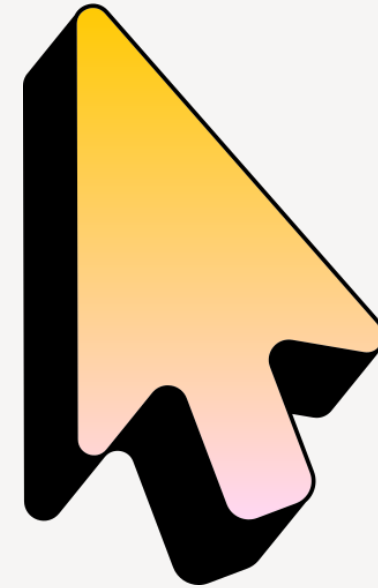


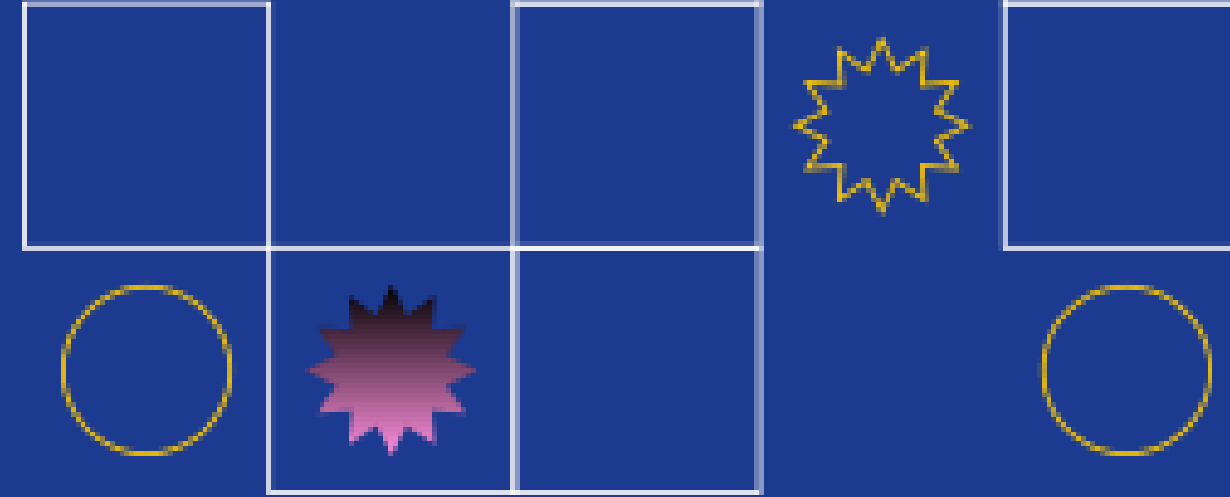
Veri Tabanı Yönetim Sistemleri

Öğr. Gör. Furkan DURMUŞ

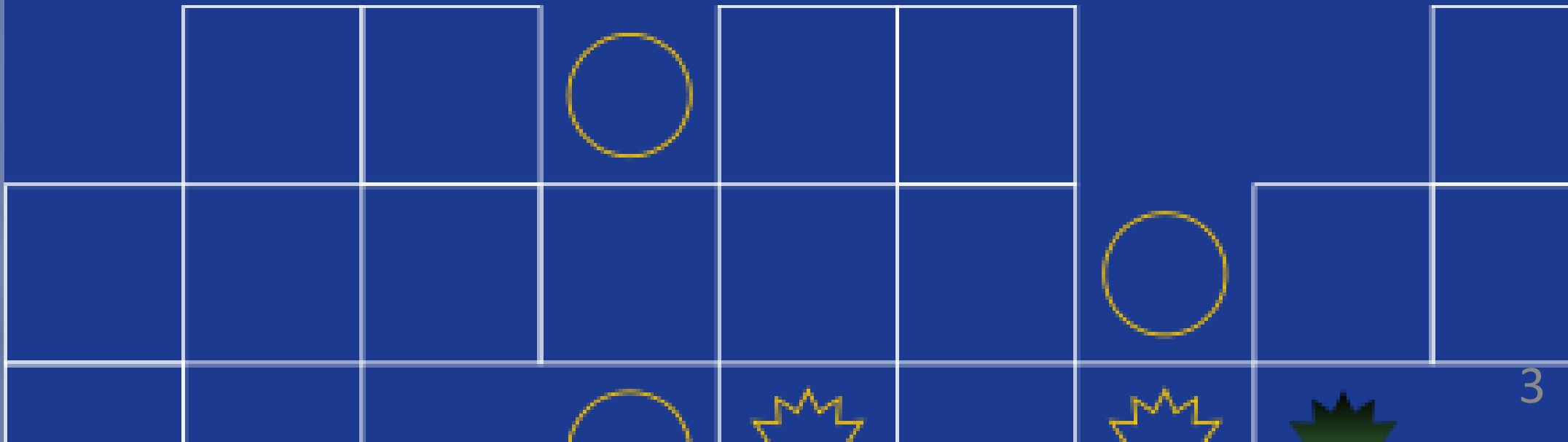
Kazanımlar

- 01** Veri ve Veri Modelleri
- 02** Veri Tabanı Yapıları
- 03** Varlık İlişki Diyagramları
- 04** İlişki Türleri
- 05** İlişki - Tablo dönüşümleri
- 06** Anahtar Türleri ve Kısıtlamalar





