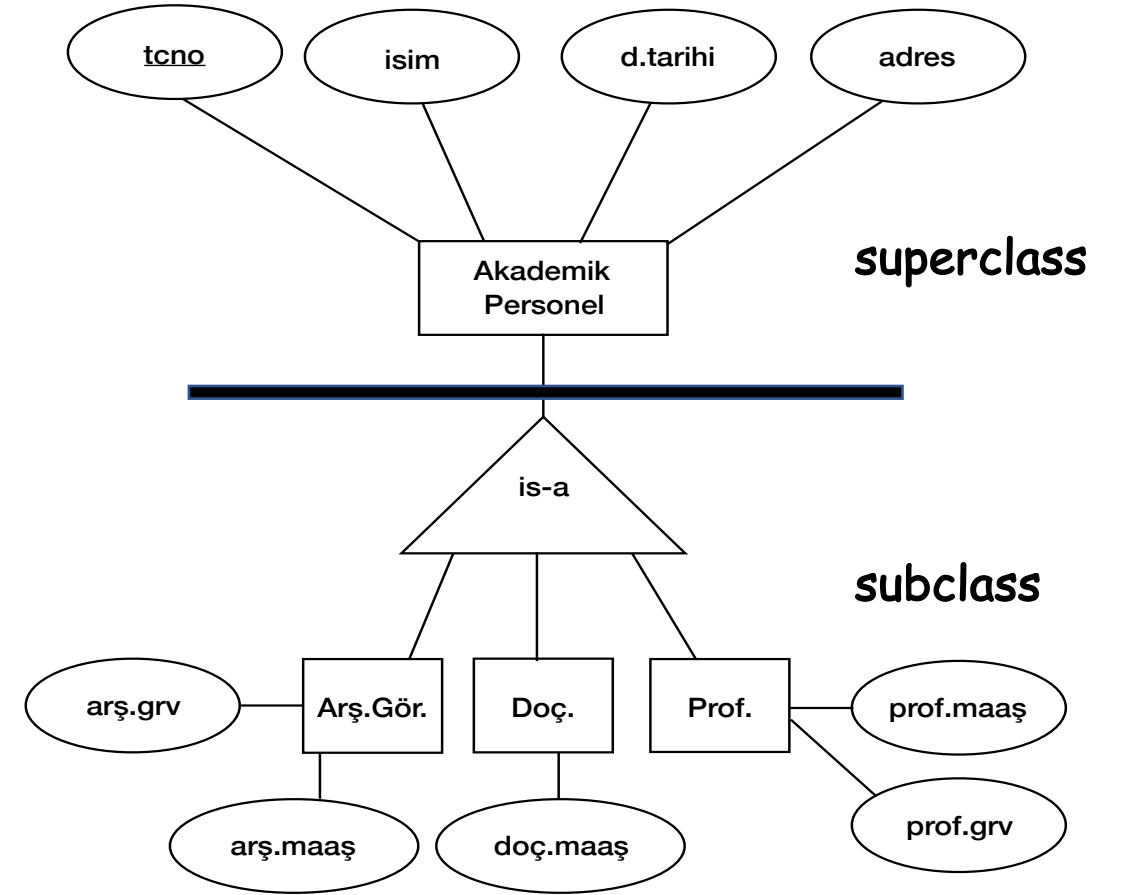
Sınıf Hiyerarşileri: Kalıtım (inheritance): is a ilişkisi

- Alt sınıfta (subclass) bulunan varlığın bir üst seviyedeki (superclass) varlık setinin bir üyesi olduğunu tanımlar.
- Alt sınıfları bulunan varlık setleri is-a ilişkisi ile tanımlanır



Sınıf Hiyerarşileri: kalıtım (inheritance): is a ilişkisi

- “araştırma görevlisi bir akademik personel dir”
- “Doçent bir akademik personel dir”
- “Prof bir akademik personel dir”
 - Bir Prof. akademik personel varlık kümesinin bütün niteliklerini içerir
 - Kendinine ait özel nitelikleri bulunabilir.
 - Diğer akademik personelden farklı olan sorumlulukları bulunur.

