

Veritabanı Yönetim Sistemleri



Dr. Öğr. Üyesi Durmuş ÖZDEMİR

e-mail: durmus.ozdemir@dpu.edu.tr

VERİTABANI İLİŞKİ TÜRLERİ

Tablo yapısına ve içerdiği bilgilere göre ilişkiler farklı yapıda olabilir.

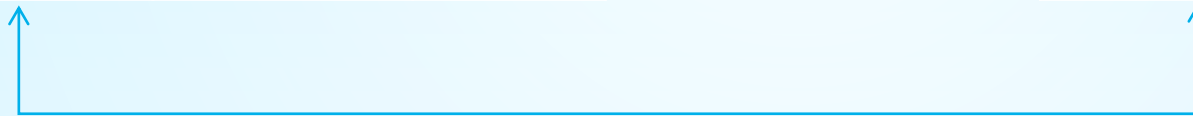
- **Bire-Bir (1-1) İlişkiler:** İlişki kurulan tablolardaki kayıtların aynısından ikincisi olmadığı durumlardaki ilişkidir. Bir tablodaki primary key diğer tabloda sadece bir kayıta karşılık geliyorsa ilişki bire-bir olur.
- **Bire-Çok (1-n) ilişkiler:** İlişki kurulan tablolardaki alanlardan birisindeki primary key alanı diğer tabloda birden fazla kayıta karşılık geliyorsa oluşturulacak ilişki bire-çok olur.
- **Çoğa-Çok (n-m) ilişkiler:** İlişki kurulan tablolardan birindeki birçok kaydın değeri diğer tabloda birden fazla kayda karşılık geliyorsa oluşturulacak ilişki çoğa-çok olur.

Bire-Bir (1-1) İlişkiler:

- Örnekteki gibi;
- 1 kişiye ait sadece 1 şifre olabilir, 1 şifre 1 kişiye ait olabilir.

KimlikBilgileri		
TcNo	Ad	Soyad
121	Ayşe	Berk
243	Kemal	Kurt
982	Musa	Tufan

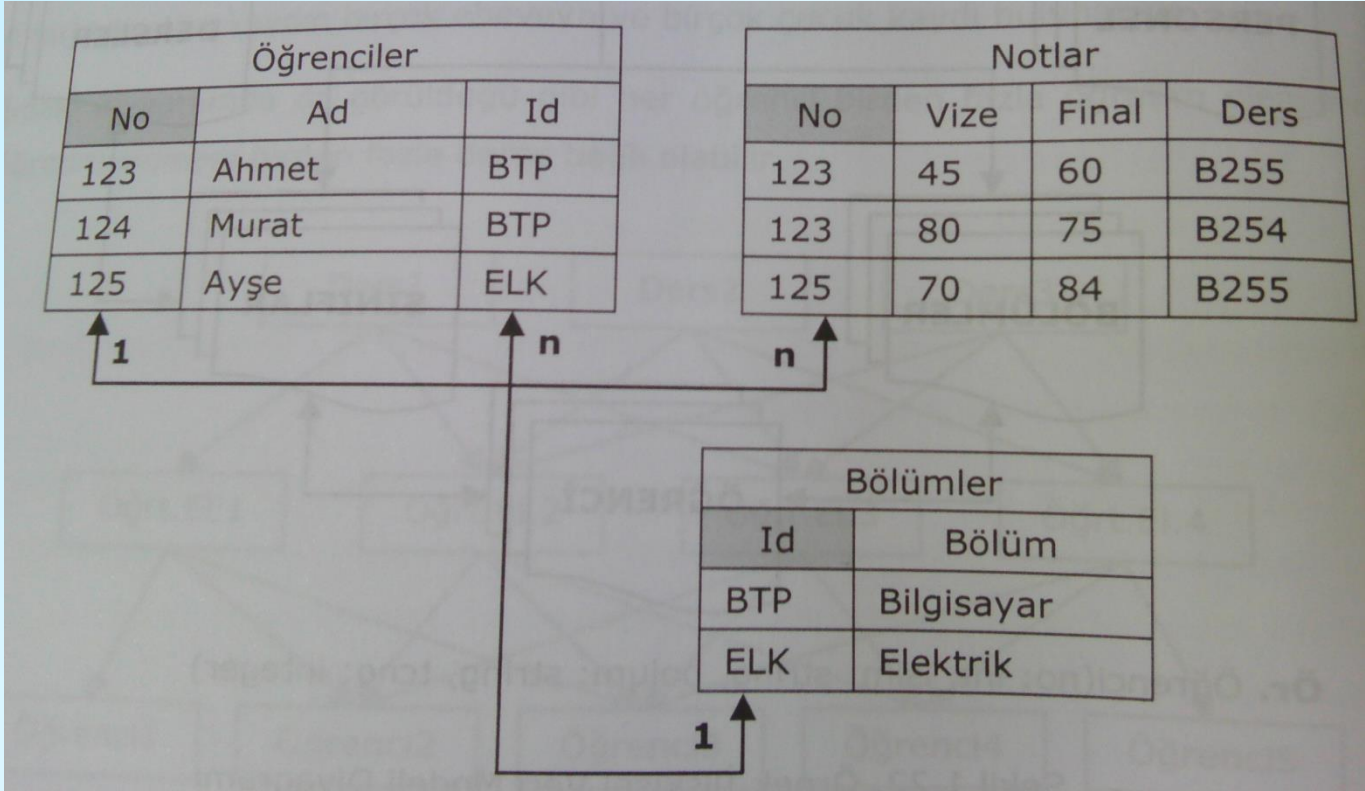
Sifre	
TcNo	Sifre
121	sifre1
243	asd123
982	111111



id	musteri_adi	musteri_soyadi	adres	bilet_id
1	Selin	Demir	İstanbul	1
2	Ahmet	Kara	Ankara	2
3	Selim	Nadir	İstanbul	3

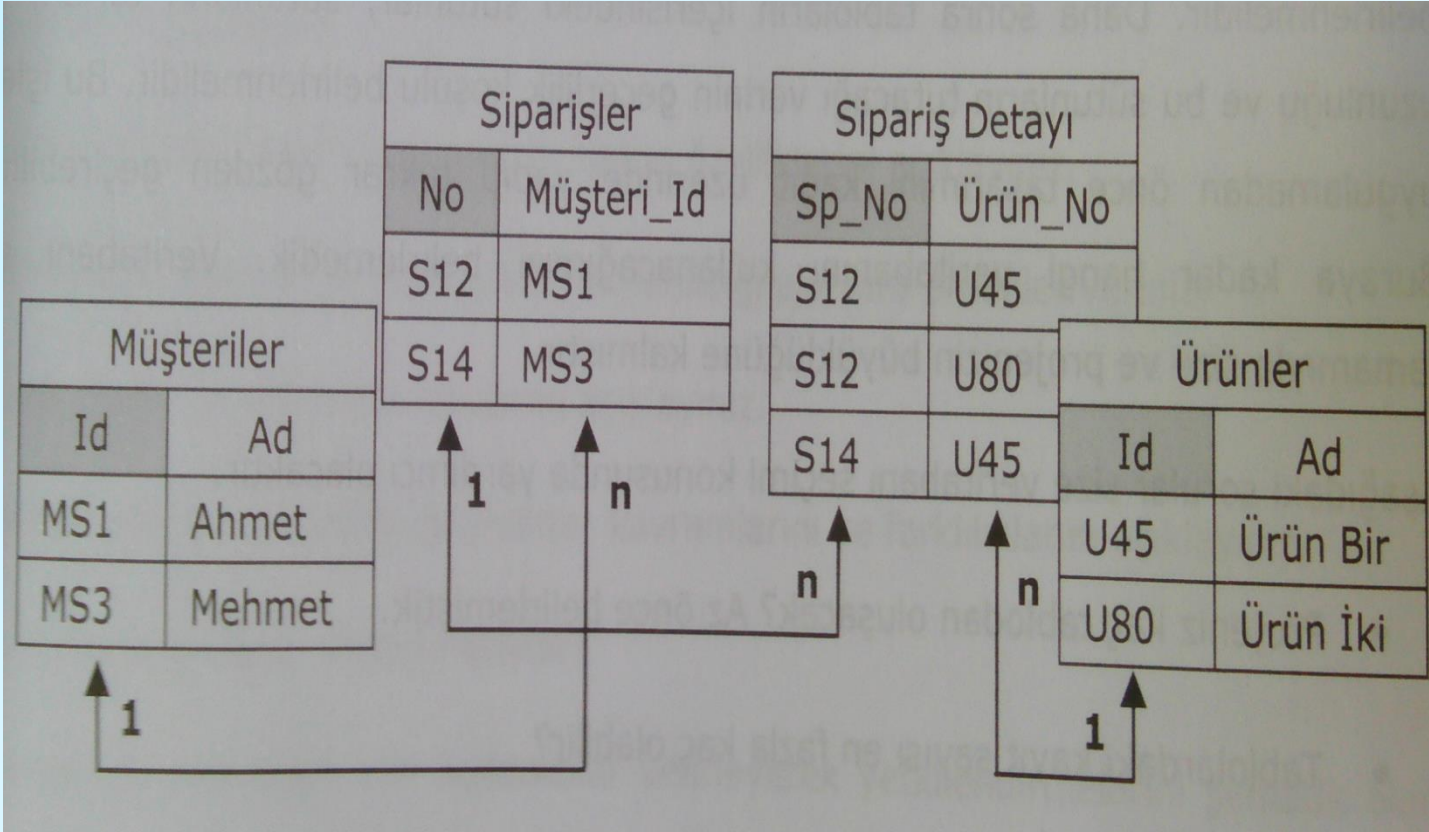
bilet_id	koltuk_no	fiyat
1	23	20
2	12	35
3	45	15
4	54	15

