



VERİ TABANI VE YÖNETİM SİSTEMLERİ

Dr. Öğretim Üyesi
Alper Talha KARADENİZ





GENEL
DERS
İÇERİĞİ

- 1. Veritabanı Kavramları**
- 2. Geçerlilik Kuralı**
- 3. İlişkisel Veri Modeli**
- 4. Varlık-İlişki Modelindeki Temel Ögeler**
- 5. Örnek Soru**



Veritabanı Kavramları

Tablo

- Veritabanı içerisinde verilerin tutulduğu yapılardır.
- Tablolar satır ve sütunlardan oluşur.
- Bir veritabanı içerisinde birden fazla tablo olabilir ve bu tablolar birbiriyle ilişkili olabilir.

- **Satır (Row) ve Sütun (Column)**
- Sütun tablo içerisinde tutulan her bir veri türüne göre verilen isimdir.
- Satır ise tablo içerisinde sütuna ait veri grubudur.





Veritabanı Kavramları

Veri Tipleri

- Veri tipleri, bir sütunun ne tür veriler tutacağını belirtir.
- Tablolar oluşturulurken alanların saklayacağı veri tiplerinin belirtilmesi gerekir. Girilecek veri, metin, tam sayı, ondalıklı sayı ya da tarih olabilir

Anahtar Kullanımı

- Anahtarlar, tablolarda veri bütünlüğünü ve ilişkileri sağlamak için kullanılır.
- Birincil anahtar, yabancı anahtar ve tekil anahtar olarak üç başlıkta inceleyeceğiz.



1-Birincil Anahtar (Primary Key)

- Kayıt içerisinde benzersiz olan sütundur. Örneğin öğrenci tablosundaki öğrenci numarası sütunu bir birincil anahtardır.
- Aynı öğrenci numarası 2 farklı öğrencide olamaz. Kayıtları birbirinden ayırmamızı sağlar.
- Birincil anahtar olarak tanımlanan alan boş (null) veya birbirinin aynı olan değerler alamaz.

```
CREATE TABLE Ogrenciler (  
    OgrenciID INT PRIMARY KEY,  
    Ad VARCHAR(50),  
    Soyad VARCHAR(50)  
);
```





2-Yabancı Anahtar (Foreign Key)

- Yabancı anahtar bir sütun veya birden fazla sütunun birleşiminden oluşabilir.
- Tablolar arası ilişki kurmak için kullanılır.
- Bir tablodaki birincil anahtar ile başka bir tabloda ilişki kurmak için kullanılır.
- Birincil anahtar gibi bol (null) olamaz.



```
CREATE TABLE Ogrenciler (  
    OgrenciNo INT PRIMARY KEY,  
    Ad VARCHAR(50),  
    Soyad VARCHAR(50)  
);
```

```
CREATE TABLE AldigiDersler (  
    KayitID INT PRIMARY KEY,  
    OgrenciNo INT,  
    DersAdi VARCHAR(100),  
    FOREIGN KEY (OgrenciNo) REFERENCES Ogrenciler(OgrenciNo)  
);
```

3-Tekil Anahtar (Unique Key)



- Birincil anahtarda olduğu gibii benzersiz bir değere sahiptir.
- Birincil anahtardan farkı ise boş (null) olabilir.
- Bir tabloda birden fazla olabilir.

```
CREATE TABLE Ogrenciler (  
    OgrenciNo INT PRIMARY KEY,  
    Ad VARCHAR(50),  
    Soyad VARCHAR(50),  
    Email VARCHAR(100) UNIQUE  
);
```



Geçerlilik Kuralı



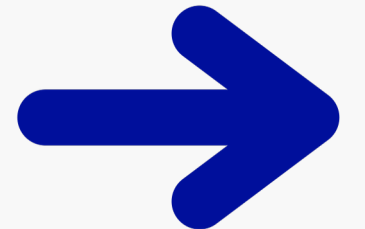
Herhangi bir sütun içerisinde girilebilecek verileri sınırlamak için kullanılır.

Geçerlilik kuralında belirtilenden daha fazla veri girişine izin verilmez. Örneğin öğrenci notlarının girildiği bir tabloda not alanına 0 ve 100 arasında not girişine izin vermek için kullanılır.

Veri Modeli

Veri modelleri, verilerin depolanması, işlenmesi, veriler arasında ilişkilerin kurulmasını sağlar. Verileri mantıksal düzeyde düzenlemek için her VTYS belirli bir veri modelini kullanır.

- 1.Sıra düzensel veri modeli
- 2.Ağ veri modeli
- 3.İlişkisel veri modeli
- 4.Nesneye yönelik veri modeli



İlişkisel Veri Modeli



İlişkisel Veri Modeli, verilerin tablolar (ilişkiler) şeklinde düzenlendiği bir veri modelidir.

Bu modelde her tablo (relation), bir varlık kümesini temsil eder. Her tablo, satır ve sütun içerir.