Veri ve Veri Modelleri

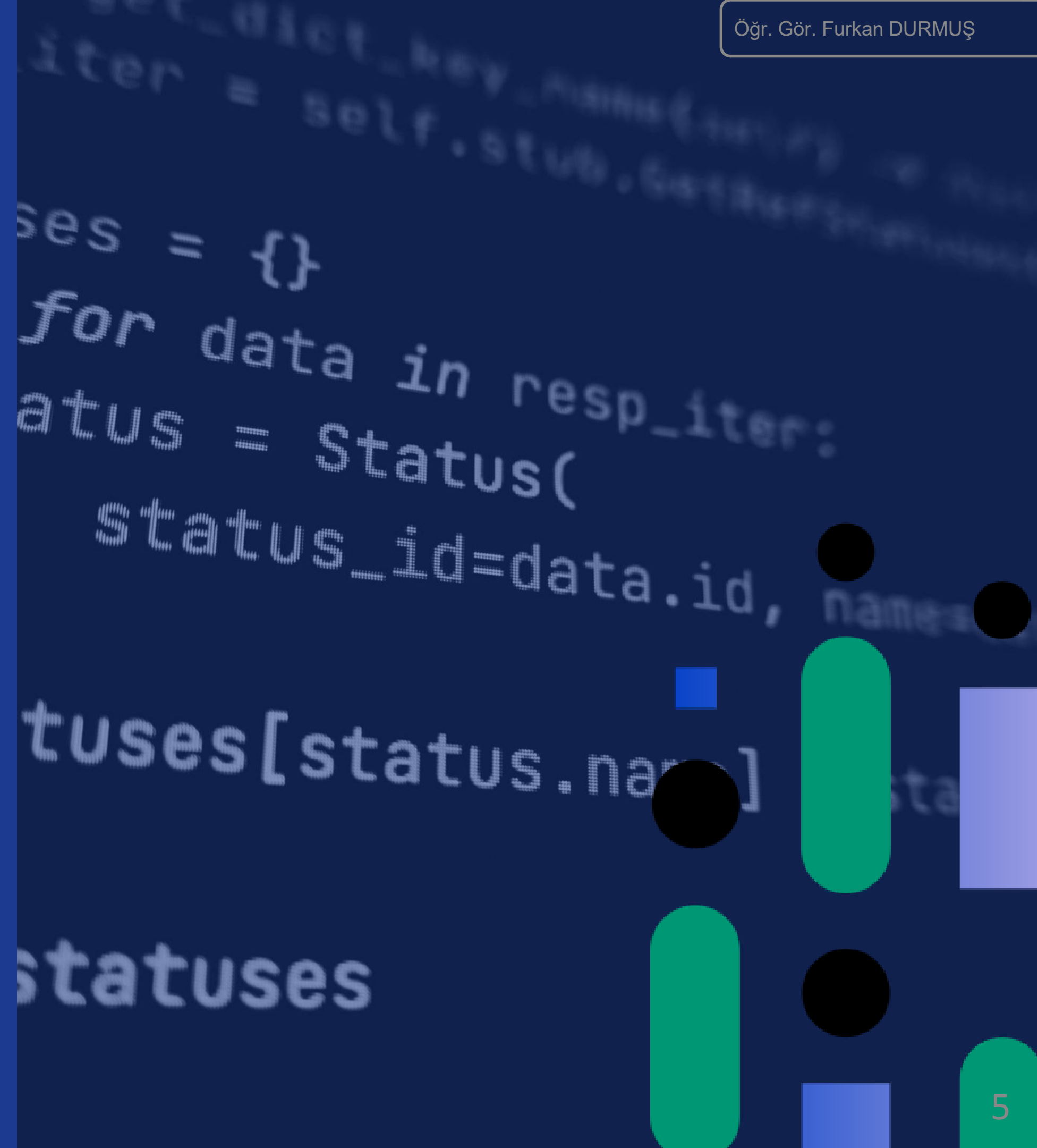


Veri

- Bir anlamı olan ve kaydedilebilen gerçekler
- Bilgisayarda işlenebilen ve kayıt edilebilen her türlü bilgi

Veri Modeli

Veri modeli; veritabanının yapısını tanımlayan kavramların bir kümesidir.



Veritabanı Modeli

Bir veritabanı modeli, veritabanının nasıl tasarlandığını ve nasıl çalıştığını belirleyen bir yapıdır. Bu model;

- Veritabanında hangi türde verilerin saklanacağını (örneğin sayı, metin gibi veri tipleri)
- Veriler üzerinde hangi işlemlerin yapılabileceğini (örneğin veriyi sorgulama veya güncelleme),
- Veritabanında hangi kuralların geçerli olacağını tanımlar (örneğin bazı verilerin değiştirilememesi gibi kısıtlamalar)

Veritabanı Modeli

Yani kısaca, veritabanı modeli, veritabanının yapısını, işleyişini ve kurallarını belirleyen plan gibidir. Bu plan sayesinde veritabanı hem düzenli çalışır hem de veriler güvenli ve tutarlı şekilde yönetilir.

Veri Modeline Göre Veri Tabanı Yapıları

Temel veri tabanı yapıları şunlardır;