Bire-Çok (1-n) ilişkiler:



- Bir öğrencinin birden fazla dersi ve o derslerin final sınav sonucu olabilir. Ama aynı öğrenci numarasına sahip bir başka öğrenci olamaz.
- Müşteriyi bir kere tanımlarsınız çok kere ürün satarsınız.
- Hastayı bir kere tanımlarsınız çok kere muayene edersiniz.
- En yaygınıdır.

Çoğa-Çok (n-m) İlişkiler



- Ürünler, siparişler ve müşteriler tablosu olan bir veritabanında bir müşteri birden fazla sipariş verebilir, bir sipariş birden fazla ürün içerebilir ve bir ürün birden fazla siparişte yer alabilir.
- Örneğin birden fazla ders bir öğrenciye ait olabilir veya birden fazla öğrenci aynı dersi alabilir.

İlişki Kuralları

- Eğer tablodaki alanlara girilen **bilgiler tekrarlıyorsa, yani aynı bilgi defalarca giriliyorsa ve bu bilgi değişkense**, bunu ayrı bir tablo ile tanımlamak hem veritabanının **boyutu ve hızı** açısından hem de bilgi girişinin kolay olması açısından avantajlıdır. Örn: çalıştığı birim, memleketi, bölüm vs. gibi alanlar ayrı bir tabloda tanımlanarak asıl tablo ile ilişkilendirilmelidir.
 - Cinsiyeti, medeni hali gibi tekrarlayan ancak **değişken olmayan** bilgiler için ayrıca bir tablo tanımlanmaz.
 - İlişki kurulacak alanın veri tipi aynı olmalıdır. Metin ile Sayı arasında ilişki kurulmaz.
 - Alanların boyutları da aynı olmalıdır. int-long int
- * Son iki durumla ilgili bazı VTYS sistemlerinde istisnalar olabilmektedir.

Veritabanı tasarım adımları:

- Gereksinim Analizi:
- Kavramsal Tasarım → Varlık türleri ve öğeleri belirlenir TABLO GÖSTERİMİ
- Mantıksal Tasarım → Varlık İlişki diyagramları
- Sistemin Uyarlanması → Tabloların oluşturulması, veri girişi ve arayüz programı
- Sistemin test edilmesi ve bakım

VERİTABANI TASARIMI

- **1-) Depolanacak Veriler:** Veritabanı içerisinde tutmak istediğiniz bilgiler belirlenerek, bunlar gruplanmalıdır.
- OKUL VERİTABANI
 - Öğrenciler (no, adi, tcno, bölümü)
 - Kurum Bilgileri (adi, mudur, adres, telefon)
 - Notlar (öğrenci no, dersi, not1, not2)
 - Dersler (ders kodu, ders adı)
 - Bölümler (bölüm kodu, bölüm adı)
- **GRUPLAR → TABLOLARI**
- **İÇERİSİNDEKİ BİLGİLER İSE → SÜTUNLARI OLUŞTURUR**

VERİTABANI TASARIMI-devamı

2-) Tabloların Oluşturulması: Belirlenen veri grupları ve sütunlar doğrultusunda tablolar oluşturulur. Tablo ve sütun isimlerinde Türkçe karakter, fonksiyon ismi kullanılmamalı ve isimler çağrıştıracak şekilde belirlenmelidir.

3-) Anahtar Sütunların Belirlenmesi: Kayıtların birbirinden ayırt edilebilmesi için anahtar sütun oluşturulur. Anahtar sütunun tanımlanma zorunluluğu yoktur ama verilere daha çabuk ulaşmak ve tekrar eden kayıtların önlenmesi için kullanılır.

4-) Tabloları Bölme: Tasarım esnasında tabloda tekrar eden kayıt ile karşılaşılacaksa tekrar eden sütun için yeni tablo oluşturulur.

5-) İlişkilerin Kurulması: İlişkiler sorgu yardımıyla oluşturulacaksa bu aşamada gerek yoktur, ancak şematik olarak oluşturulacaksa bu aşamada değerlendirilmelidir.

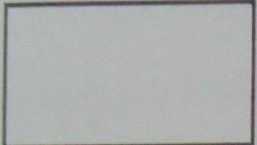
İLİŞKİSEL VERİTABANININ KAVRAMSAL TASARIMI

- Veritabanında tutulacak verilerin yönetim seviyesinde gösterilmesi için kullanılır. Kavramsal Tasarım için en çok kullanılan model ER (Entity Relationship-Varlık İlişki) modelidir.
- Kullanılan VTYS sisteminden bağımsız modelleme yapma imkanı sağlar. Bu sayede daha sonra istenilen herhangi bir VTYS veritabanına dönüştürülebilir.
- Varlık İlişki modelinde kullanılan şekiller veritabanlarının şematik olarak tasarlanması için kullanılır.
- Varlık İlişki modelinde;
 - Varlık
 - Nitelik
 - Domain (Etki Alanı)
 - İlişki

Olmak üzere 4 temel öge vardır.