İlk olarak 1970 yılında E. F. Codd tarafından geliştirilmiştir.

id	musteri_adi	musteri_soyadi	adres	personel_id
1	Ahmet	Sarı	İstanbul	1
2	Süleyman	Işık	İstanbul	2
3	Neriman	Yıldız	Ankara	2
4	Tolga	Deniz	İzmir	1

id	personel_adi	personel_soyadi
1	Rıza	Keleş
2	Mehmet	Soylu

Bire bir (1-1) İlişkiler



- Bir tablodaki her kayıt, diğer tablodaki yalnızca bir kayıtla eşleşir.
- A varlık kümesinin bir elemanı, B varlık kümesinin bir elemanı ile ilişki kurabiliyorsa bu ilişki türüne bire-bir ilişki denir
- Her öğrencinin bir öğrenci kartı vardır. Her öğrenci kartı yalnızca bir öğrenciye aittir.

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

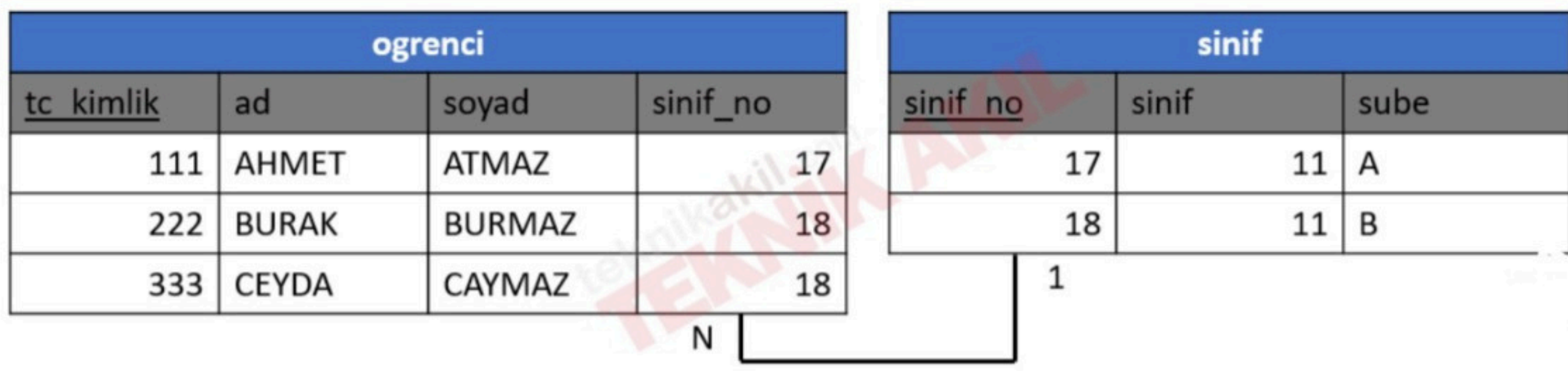


Bire çok (1-n) İlişkiler

Bir kayıt, başka bir tabloda birden fazla kayıtle eşleşebilir; ancak o kayıtlar yalnızca bir kayıtle bağlıdır.

A varlık kümesinin bir elemanı, B varlık kümesinin birden çok elemanı ile ilişki kurabiliyorsa bu ilişki bire çok ilişkidir

- Bir öğrenci birden fazla ders alabilir.
- Her ders kaydı yalnızca bir öğrenciye aittir.



Çoğa çok (n-n) İlişkiler

Bir tablodaki birden fazla kayıt, diğer tablodaki birden fazla kayıtle eşleşebilir. Avarlık kümesinin bir elemanı, B varlık kümesinin sıfır ya da birçok eleman ile bu ilişki türü çoğa çok ilişki türüdür

- Bir öğrenci birden çok dersi alabilir.
- Bir ders birden çok öğrenci tarafından alınabilir.

id	ogrenci_adl	ogrenci_soyadi
1	Elif	Türkmen
2	Ayşe	Sarı
3	Ender	Kaya
4	Ali	Demir
5	Adem	Salih

id	ogrenci_id	ders_id
1	1	3
2	1	5
3	2	1
4	3	4
5	4	2
6	4	3

id	ders_adl
1	Matematik
2	Tarih
3	Edebiyat
4	Yazılım
5	İstatistik



İlişkisel Veri Modeli ve Kavramsal Tasarımı

Şu anda kullanılan veri tabanlarının çoğu ilişkisel veri modeline destek verirler. Bu modelde ortak özelliğe sahip veriler tablolar aracılığı ile tutulur. Veriler ve ilişkiler tablolar üzerinde tanımlanır ve tüm bilgiler görülmeyecek şekildedir. İlişkiler kurulurken birincil anahtarlar ve yabancı anahtarlar kullanılır

Kavramsal Tasarım: Veri tabanında tutulacak verilerin daha üst seviyede gösterilmesi için kullanılır.

Kavramsal tasarım için en çok kullanılan model ER(Entity-Relationship) yani Varlık-İlişki modelidir.