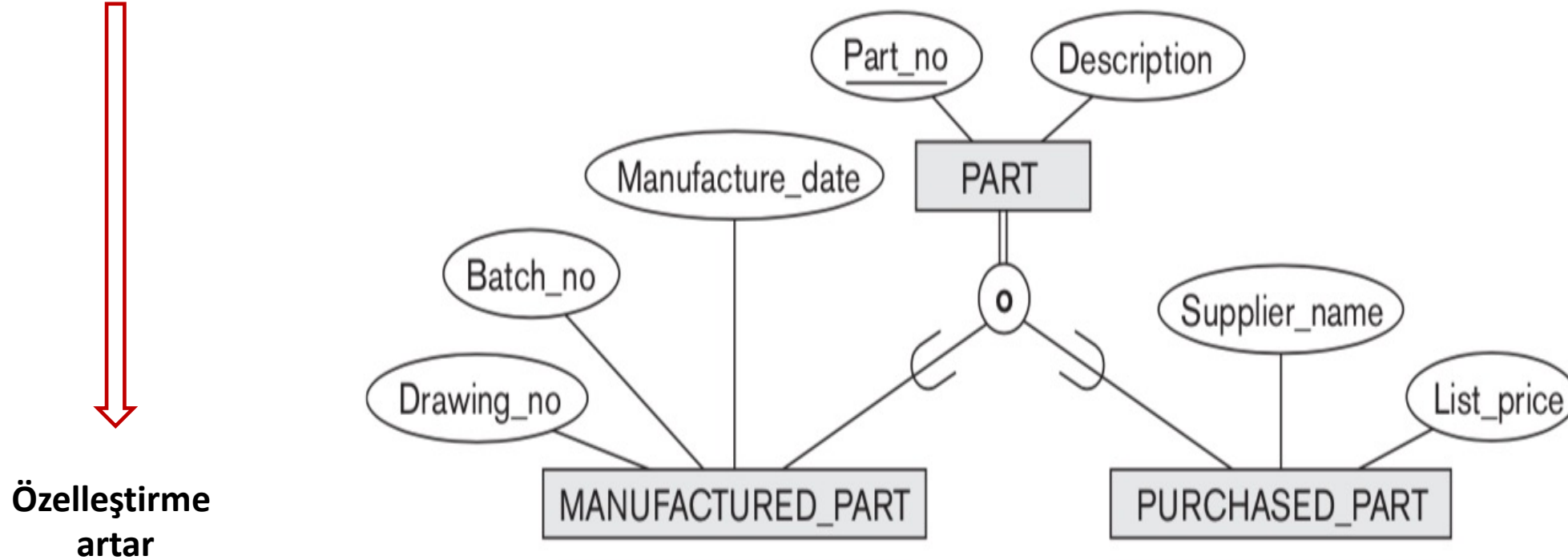


Sınıf Hiyerarşileri (Class Hierarchies)

Sınıf hiyerarşik yapısı

özelleştirme (specialization) ve genelleştirme (generalization)
kavramlarını sağlar.