1- Hiyerarşik
Veri Tabanları

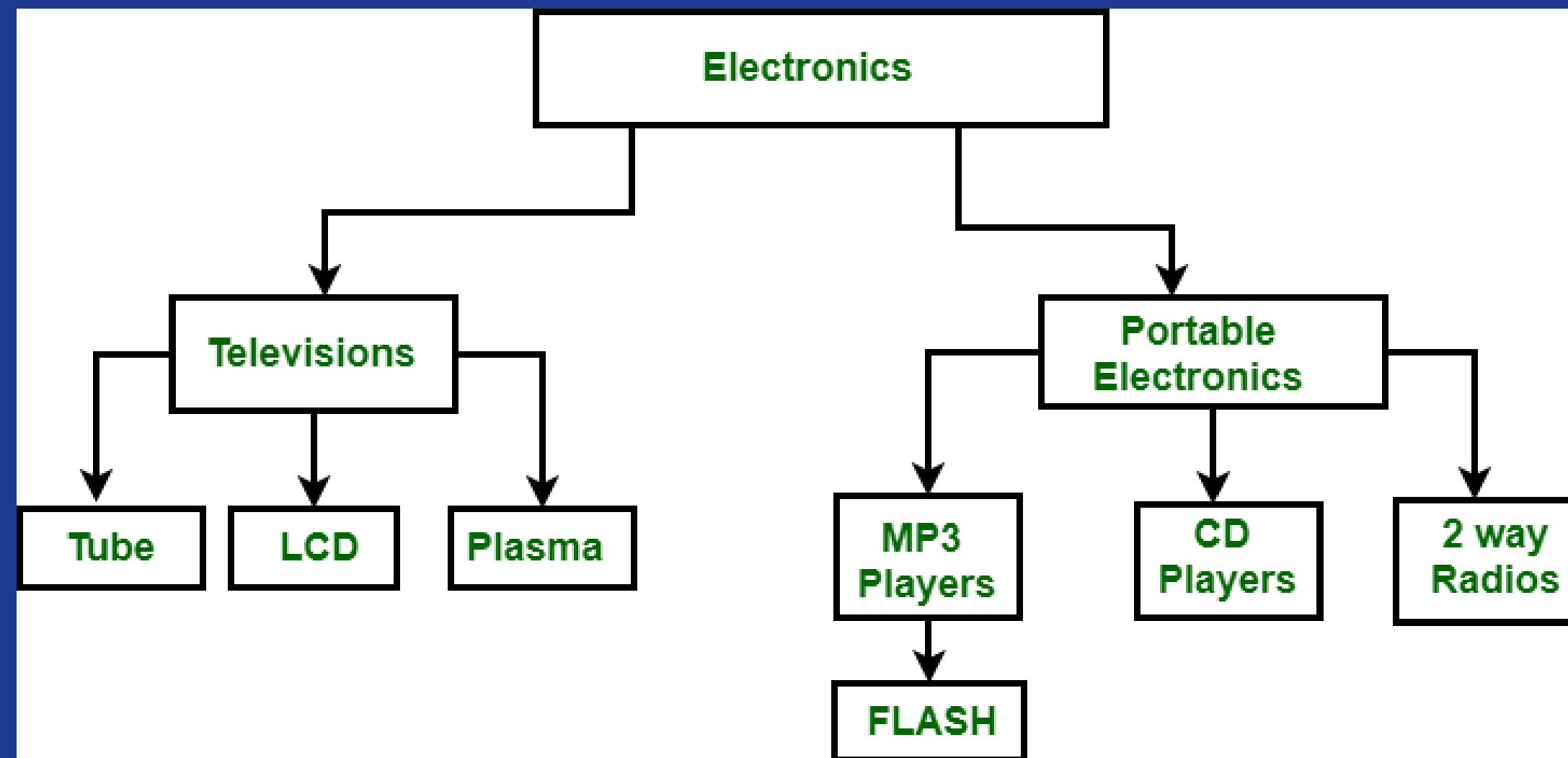
İlişkisel veri
tabanı

Ağ veri tabanı

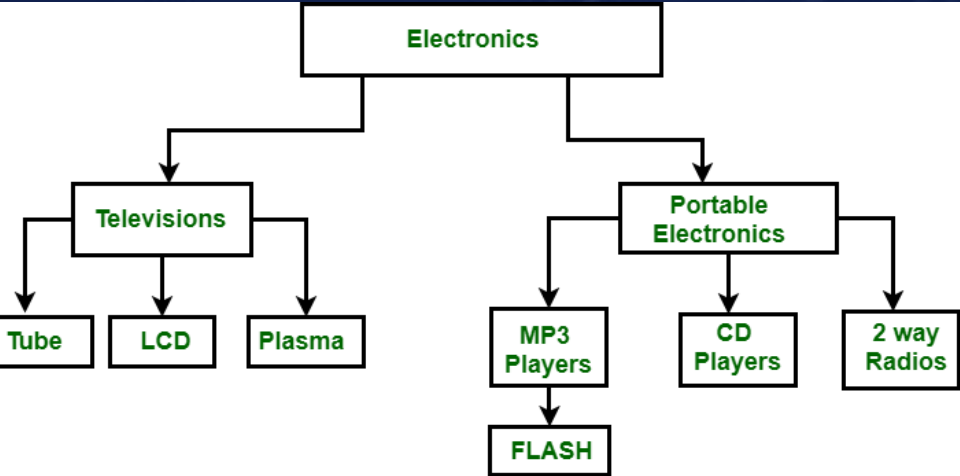
Nesne yönelimli
veri tabanı



1- Hiyerarşik Veri Tabanları



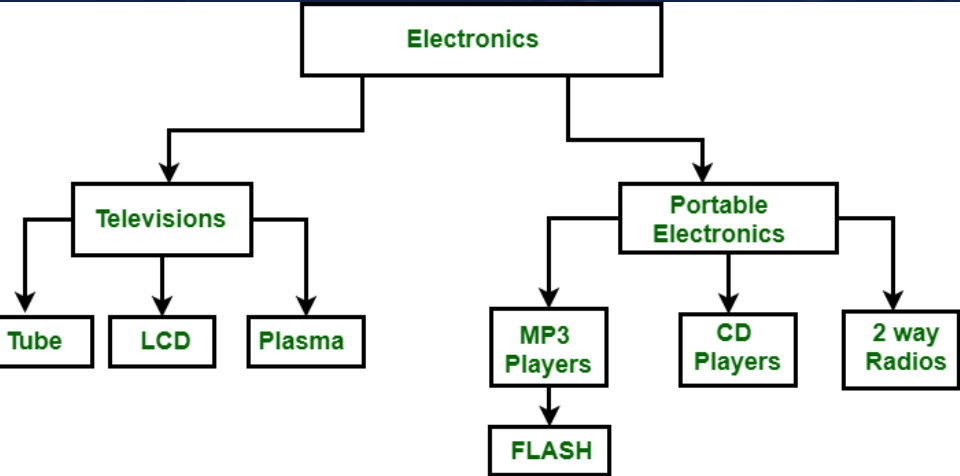
```
class("ui-icon-triangle-l-e")  
if(_.isObject(params)&&  
click(function(e){s(this).rel
```



Hiyerarşik Veri Tabanları

Bu yapı, bilinen en eski veri modelleme sistemidir. Kullandığı veri modeli ağaç yapısıdır. Ebeveyn ve çocuk ilişkisine dayalı bir sistemdir. Hiyerarşim modelde veri türleri üstten aşağıya doğru bir usulde düzenlenir. Bu modelde çoklu ilişkileri temsil edebilmek için, varlık tiplerinin her ilişki için ayrı ayrı tanımlanması gerekir. Bu da gereksiz veri tekrarına sebep olur. Bu modelde, veriler ağaç yapısına benzer bir biçimde modellenir. En üste kök ve kökün dalları bulunur. Ayrıca her dalın alt dalı sayesinde dallanma ve çeşitlilik artar. Bu modelde her bir alt dalın sadece bir tane noktadan bağlanma şartı bu modelin en büyük kısıtlamalarından biridir.

```
class("ui-icon-triangle-l-e")  
if(_.isObject(params)&&  
click(function(e){s(this),e
```



Hiyerarşik Veri Tabanları

Bir ağaç yapısına sahip olması nedeniyle hiyerarşik olarak adlandırılmıştır. Ağaç yapısı şekilde olan bu veritabanında bir merkezden başlayan hiyerarşik yapı birçok alt kırılımlar ile sayısını artırır. Sayı arttıkça da ciddi miktarda problemler bizi karşılar. Alt gruplar, kırılımlar arttıkça da tablolar bir yerden sonra yönetilemez hale geleceklerdir.

Veri Modeline Göre Veri Tabanı Yapıları

Temel veri tabanı yapıları şunlardır;