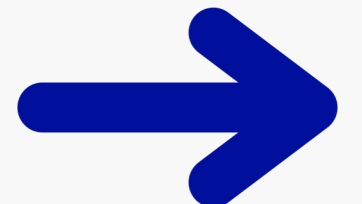




Varlık-İlişki Modelindeki Temel Ögeler

Varlık: Modelin en temel ögesidir. Var olan ve benzerlerinden ayırt edilebilen her şeydir.

- Kitap, öğrenci, araba gibi kavramlar varlık olarak kullanılabilir. Bu varlıklar model içerisinde dikdörtgen içinde gösterilir.





Varlık-İlişki Modelindeki Temel Ögeler

Nitelik: Varlıkların her bir özelliği bir nitelikdir.

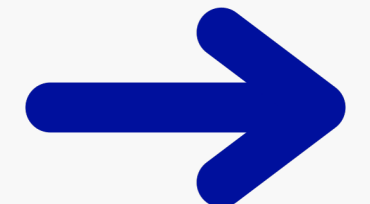
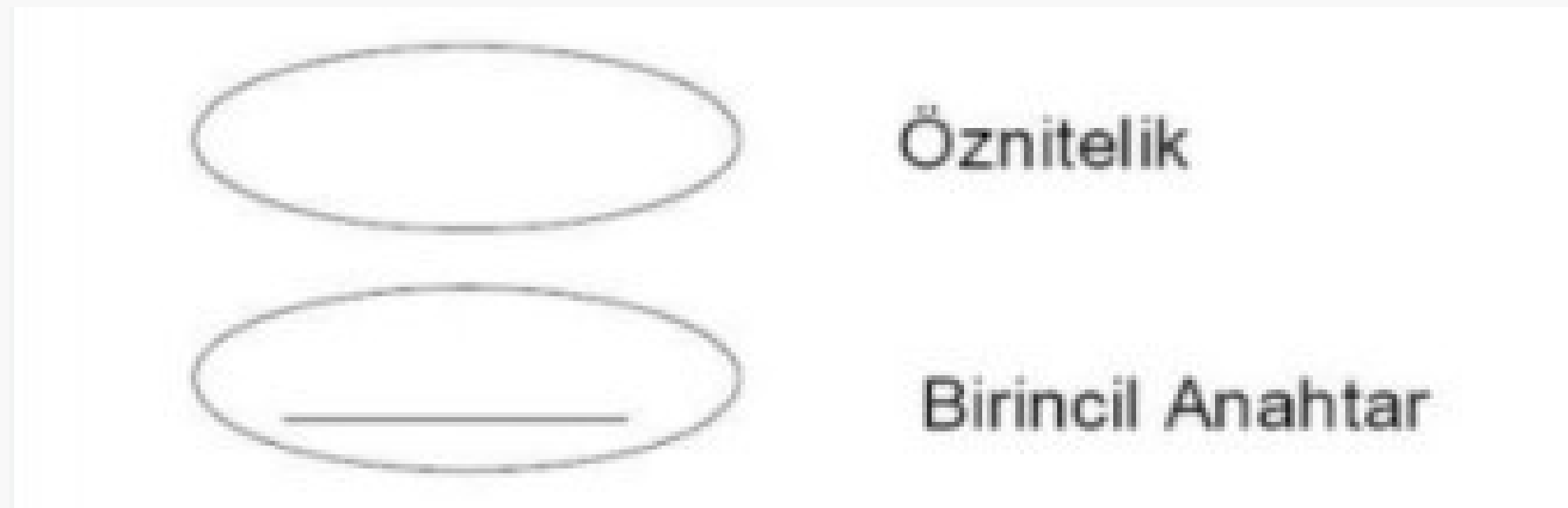
- Öğrencinin numarası, bölümü birer nitelikdir.

Model içerisinde nitelikler oval olarak gösterilir ve içerisinde niteliğin ismi yazılır.

Nitelik bağlı olduğu varlığa düz bir çizgi ile bağlanır.

Bir niteliğin değeri her bir varlık için farklıysa bu nitelik **anahtar nitelik** olarak belirlenir. Model içerisinde niteliğin altı çizilerek gösterilir.

- Öğrenci numarası, TC kimlik numarası gibi değerler anahtar niteliğe örnektir.