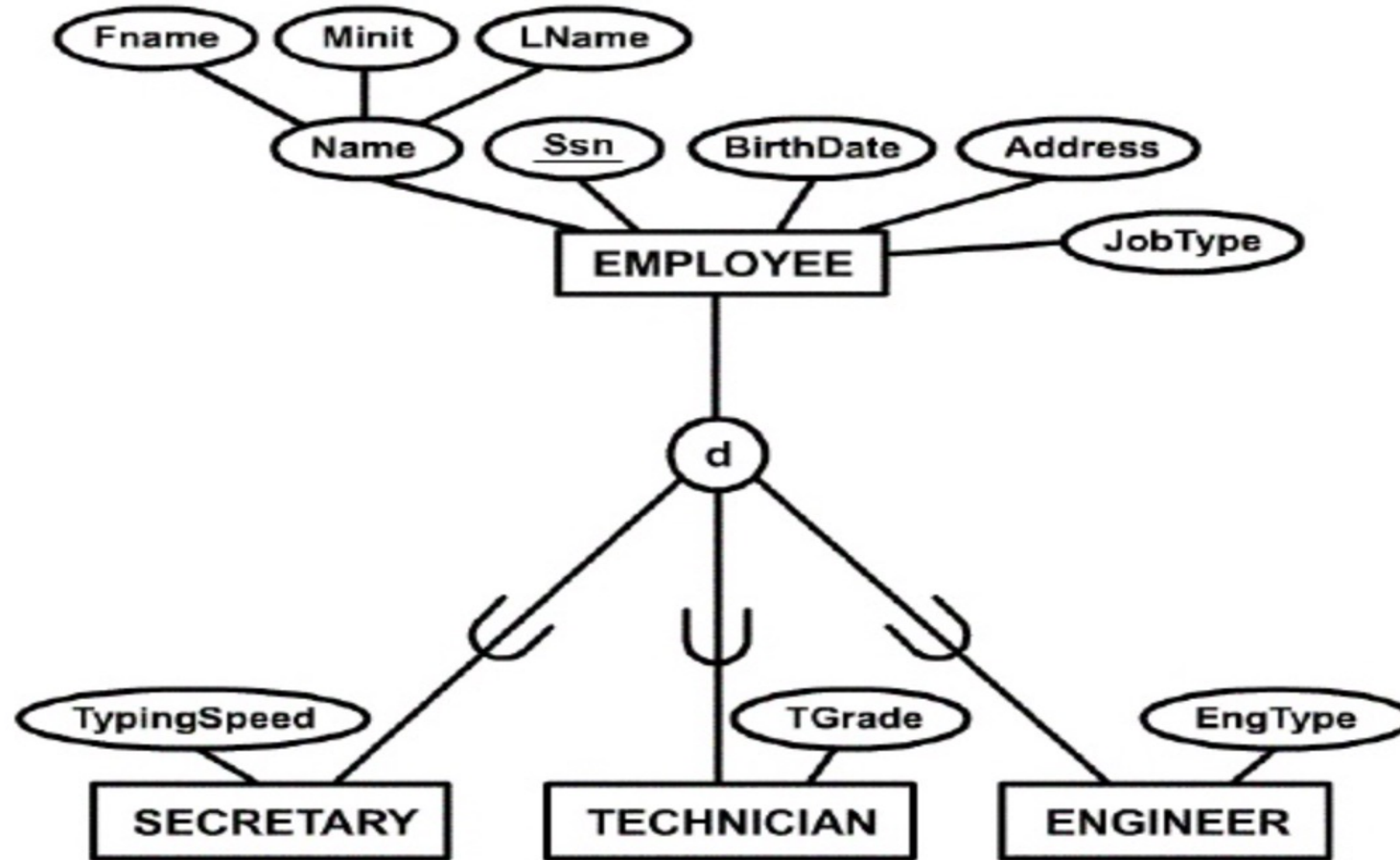
Sınıf Hiyerarşileri: özelleştirme (specialization)



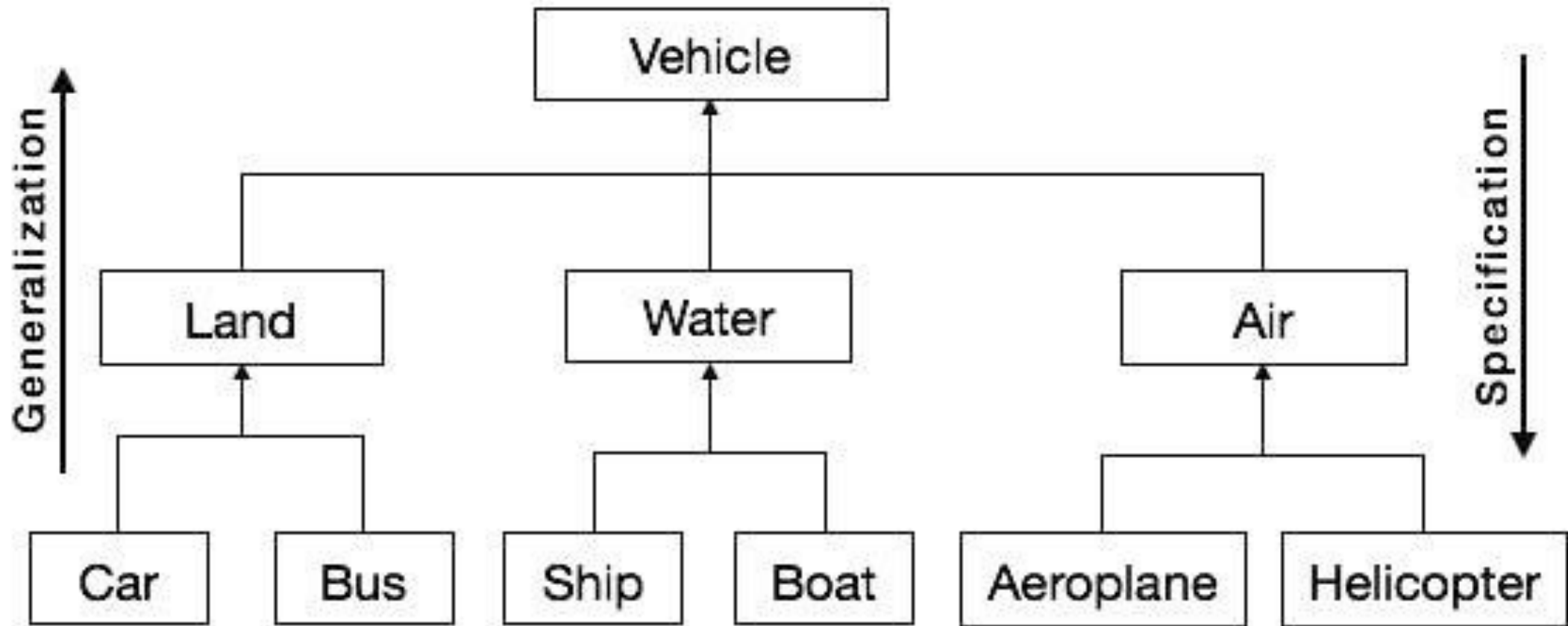
Özelleştirme (specialization) üst sınıftan devralınan niteliklere ek olarak sadece alt sınıfta bulunan varlıklara ait yeni niteliklerin tanımlanmasına imkan sağlar.