1- Hiyerarşik
Veri Tabanları

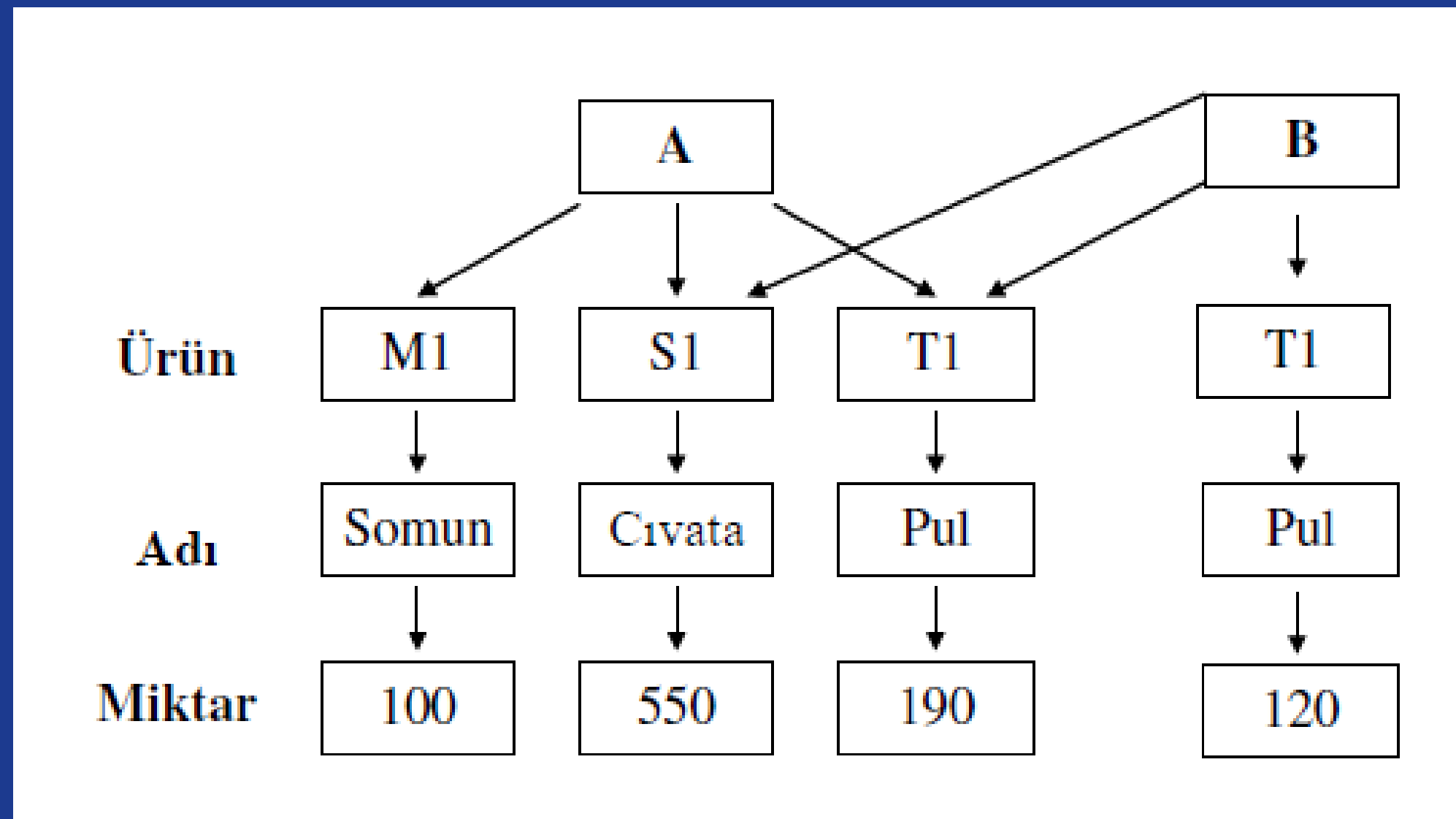
İlişkisel veri
tabanı

2- Ağ Veri
Tabanları

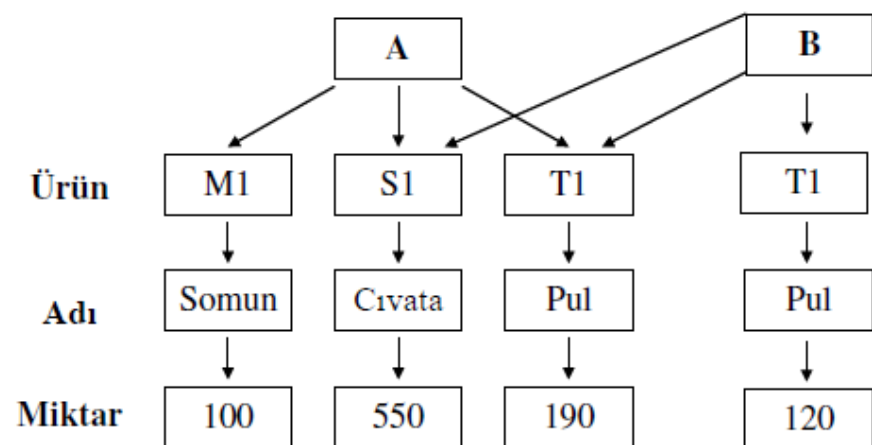
Nesne yönelimli
veri tabanı



2- Ağ Veri Tabanları



2- Ağ Veri Tabanları



1960'lı yılların sonunda yapılan bir konferansın ardından toplanan veritabanı çalışma grubu hiyerarşik modeldeki bazı eksiklikleri gidermek adına bu ağ modelini geliştirmiştir. Bilinen en karmaşık veri tabanı metodudur. Bu yapı daha çok karmaşık bağlantılara izin verir. Veri türleri arasında yan bağlantılarla ilişki kurulur. Burada aynı veri türlerinin paylaşımı söz konusudur (Mesela, A ve B müşterileri Şekil de görüldüğü gibi S1 ve T1 'i paylaşmaktadır).

Veri Modeline Göre Veri Tabanı Yapıları

Temel veri tabanı yapıları şunlardır;