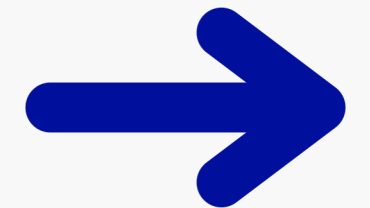
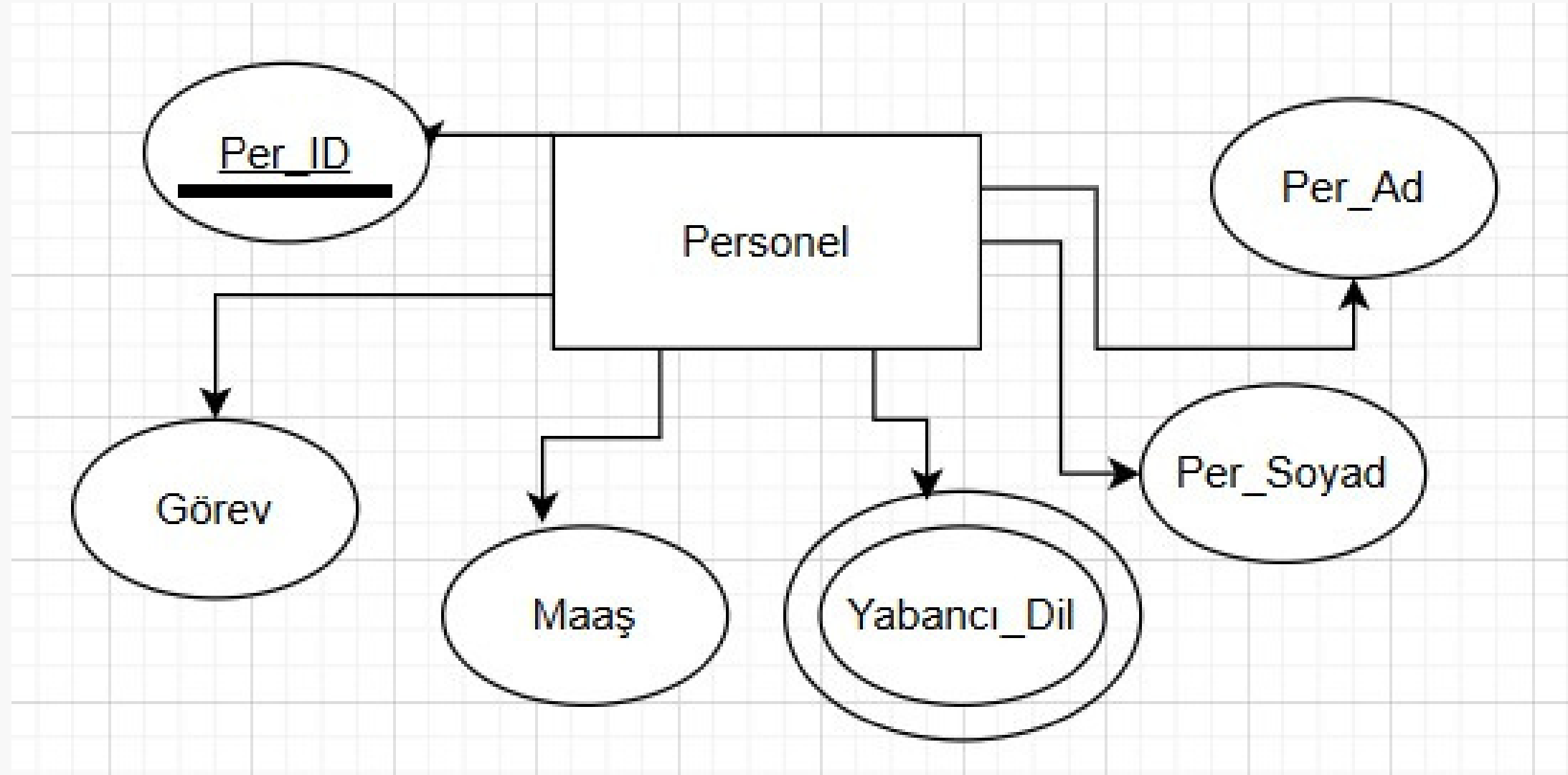


Varlık-İlişki Modelindeki Temel Ögeler



Bazı nitelikler, diğer niteliklere göre daha fazla tekrar edebilir. Böyle niteliklere çok değerli nitelikler denir.

- Bir personelin birden fazla yabancı dil bilmesi örnek olarak verilebilir.



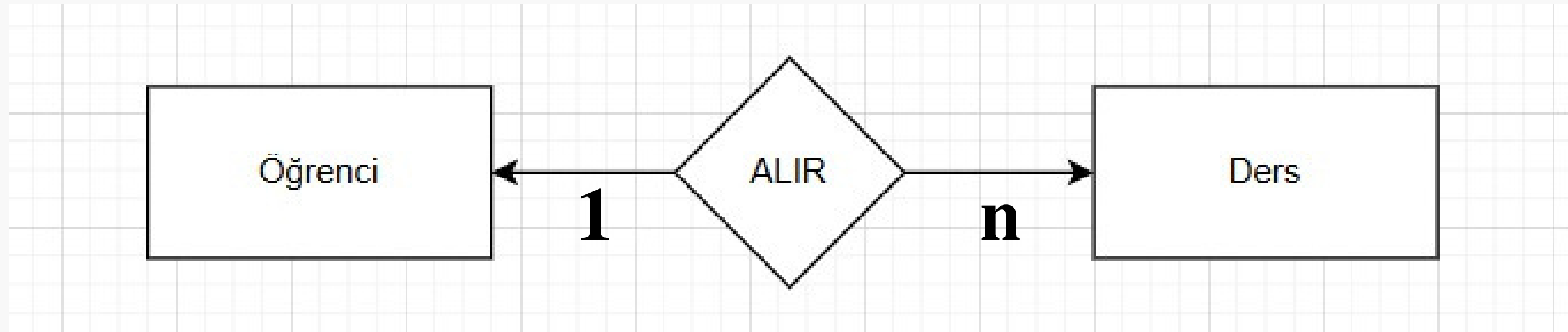
İlişki



Farklı varlıkların birbirleriyle bağlantısını ifade eder.