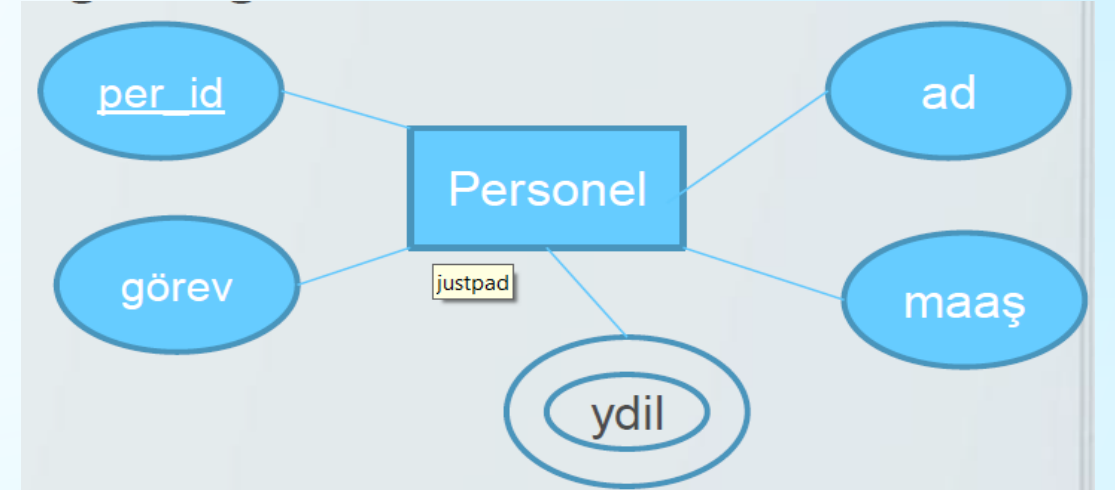
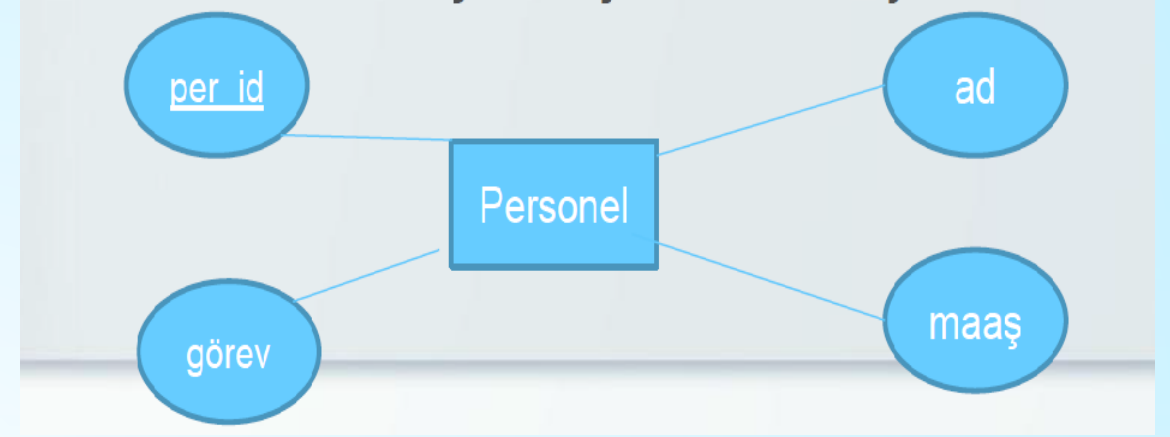
Varlık (Entity)

- Modelin en temel ögesidir. Var olan ve benzerlerinden ayırt edilebilen her şey varlıktır. Örneğin öğrenci, kitap, araba birer varlıktır. Birden fazla varlığın oluşturduğu kümeye varlık kümesi denir.
- Gösterim şekli:
 - Dikdörtgen şekil içine varlık ismi yazılarak gösterilir.
- **Veritabanı olarak düşünürsek her tablo bir varlık kümesidir.**

Sembol	
	Varlık Kümesi

Nitelik

- Varlıkların her bir özelliği nitelik olarak ifade edilir. Örneğin öğrencinin numarası ve bölümü öğrenci varlığının niteliğidir.
- Gösterim şekli: Oval ile gösterilir** ve içine ismi yazılır. Nitelik bağlı olduğu Varlığa düz bir çizgi ile birleştirilir.
- Veritabanı olarak **düşünülürse her bir sütun bir varlığı** gösterir.
- ANAHTAR NİTELİK:** Bir niteliğin değeri her bir varlık için farklı ise anahtar nitelik olarak belirlenir. Örneğin her bir öğrenci varlığı için öğrenci numarası farklı olacağından anahtar nitelik olarak gösterilir.
- ÇOK DEĞERLİ NİTELİK** ise şema **içerisinde çift oval** olarak gösterilir.



Domain (Etki Alanı):

Niteliğin alabileceği değer aralığıdır. Örneğin öğrenci notları 0-100 arasında olmalıdır. Etki alanı ER şemasında gösterilmez.

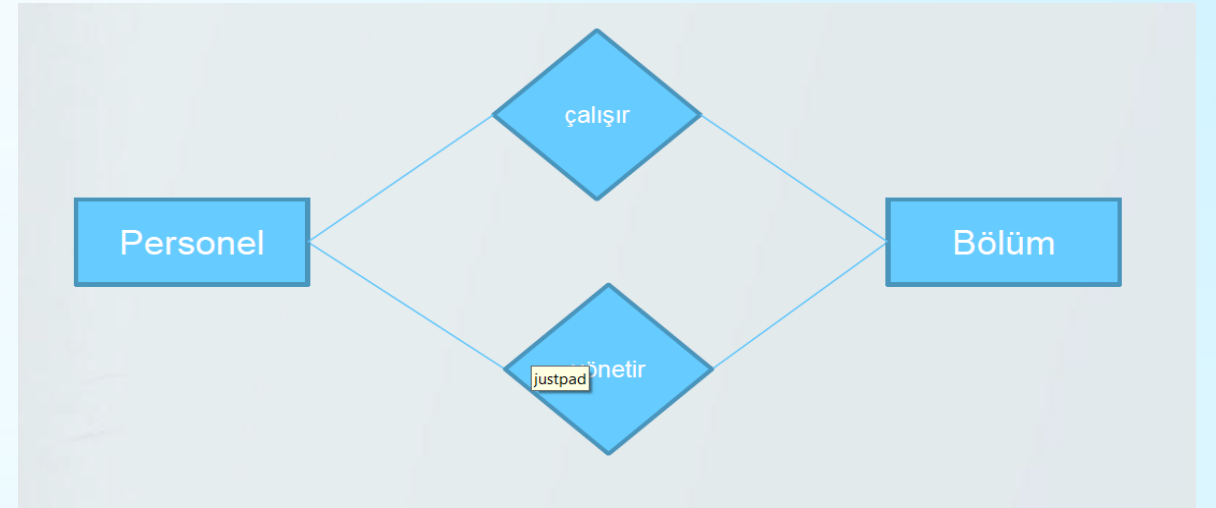
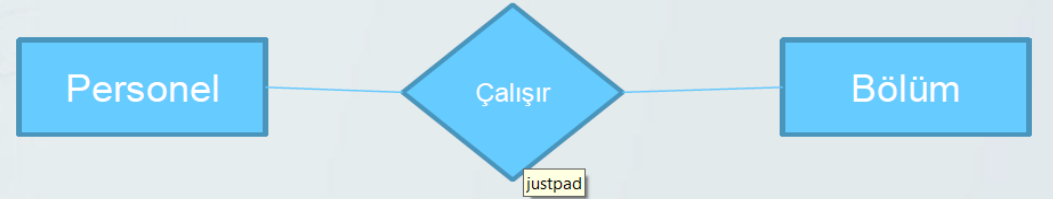
İlişki (Varlık İlişki Modeli)

- Farklı varlık kümeleri arasındaki ilişkileri ifade eder. Öğrenci ve dersler arasında ders alma ilişkisi vardır.
- Model içerisinde baklava dilimi ile gösterilir. İlişkinin ismi içerisine yazılır.
- Baklava dilimi ilişkili olduğu varlıklara düz çizgi ile bağlanır.
- Varlıklar arasında 1-1,1-n,ve n-m ilişki olabilir.
- İki varlık kümesi arasında birden fazla ilişki olabilir.