1- Hiyerarşik
Veri Tabanları

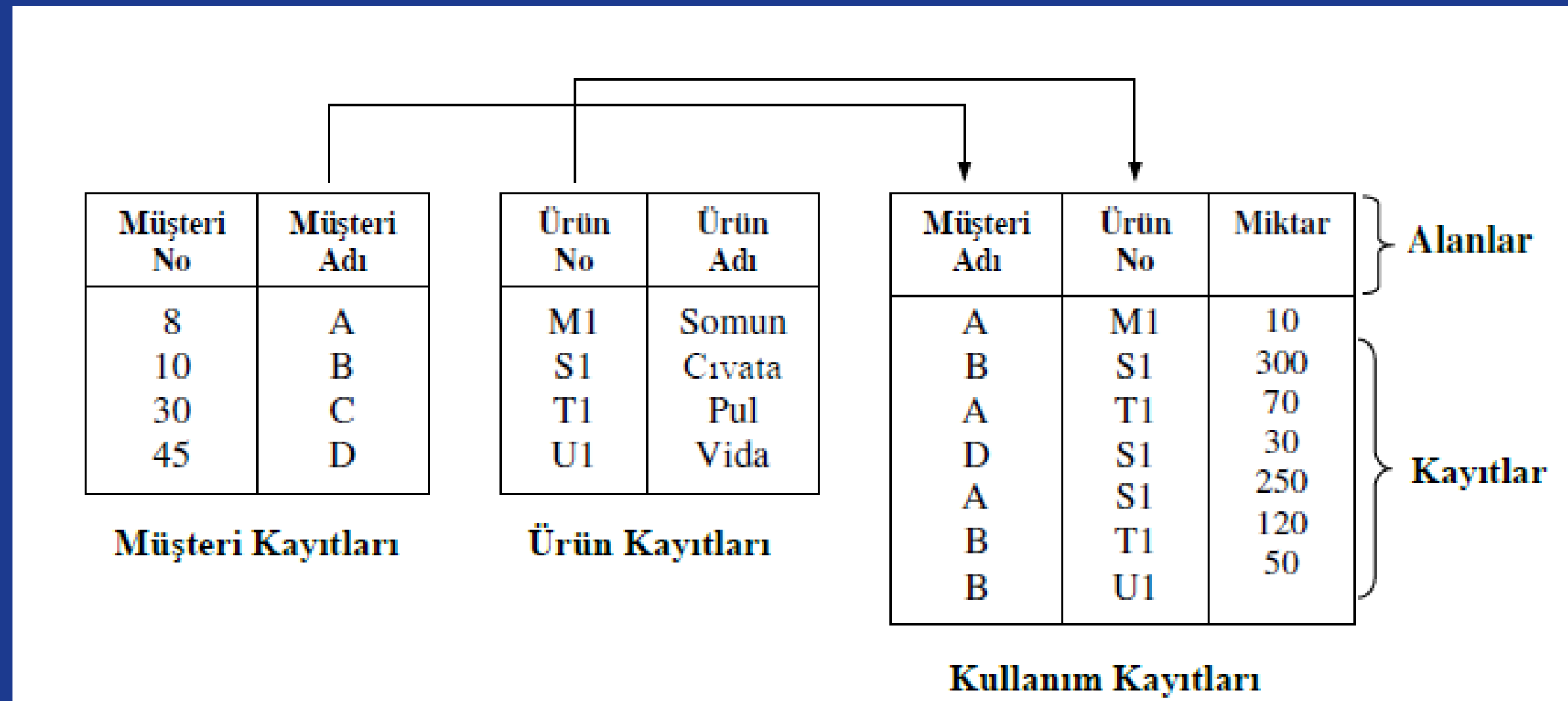
3- İlişkisel Veri
Tabanı Sistemi

2- Ağ Veri
Tabanları

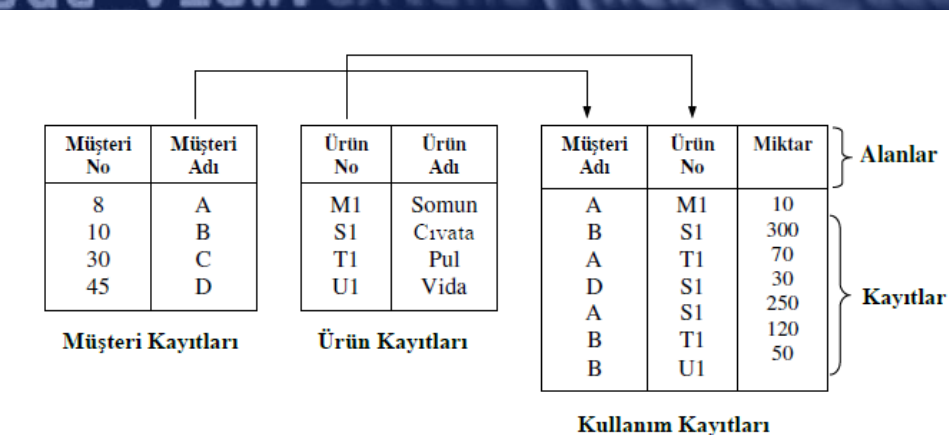
Nesne yönelimli
veri tabanı



3- İlişkisel Veri Tabanı Sistemi



3- İlişkisel Veri Tabanı Sistemi



Bu modelde temel alınan şeyler verilerin birbiriyle ilişkileridir. 1970'li yılların başında geliştirilmiştir. Günümüzde en çok kullanılan modeldir. PC'lerden sunuculara kadar yayılmış bir yelpazeye sahiptir ve bu mimari kullanılarak birçok yazılım üretilmiştir. İlişkisel veri modelinde veriler basit tablolar halinde tutulur. Tablolar, satır ve sütunlardan oluşur.

Veri Modeline Göre Veri Tabanı Yapıları

Temel veri tabanı yapıları şunlardır;