- Öğrenci ve dersler farklı iki varlıktır. Öğrenci ders almak zorunda olduğu için aralarında bir ilişki kurulmak zorundadır.

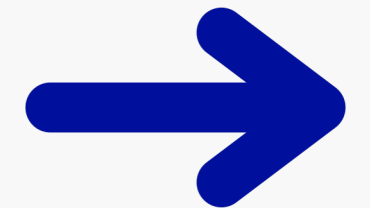
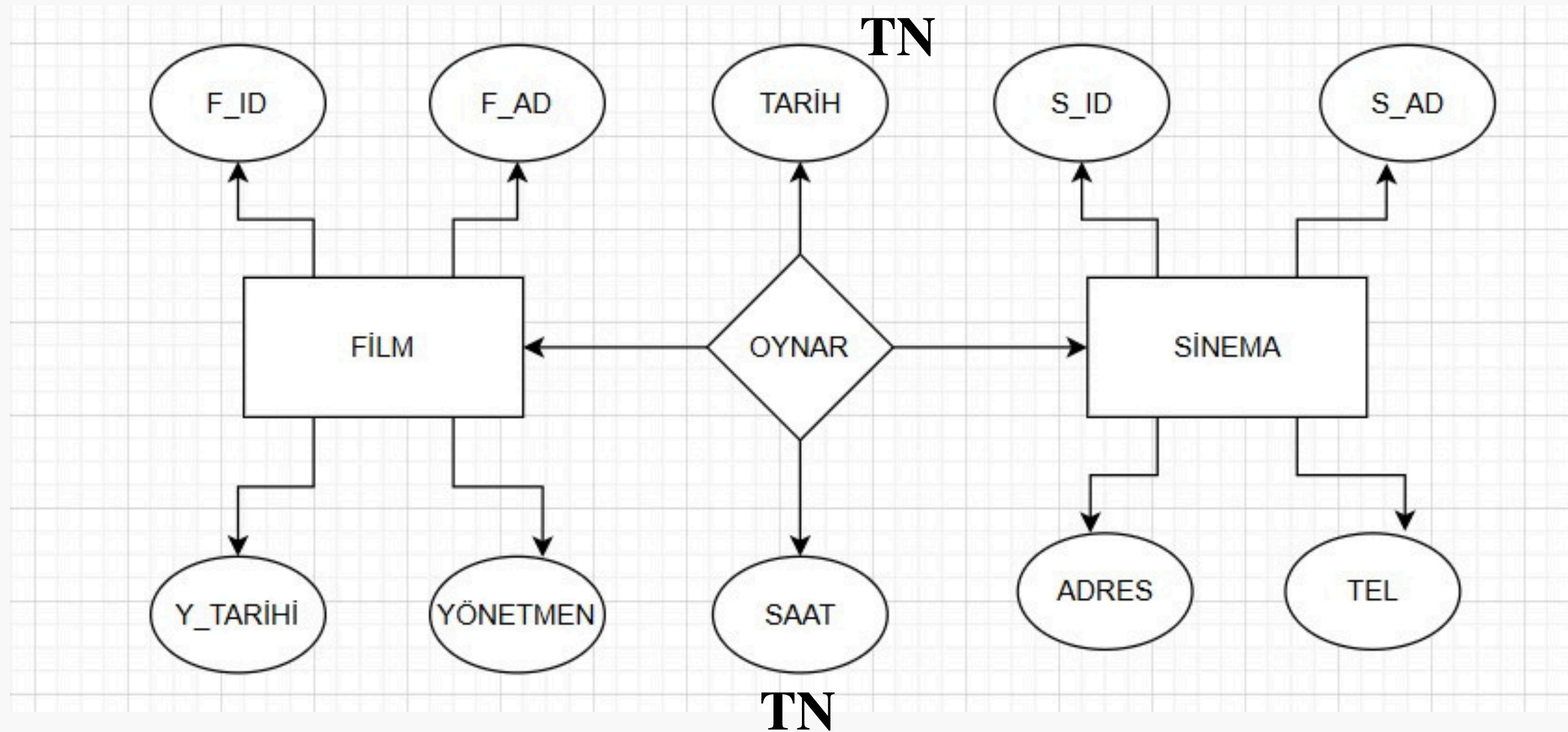
Model içerisinde baklava dilimi ile gösterilir ve içerisinde ilişkinin adı yazılır.





Varlık kümeleri arasında oluşturulan ilişkiler sonucunda nitelikler oluşabilir. Bu tür niteliklere *tanımlayıcı nitelik* denir.

Bu nitelikler diğer nitelikler gibi oval bir şekilde gösterilir ve ilişkiye bağlanır.