Öğrenci ders arasındaki ilişki

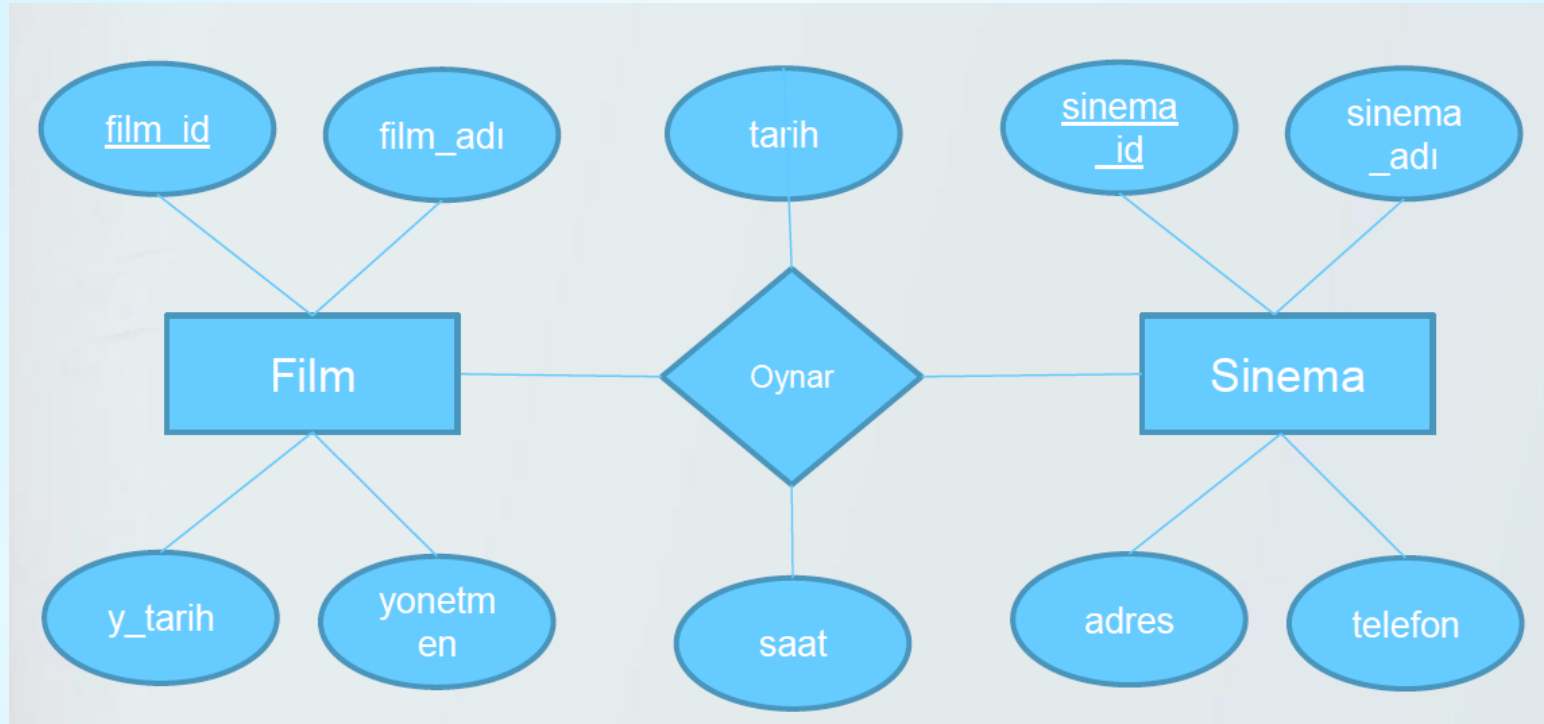


Personel bölüm arasındaki ilişki



Varlık-İlişki Modeli –İlişki (Devam)

- Varlık kümeleri arasında oluşturulan ilişkilerde ilişki sonucu nitelikler oluşabilir. Bu niteliklere **tanımlayıcı nitelik** denir.
- Örnek ; gösterime giren bir filmin sinemalarda gösterim saati ve tarihi farklıdır.

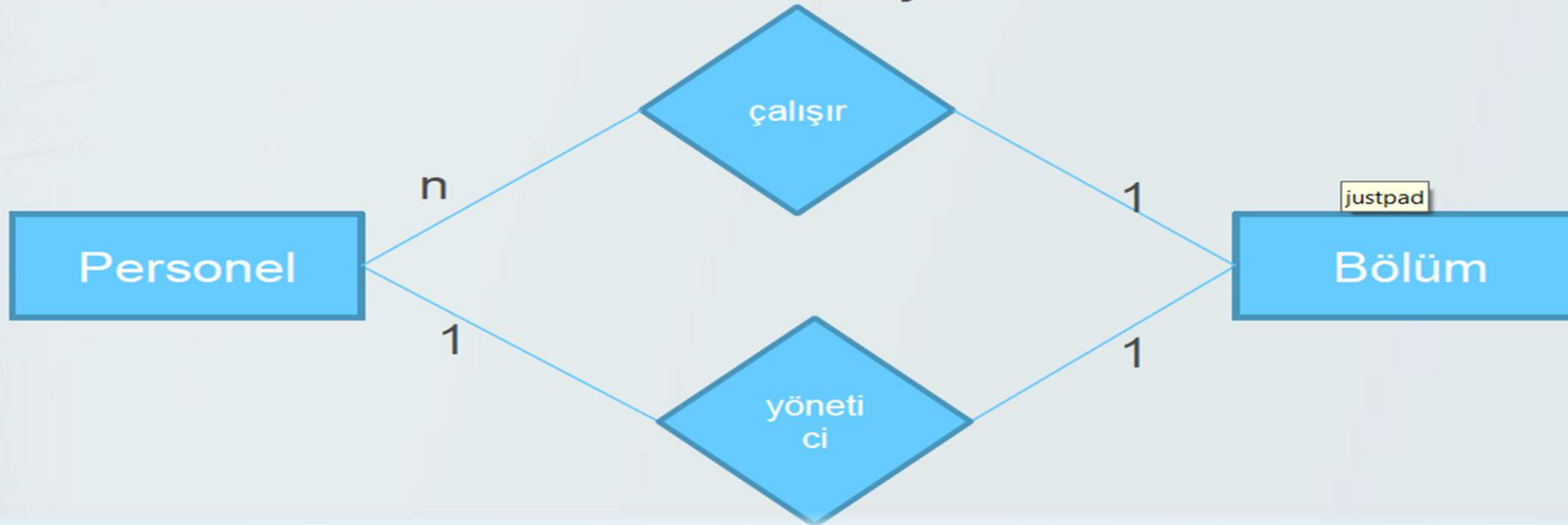


Varlık-İlişki Modeli –İlişki (Devam)