1- Hiyerarşik
Veri Tabanları

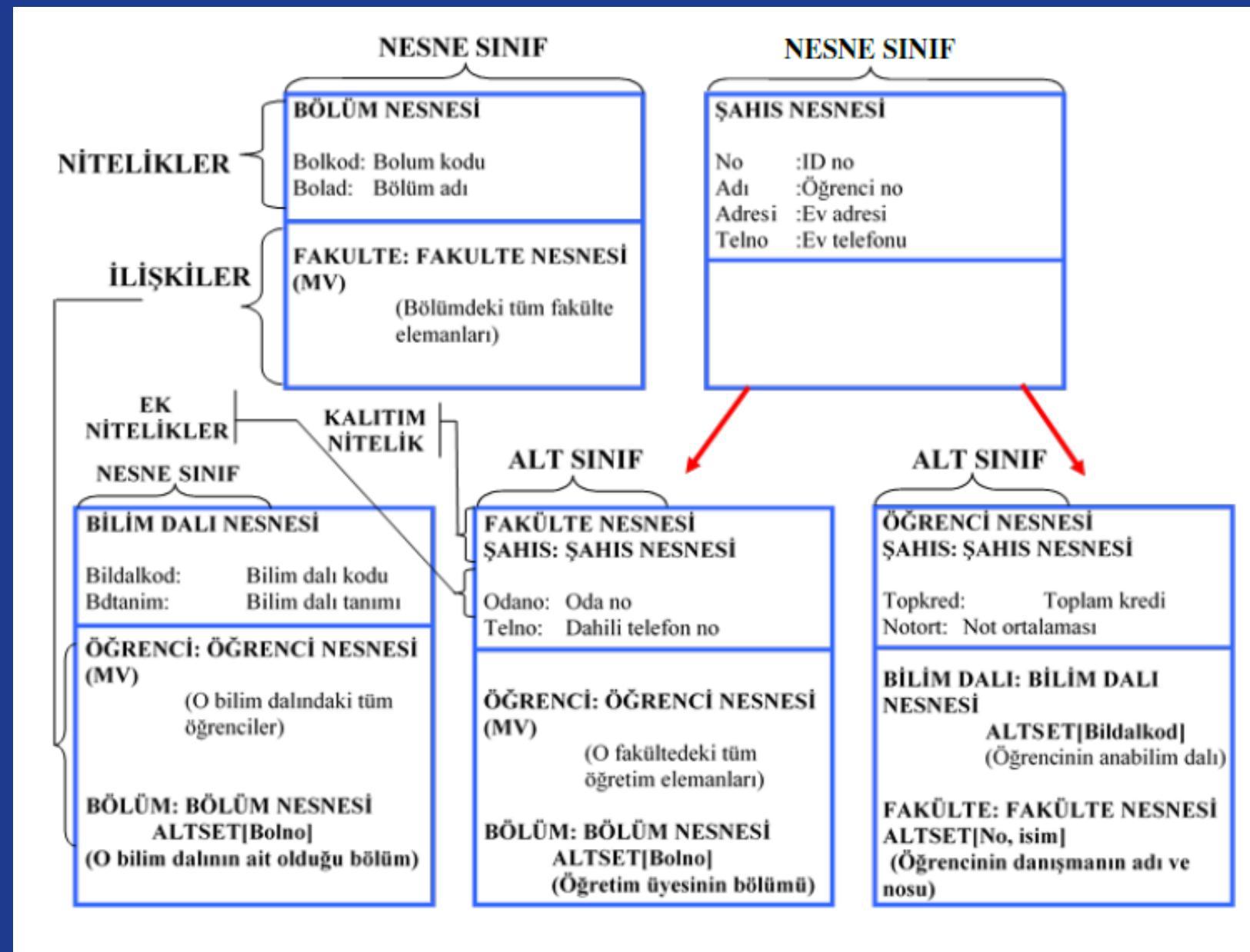
3- İlişkisel Veri
Tabanı Sistemi

2- Ağ Veri
Tabanları

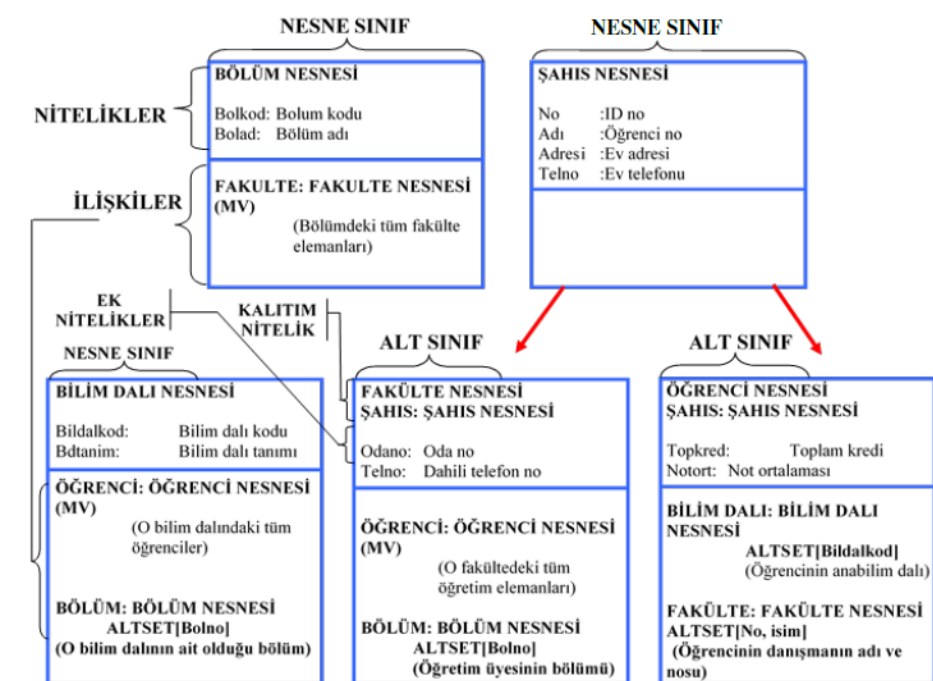
4- Nesneye Yönelik
Veri Tabanları



4- Nesneye Yönelik Veri Tabanları



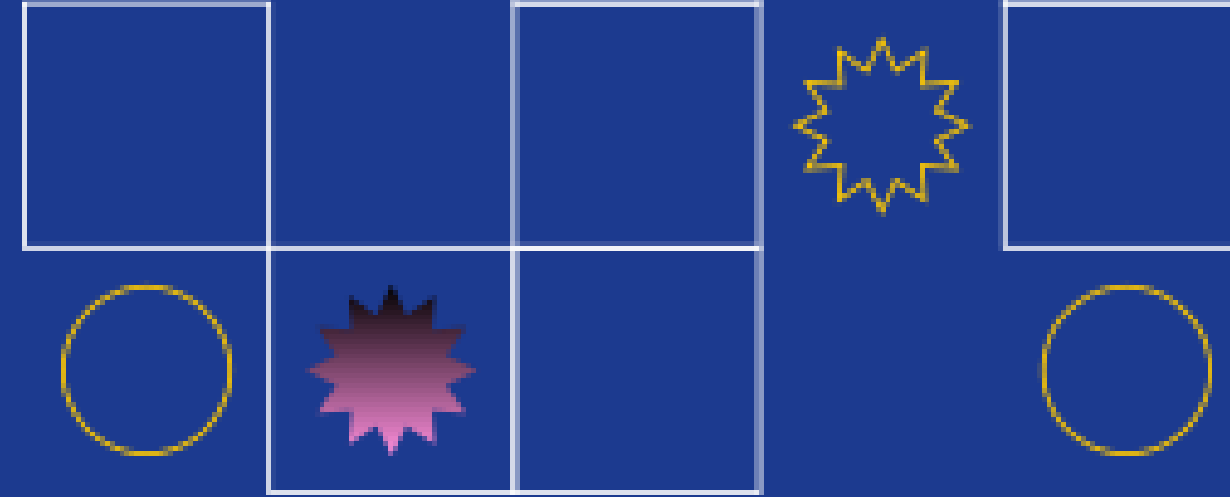
4- Nesneye Yönelik Veri Tabanları



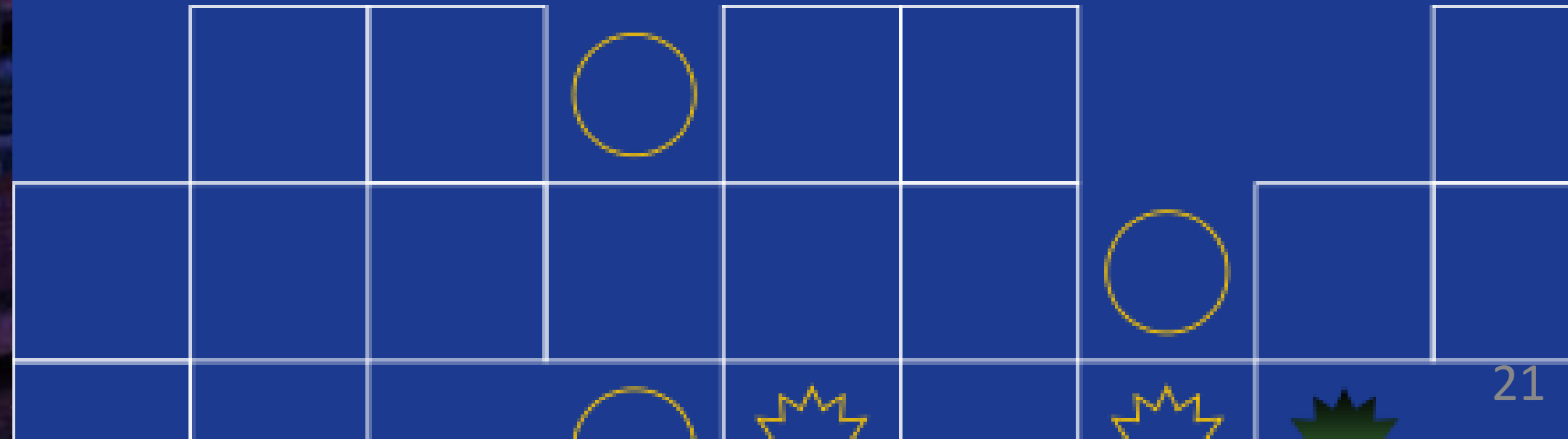
Günümüzde nesne kavramı her yerde kullanılmaktadır. Pek çok kelime işlemci ve hesap tablosu programlarının alıştığımız görünümüne artık bir de nesnelere eklenmiştir. Ancak bu gerçek anlamda bir nesneye yönelik yazılım demek değildir. Yüzde yüz nesneye yönelik bir yazılımın tamamen nesneye temelli çalışması gerekir.