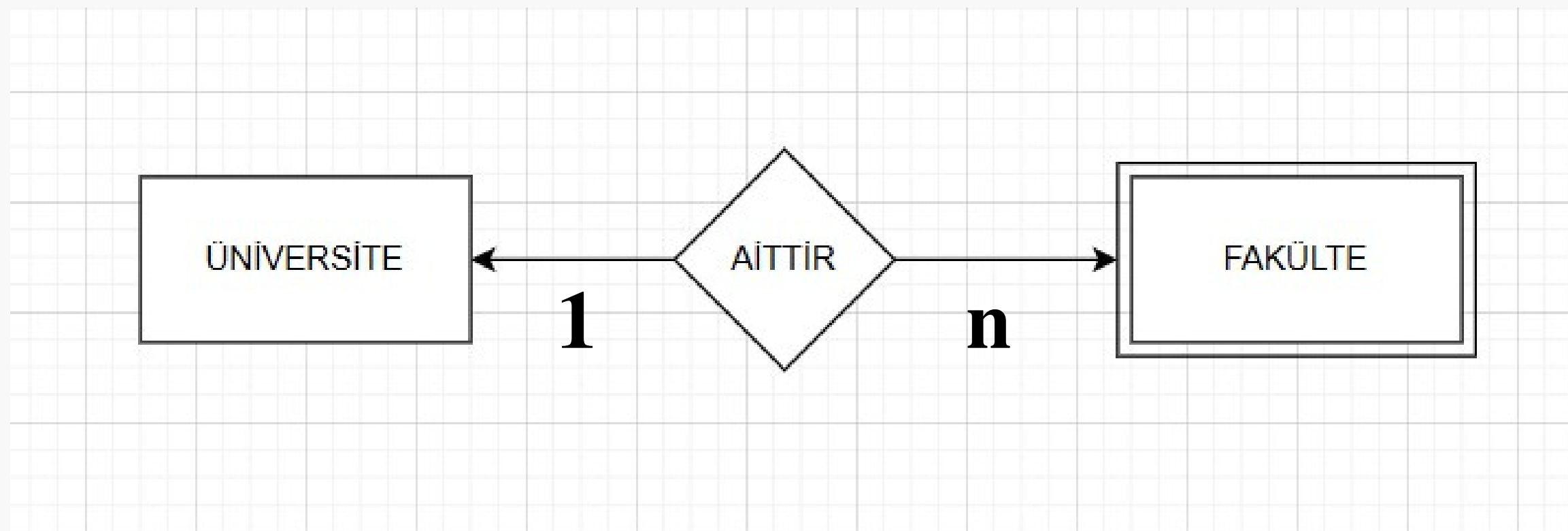




Bir varlık kümesi anahtar niteliğe sahip değilse *zayıf varlık kümesi* denir.

Zayıf varlıklar güçlü varlıklarla ilişki kurarak kullanılırlar. Yani güçlü varlıklara bağımlıdırlar.

- Örneğin üniversite olmadan fakülte olmaz ancak fakülte olmadan üniversite olabilir.



Bire-bir (1-1) İlişkilerinin Tablolara Dönüştürülmesi

- Her varlık (entity) için bir tablo oluşturulur.
- Varlığın her niteliği (attribute), sütun olur.
- Varlığın birincil anahtar (primary key) niteliği, tabloda PRIMARY KEY olarak tanımlanır.
- İki varlıktan birinin tablosuna, diğerinin primary key'i foreign key olarak eklenir.
- Tanımlayıcı nitelikler bulunuyorsa tabloya sütun olarak eklenir.

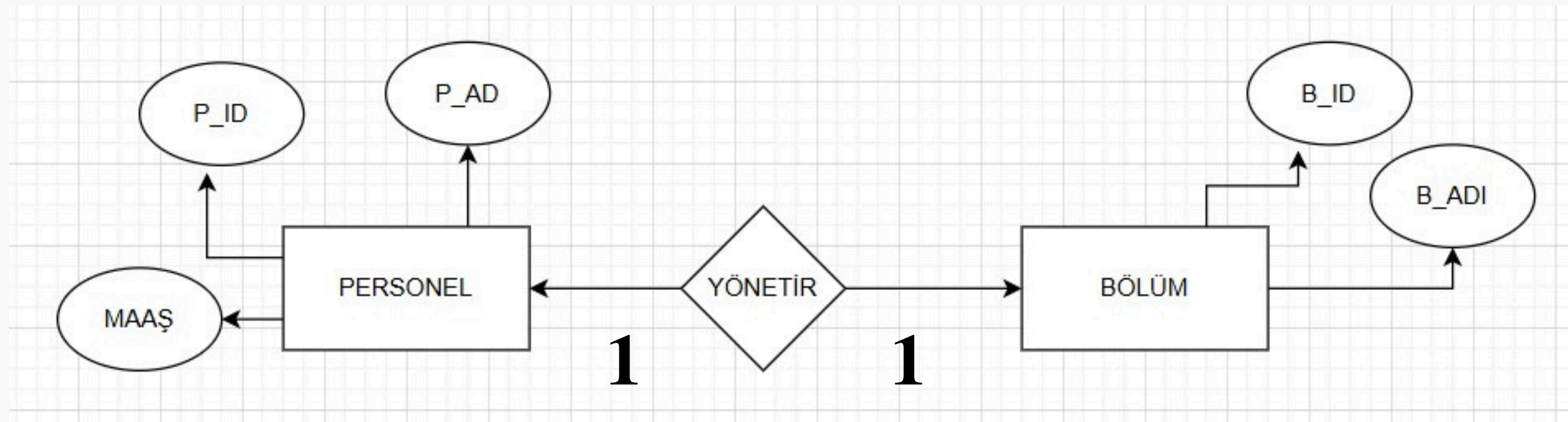


```
-- Bölüm tablosu
```

```
CREATE TABLE Bolum (  
    B_ID INT PRIMARY KEY,  
    B_Adi VARCHAR(50)  
);
```

```
-- Personel tablosu
```

```
CREATE TABLE Personel (  
    P_ID INT PRIMARY KEY,  
    P_Adi VARCHAR(50),  
    Maas DECIMAL(10, 2),  
    B_ID INT UNIQUE, -- UNIQUE eklenerek 1-1 ilişki sağlanıyor  
    FOREIGN KEY (B_ID) REFERENCES Bolum(B_ID)  
);
```