Sınıf Hiyerarşileri: genelleştirme (generalization)

genelleştirme
artar



Sınıf Hiyerarşileri: genelleştirme (generalization)



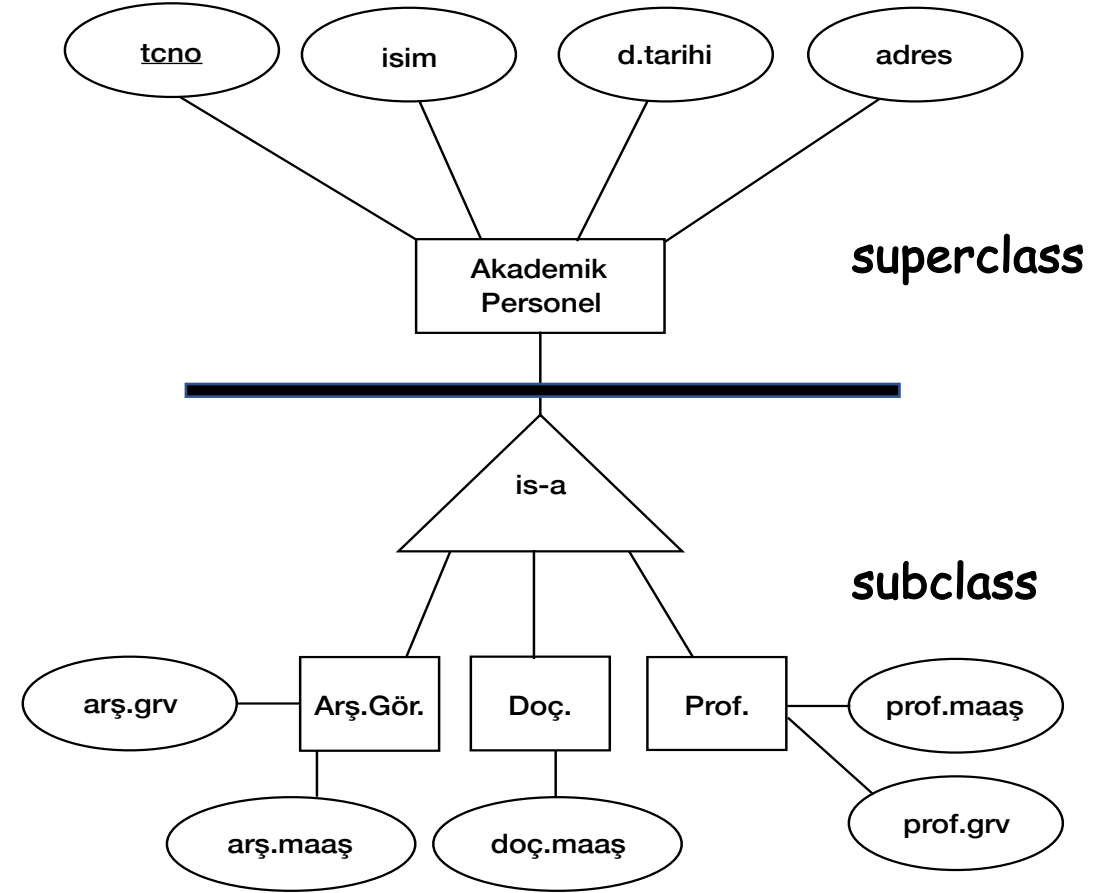
Genelleştirme ortak özellikleri bulunan varlıkların sınıflandırılmasını sağlar.