Oğrenci ders arasındaki 1-n ilişki

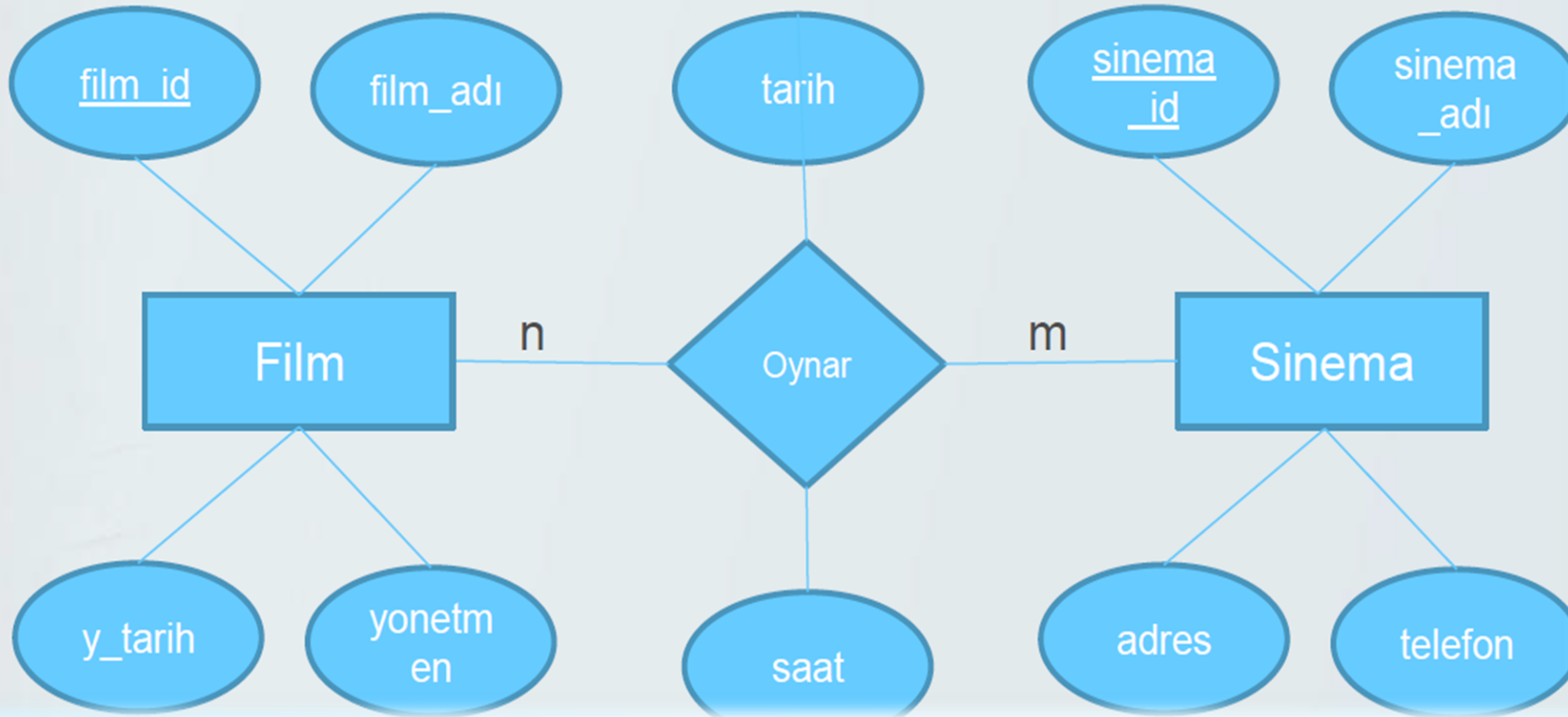


Personel bölüm arasındaki n-1 ve 1-1 ilişki



Varlık-İlişki Modeli –İlişki (Devam)

- Film ve sinema arasında n-m ilişkisi



- Çoğa-Çok ilişki örneği olarak yine film-sinema örneği verilebilir.

Varlık-İlişki Modeli –İlişki «Recursive İlişki»(Devam)

- Bir kişi hem personel hem de yönetici olabilir. Yani birden fazla kişi bir kişinin yönetiminde olabilir ve bir kişi birden fazla kişiyi yönetebilir.
- Bu tür bir ilişkii personel varlığının kendisi ile ilişkilendirilmesi sonucu elde edilir.

Varlık İlişki Modeli (Zayıf Varlık Kümeleri)

- Bir varlık kümesi anahtar niteliğe sahip değilse zayıf varlık kümesi olarak adlandırılır.
- Zayıf varlık kümeleri çift çizgili dörtgen ile gösterilir.



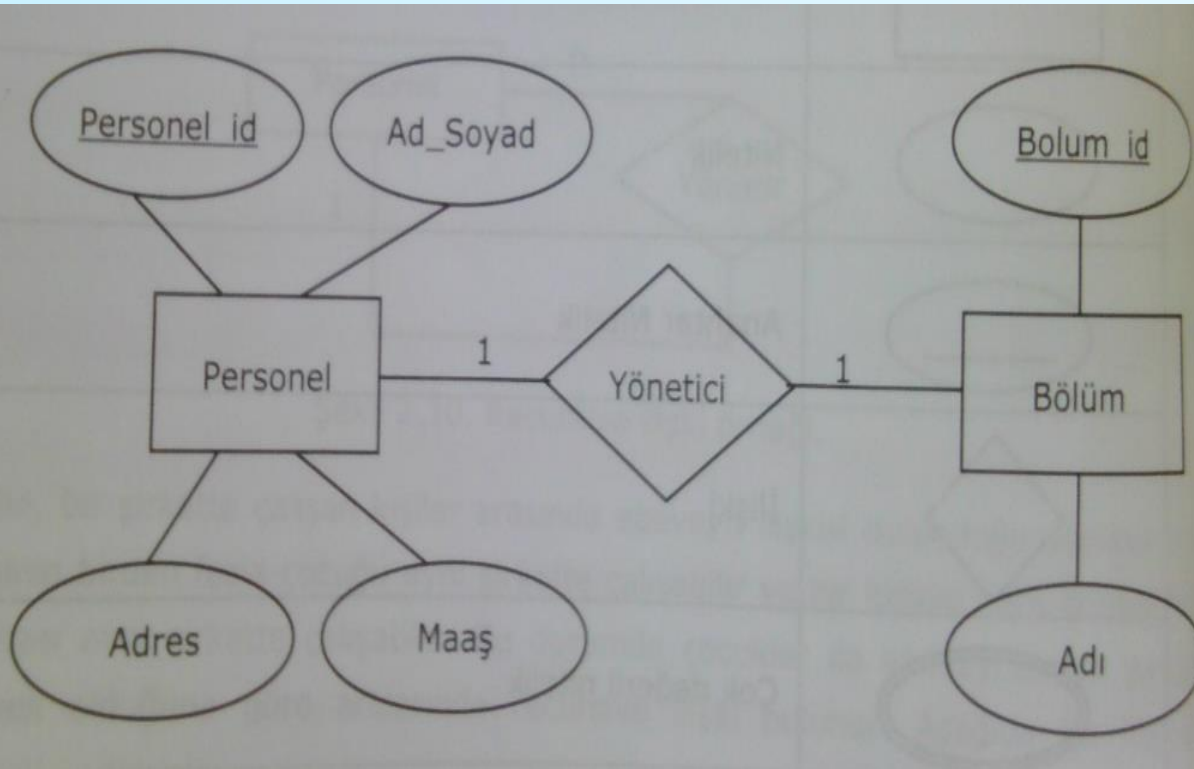
Kullanılan Semboller

Sembol	Açıklama
	Varlık kümesi
	Nitelik
	Anahtar nitelik
	İlişki
	Çok değerli nitelik
	Zayıf varlık kümesi

Bire-bir ilişkilerin tabloya dönüşümü

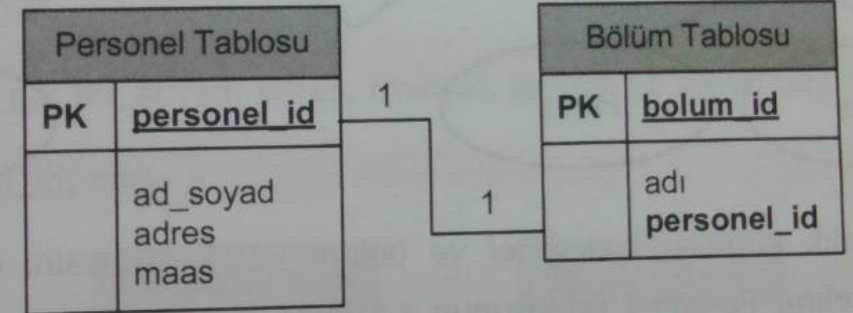
- Varlık kümelerini tablolara dönüştür
- Nitelikleri tabloların sütunlarına dönüştür
- İlişkide bir varlık kümesinin birincil anahtarı diğer varlık kümesinin yabancı anahtarı olarak belirlenir.
- Model içerisinde oluşturulan ilişkilerde tanımlayıcı nitelik bulunuyorsa, tanımlayıcı nitelikler yabancı anahtar olarak kullanılan tabloya sütun olarak eklenir.