Günümüzde hem ilişkisel hem de nesneye yönelik yaklaşımı birlikte kullanan VTYS'lerinin yaygınlaştığı görülmektedir.

Nesne yönelimli programlamaya dayanan veri modelidir.



Varlık İlişki Diyagramları



Varlık İlişki Diyagramları

İlişkisel veritabanı tasarımında en yaygın kullanılan yapısal modelleme aracı, Varlık-İlişki Diyagramları (Entity-Relationship – E-R)'dir.

Bu diyagramlar, veritabanındaki varlıkları (entity) ve bu varlıklar arasındaki ilişkileri (relationship) grafiksel olarak gösterir. Aynı zamanda, veritabanı tasarım sürecinde bir referans dokümanı olarak da kullanılabilirler.

Varlık İlişki Diyagramları

Veritabanı tasarımındaki temel amaç;

01



Veri tekrarını en aza
indirmek

02



Veri tutarlılığını
(doğruluğunu ve
bütünlüğünü)

Varlık İlişki Diyagramları

Tasarım süreci genellikle şu adımlarla ilerler:

İhtiyaç Analizi:

Veritabanı geliştirilecek kurum ya da kuruluşun hangi bilgilere ihtiyaç duyduğu belirlenir. Bu aşamada tasarımcı, doğrudan sistem kullanıcılarıyla görüşerek veri ihtiyaçlarını anlamaya çalışır.

Veri Toplama ve Sistem Analizi:

Kullanıcı ihtiyaçları doğrultusunda veriler toplanır ve sistemin genel yapısı analiz edilir.

Varlık - İlişki Diyagramının Oluşturulması:

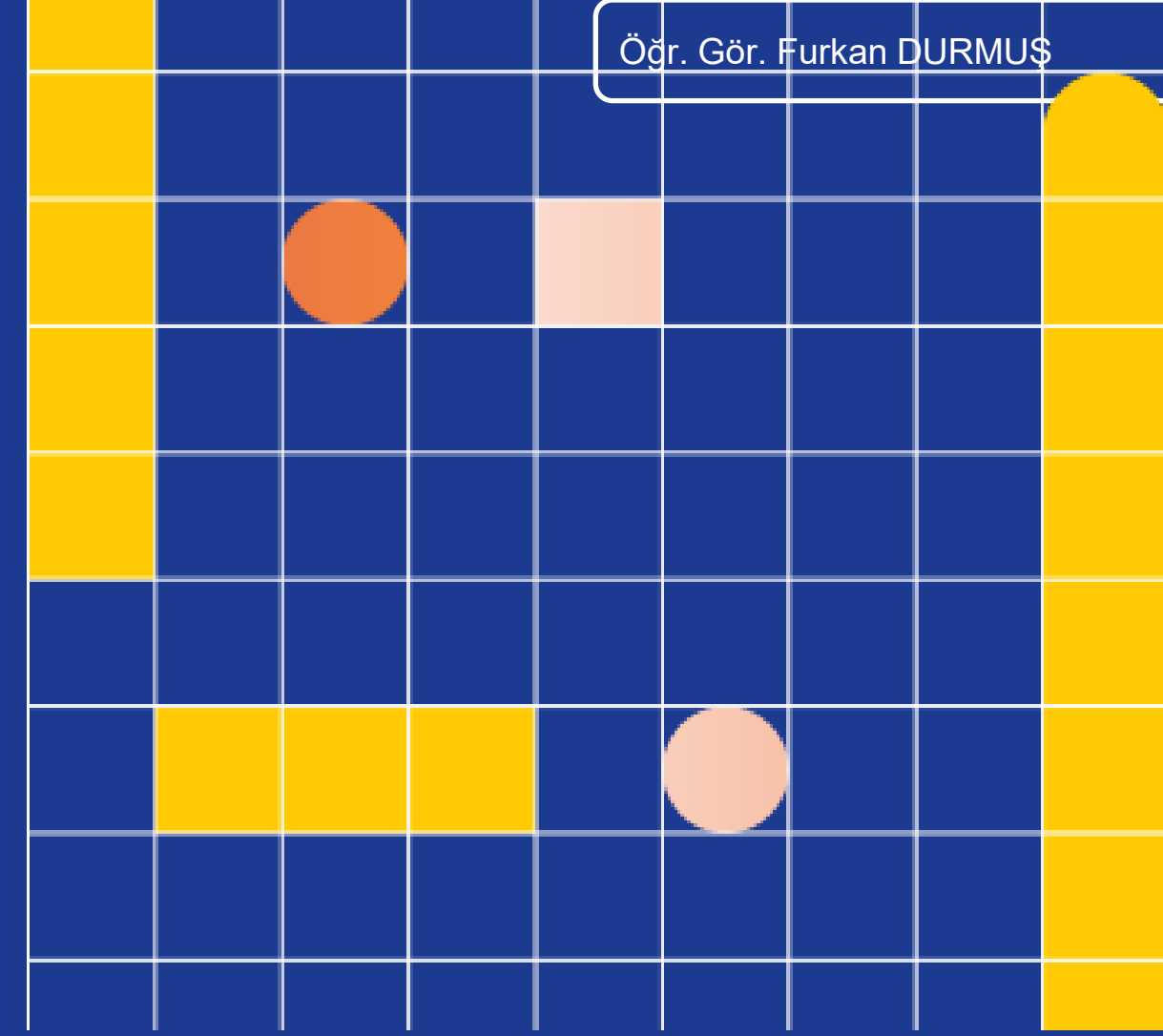
Toplanan verilere göre, ilişkisel model temel alınarak bir veritabanı şeması tasarlanır. Bu şema; Varlık tiplerini, Bu varlıkların özelliklerini (niteliklerini), Ve aralarındaki ilişkileri tanımlar.

Tabloların Oluşturulması:

Varlık - İlişki diyagramına dayanarak, her varlık için uygun tablolar tasarlanır ve veritabanı yapısı somut olarak oluşturulur.