https://www.tutorialspoint.com/object_oriented_analysis_design/ooad_object_oriented_model.htm

Çakışma (overlap) kısıtlaması

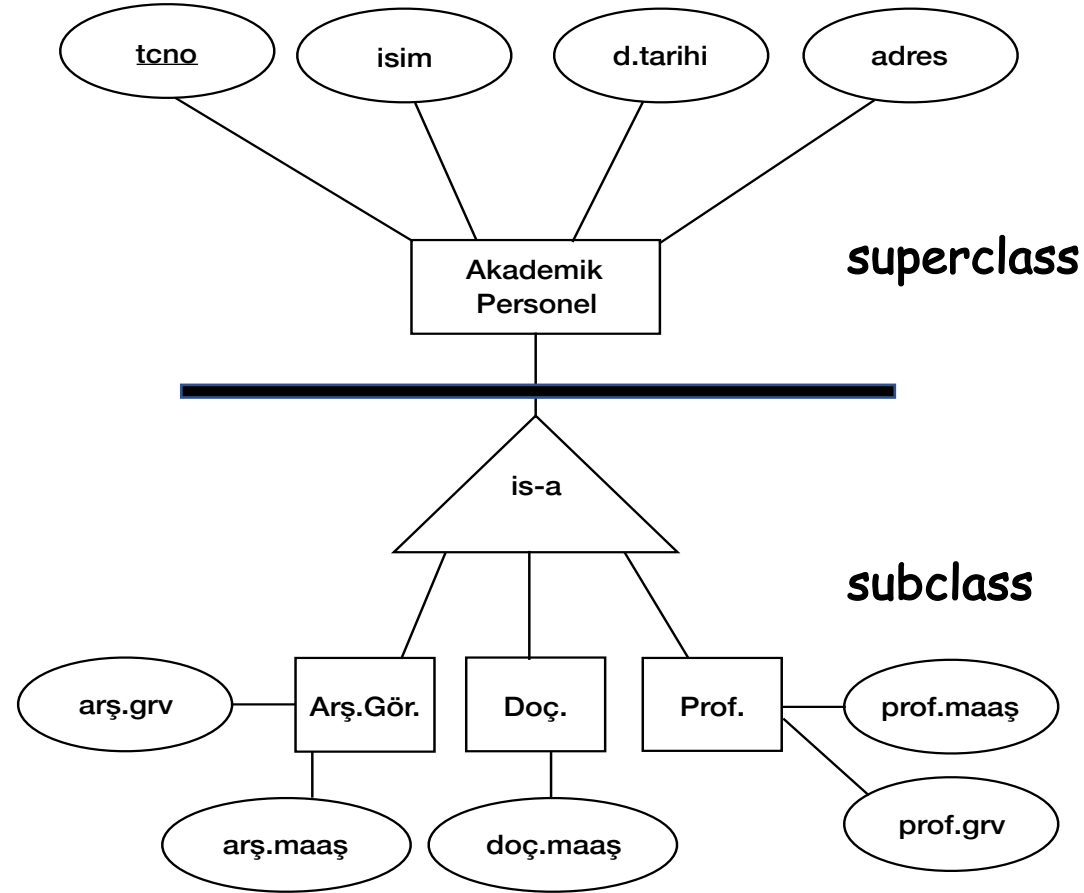
Çakışma (*overlap*) kısıtlaması iki alt sınıfın aynı varlığı içermesini engeller.

Bir akademik personel aynı anda hem Doç hemde Arş.Gör. olamaz



Kapsama (covering) kısıtlaması

Kapsama (*covering*) kısıtlaması alt sınıfllarda bulunan varlıkların toplamının üst sınıfı vermesi prensibidir.



Arş.Gör. , Doç. ve Prof. kapsar Akademik Personel

Dinlediğiniz için
Teşekkürler...
İyi çalışmalar...