Bire-bir ilişkilerin tabloya dönüşümü



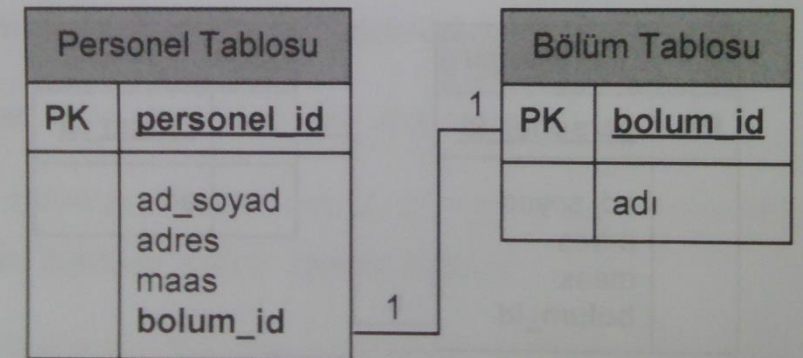
Personel(personel_id, ad_ soyad, adres, maas)

Bölüm(bolum_id, adı, personel_id)



Personel(personel_id, ad_ soyad, adres, maas, bolum_id)

Bölüm(bolum_id, adı)

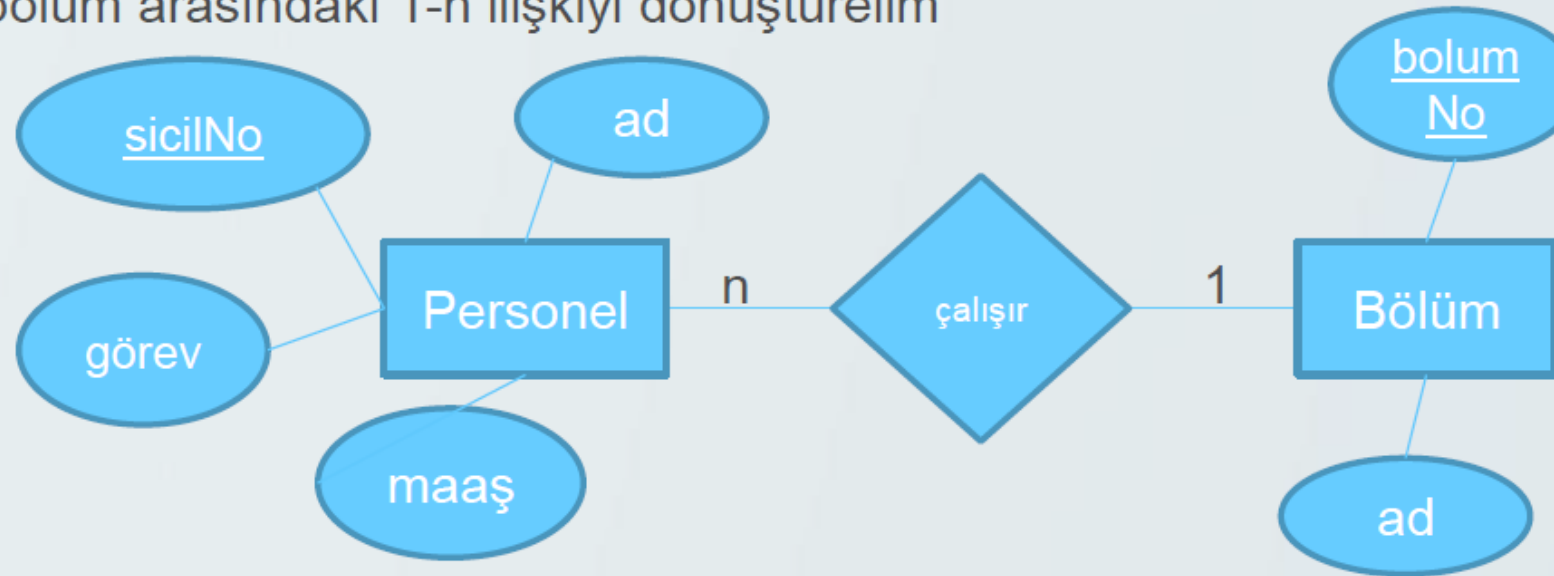


Bire-çok ilişkilerin tabloya dönüşümü

- Varlık kümelerini tablolara dönüştür
- Nitelikleri tabloların sütunlarına dönüştür
- İlişkilerin n tarafındaki tabloya 1 tarafındaki tablonun birincil anahtar sütunu yabancı anahtar olarak eklenir.
- İlişkilerde tanımlayıcı nitelik bulunuyorsa tanımlayıcı nitelikler ilişkinin n taraftaki tabloya sütun olarak eklenir.

Bire-çok ilişkilerin tabloya dönüşümü

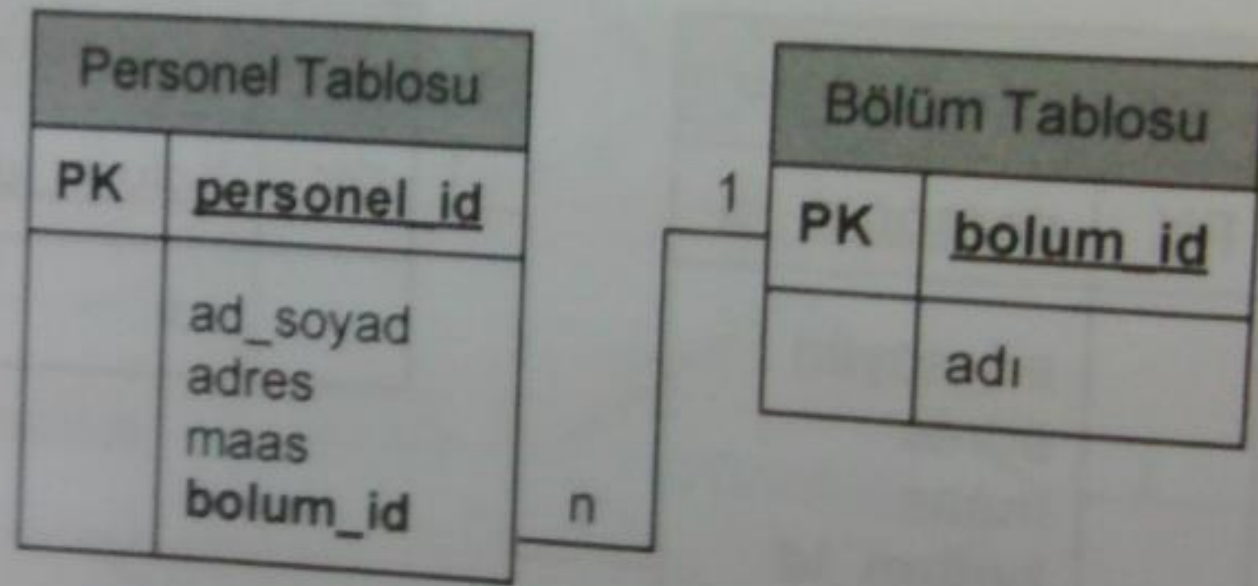
Personel bölüm arasındaki 1-n ilişkiyi dönüştürelim



1. Personel
Bölüm
2. Personel(sicilNo,ad,maas,gorev)
Bölüm(bolumNo,ad)
3. Personel(**sicilNo**,ad,maas,gorev,bolumNo)
Bölüm(**bolumNo**,ad)

Personel(personel_id, ad_soyad, adres, maas, bolum_id)