Bire-çok (1-n) İlişkilerinin Tablolara Dönüştürülmesi

- Her varlık (entity) için bir tablo oluşturulur.
- Varlığın her niteliği (attribute), sütun olur.
- İlişkinin n tarafındaki tabloya 1 tarafından tablonun birincil anahtarı sütunu yabancı anahtar olarak eklenir.
- Tanımlayıcı nitelikler bulunuyorsa ilişkinin n tarafındaki tabloya sütun olarak eklenir.

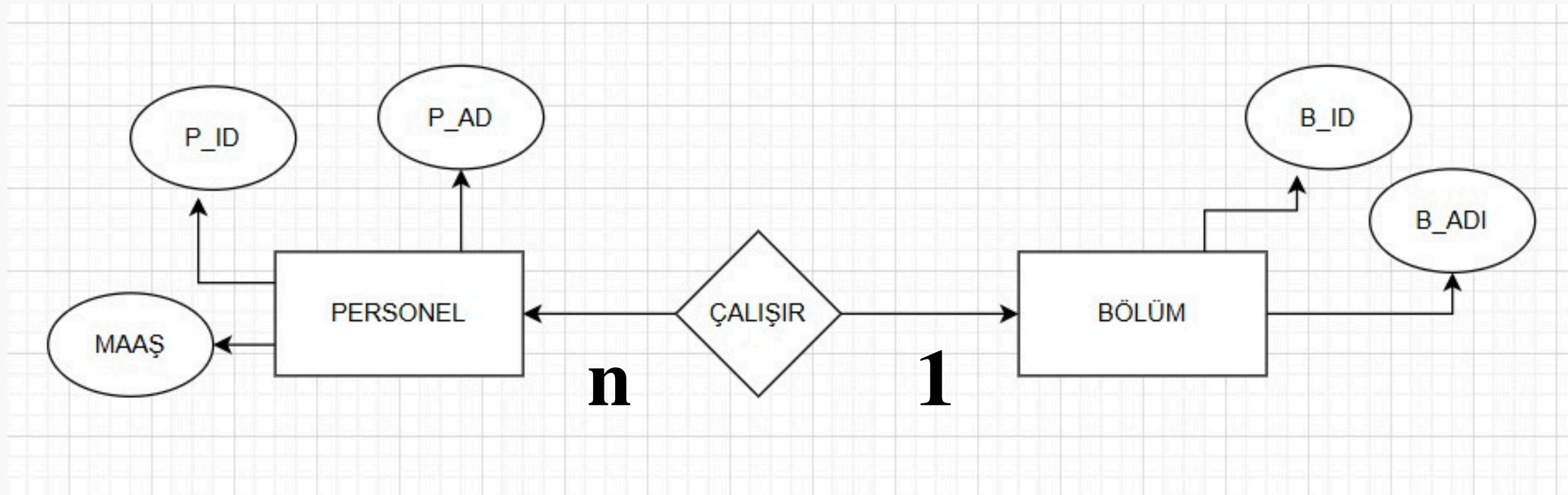


-- Bölüm tablosu

```
CREATE TABLE Bolum (  
    B_ID INT PRIMARY KEY,  
    B_Adi VARCHAR(50)  
);
```

-- Personel tablosu

```
CREATE TABLE Personel (  
    P_ID INT PRIMARY KEY,  
    P_Adi VARCHAR(50),  
    Maas DECIMAL(10, 2),  
    B_ID INT,  
    FOREIGN KEY (B_ID) REFERENCES Bolum(B_ID)  
);
```