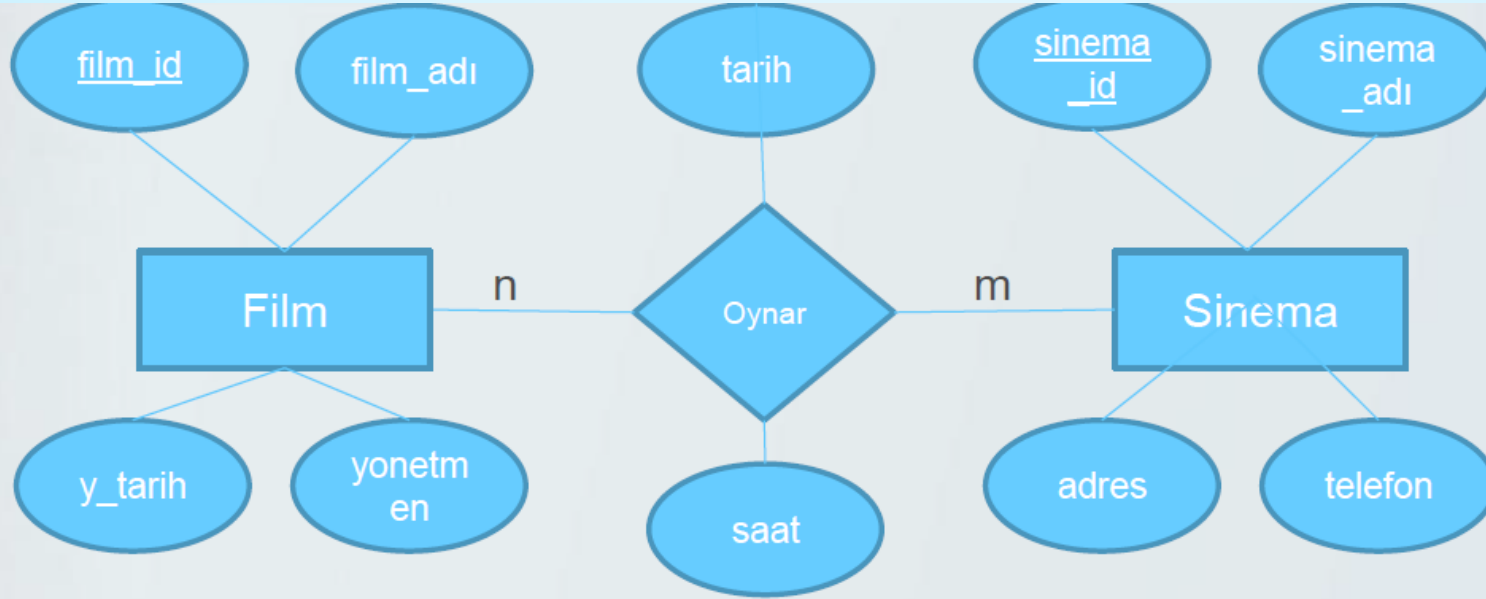
Bölüm(bolum_id, adı)



Çoğa-çok ilişkilerin tabloya dönüşümü

- Varlık kümelerini tablolara dönüştür
- Oluşturulan ilişki isminde tablo oluşturulur.
- Nitelikleri tabloların sütunlarına dönüştür.
- İlişkiyi oluşturan tabloların birincil anahtarları ilişki tablosuna (yeni oluşturduğumuz tabloya), yabancı anahtar olarak eklenir.
- İlişkide oluşturulan tablonun birincil anahtarı oluşturulan yabancı anahtarların birleşiminden oluşur. Bu şekilde oluşturulan birincil anahtar cevap vermezse yeni bir alan eklenir ve birincil anahtar yapılır.

Çoğa-Çok İlişkilerin Tabloya Dönüşümü

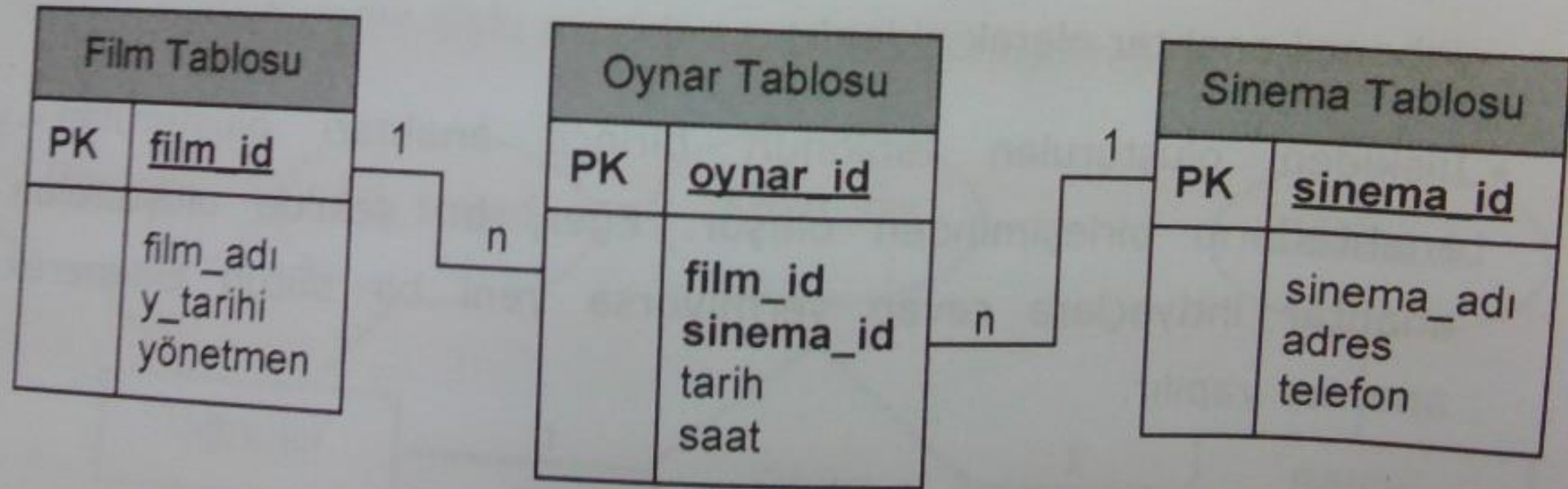


1. Film
Sinema
2. Film_Sinema_Oynar
3. Film(film_id,film_adi,y_tarih,yonetmen)
Sinema(sinema_id,sinema_adi,adres,telefon)
Film_Sinema_Oynar(tarih,saat)
4. Film(**film_id**,film_adi,y_tarih,yonetmen)
Sinema(**sinema_id**,sinema_adi,adres,telefon)
Film_Sinema_Oynar(tarih,saat,film_id,sinema_id,**oyнар_id**)

Film(film_id, film_adı, y_tarihi, yönetmen)

Sinema(sinema_id, sinema_adı, adres, telefon)

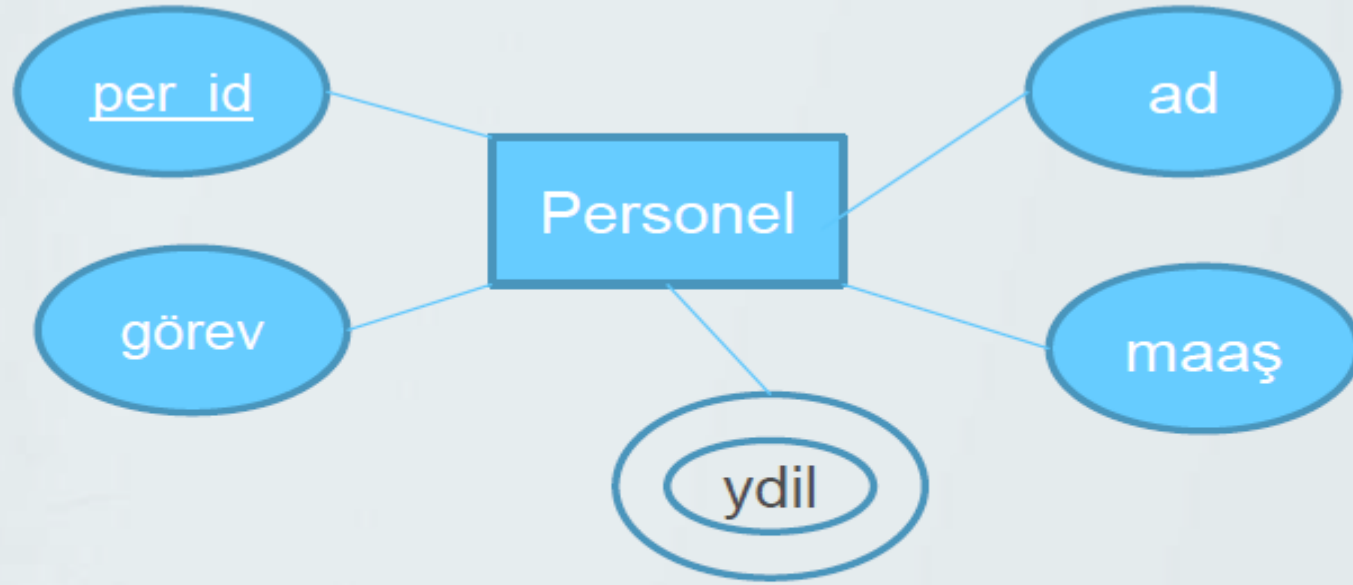
Oynar(oynar_id, film_id, sinema_id, tarih, saat)