Çoğa-çok (n-n) İlişkilerinin Tablolara Dönüştürülmesi

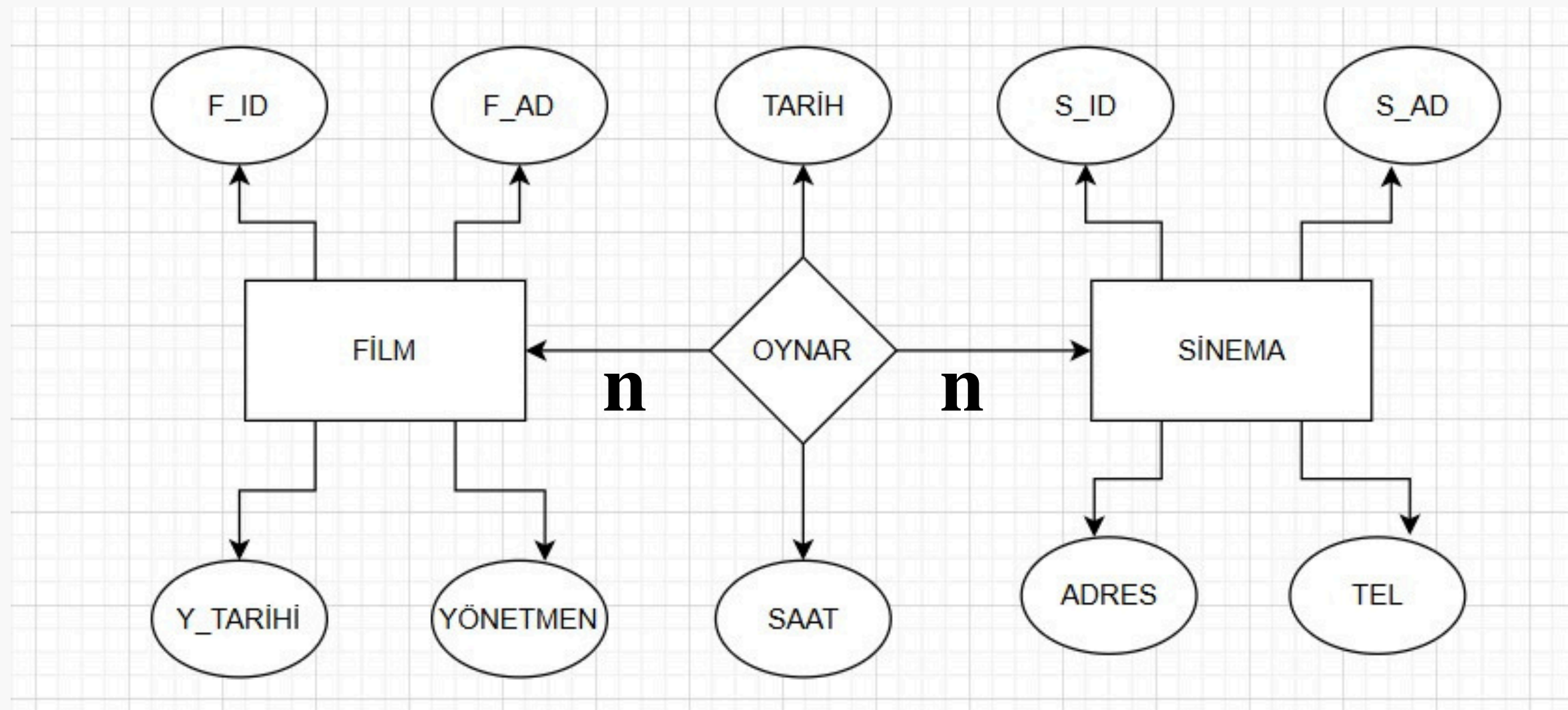
- Her varlık (entity) için bir tablo oluşturulur.
- Varlığın her niteliği (attribute), sütun olur.
- Araya bir ilişki (bağlantı) tablosu eklenir.
- Bu tablo, her iki varlığın primary key'lerini içerir → birlikte birleşik primary key olur veya id ile tekil yapılır.




```
CREATE TABLE Film (  
  F_ID INT PRIMARY KEY,  
  F_Adi VARCHAR(100),  
  Y_Tar DATE,  
  Yonetmen VARCHAR(100)  
);
```

```
CREATE TABLE Sinema (  
  S_ID INT PRIMARY KEY,  
  S_Adi VARCHAR(100),  
  Adres VARCHAR(200),  
  Tel VARCHAR(20)  
);
```

```
CREATE TABLE Oynar (  
  OynarID INT PRIMARY KEY, -- Alternatif: birl  
  F_ID INT,  
  S_ID INT,  
  Tarih DATE,  
  Saat TIME,  
  FOREIGN KEY (F_ID) REFERENCES Film(F_ID),  
  FOREIGN KEY (S_ID) REFERENCES Sinema(S_ID)  
);
```





Örnek Soru

Bir sanal alışveriş sitesi için geliştirilen veri tabanında ürünler, kategoriler, tedarikçi firmalar, müşteriler ve siparişler varlık kümelerini (ilişki türlerini de belirterek) ve size göre bu kümelerin sahip olması gereken nitelikleri de gösterecek şekilde varlık-ilişki modeli çizelgesini oluşturunuz ve bu modeli tablolara dönüştürünüz.