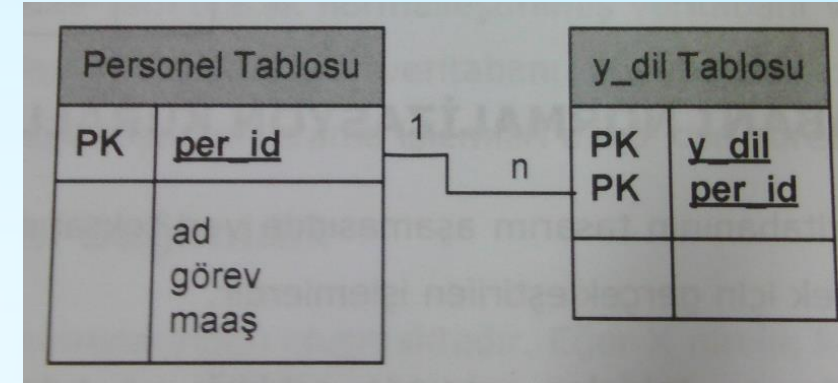
Çok değerli niteliklerin tabloya dönüşümü

- Varlık kümelerini tablolara dönüştür
- Nitelikleri tabloların sütunlarına dönüştür.
- Çok değer içeren nitelik için tablo oluştur.
- Oluşan tabloya çok değerli niteliği ve bağlı bulunduğu varlığın birincil anahtarını yabancı anahtar olarak ekle.
- Oluşan tablonun birincil anahtarı varlığın birincil anahtarı ve çok değerli niteliğin birleşiminden oluşmaktadır.

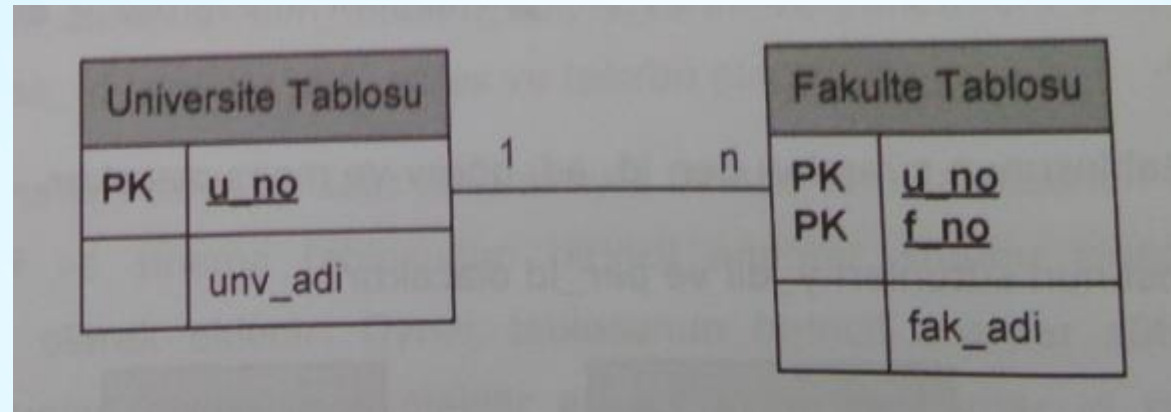
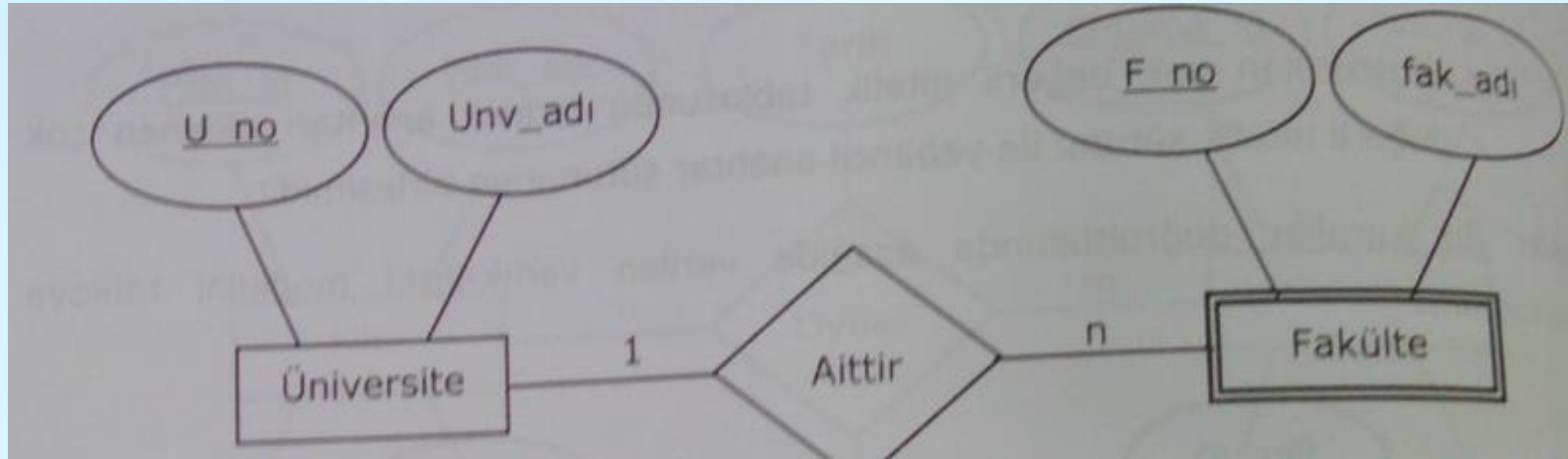
Çok değerli niteliklerin tabloya dönüşümü (devam)



1. Personel
2. Personel(per_id,ad,gorev,maas)
3. yDil(ydil,per_id)
4. yDil(ydil,per_id,siraNo)



Zayıf Varlık Kümesinin Tabloya Dönüşümü



EK BİLGİ

(Zayıf Varlık Kümesinin Tabloya Dönüşümü)

- * Her varlık kümesi için bir anahtar bulmak mümkün olmayabilir.
- * Eğer bir varlık kümesinin niteliklerinin tümü alınsa bile bir anahtar oluşturmuyorsa bu varlık kümesine **zayıf** (weak) varlık kümesi denir.

EK BİLGİ

(Zayıf Varlık Kümesinin Tabloya Dönüşümü)

- Türkiye'deki tüm lise öğrencilerinin bilgilerini içeren ÖĞRENCİ varlık kümesi zayıf bir varlık kümesidir. Çünkü farklı liselerde **öğrenci numarası, adı ve soyadı aynı olan öğrenciler** bulunabilir. (Derste belirttiğimiz üniversitelerin altında benzer isimde fakülte olabilir durumu...)
- ÖĞRENCİ varlık kümesi ile LİSE varlık kümesi arasında bir OKUYAN ilişkisi kurulursa, öğrencileri birbirinden ayırdetmek için kullanılan ÖĞR_NO niteliğine, LİSE varlık kümesinin anahtarı olan LİSE_KODU eklenir. Bu durumda ÖĞRENCİ varlık kümesinin anahtarı (LİSE_KODU, ÖĞR_NO) ikilisi olur.