Varlık İlişki Diyagramları

Özetle, ilişkisel veritabanı tasarımında, Varlık - İlişki diyagramları kullanılarak varlıklar ve aralarındaki ilişkiler belirlenir. Bu diyagramlar, veritabanı yapısını planlamak ve kullanıcı ihtiyaçlarına göre düzenlemek için önemli bir araçtır. Amaç, tekrarlardan arındırılmış, tutarlı ve anlaşılır bir veri yapısı oluşturmaktır.



```
11 </head>
12 <body>
13   <button type="button" class="btn btn-default" data-toggle="tooltip" data-
    placement="left" title="Tooltip on left">Tooltip on left</button>
14   <button type="button" class="btn btn-default" data-toggle="tooltip" data-
    placement="top" title="Tooltip on top">Tooltip on top</button>
15   <button type="button" class="btn btn-default" data-toggle="tooltip" data-
    placement="bottom" title="Tooltip on bottom">Tooltip on bottom</button>
16   <button type="button" class="btn btn-default" data-toggle="tooltip" data-
    placement="right" title="Tooltip on right">Tooltip on right</button>
17
18 <!-- Includes jQuery (OK for bootstrap)
19 <script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
20
21 <script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
22 </body>
```

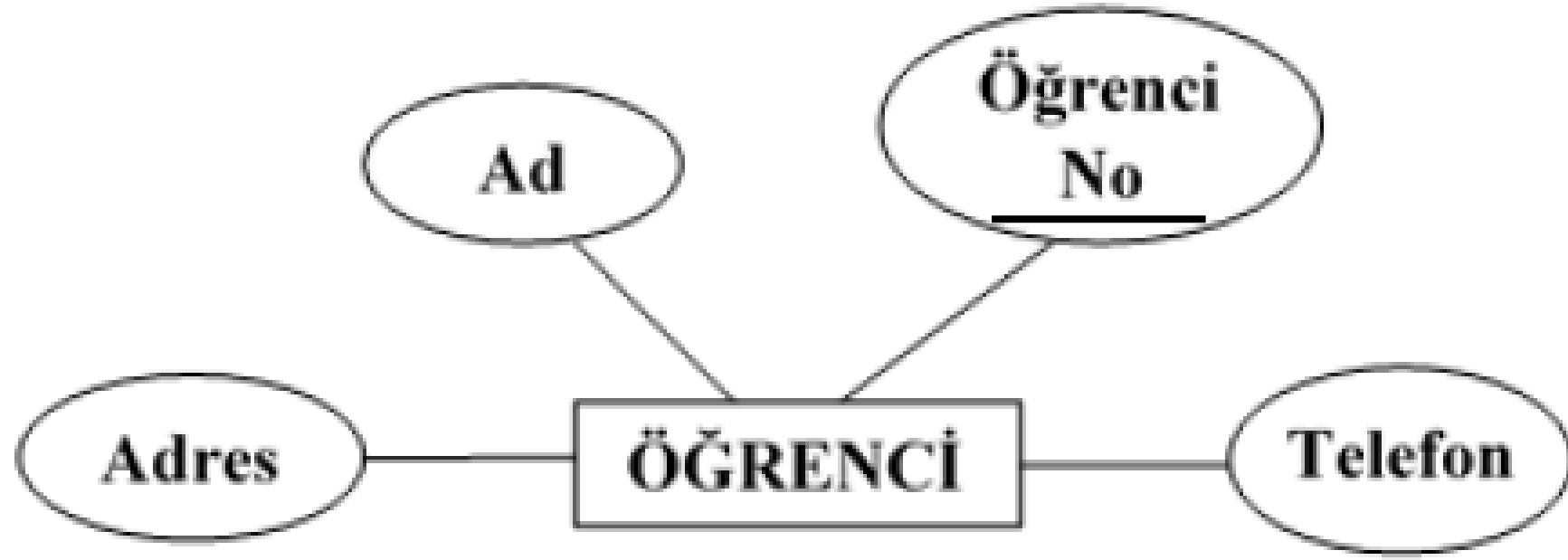
Varlık İlişki Diyagramlarının Bileşenleri

VTYS 2. Hafta

Öğr. Gör. Furkan DURMUŞ

Varlık-İlişki diyagramları oluşturulmada varlıkların ve onlarla ilgili özelliklerin belirlenmesi gereklidir. Varlıklar ve özellikler belirlendikten sonra yapılacak olan, varlıklar arasındaki ilişkilerin oluşturularak Varlık-İlişki Diyagramının yapılandırılmasıdır. Varlık İlişki Diyagramların aşağıdaki temel semboller kullanılır.



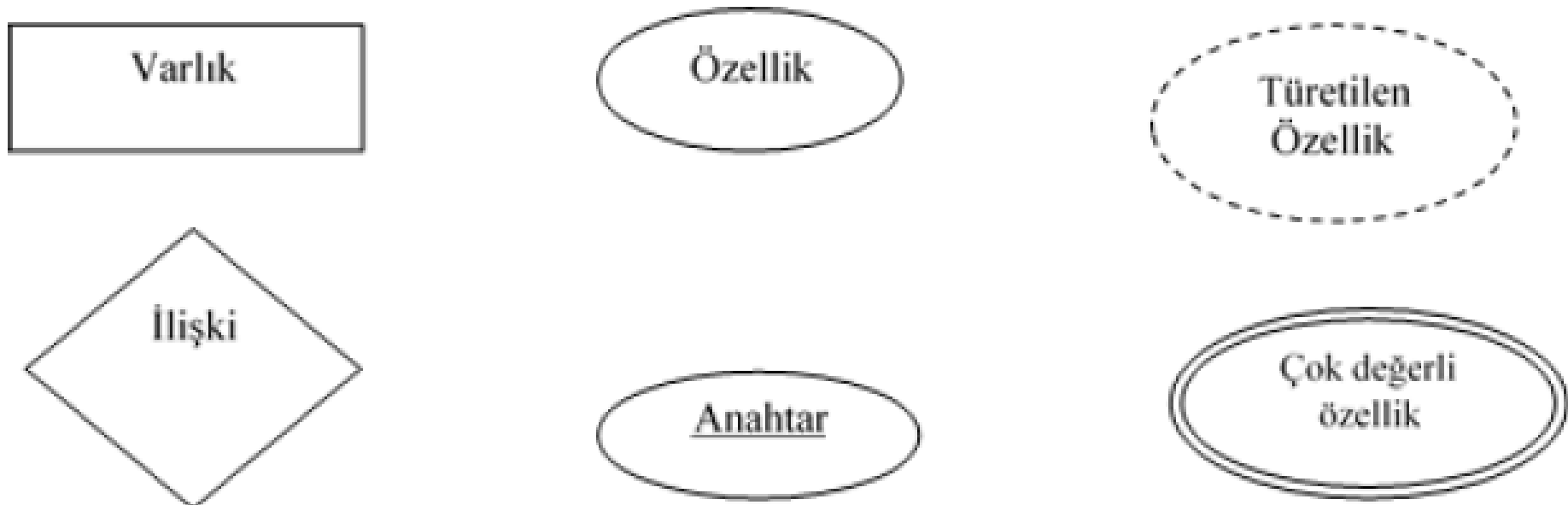


1. Varlık

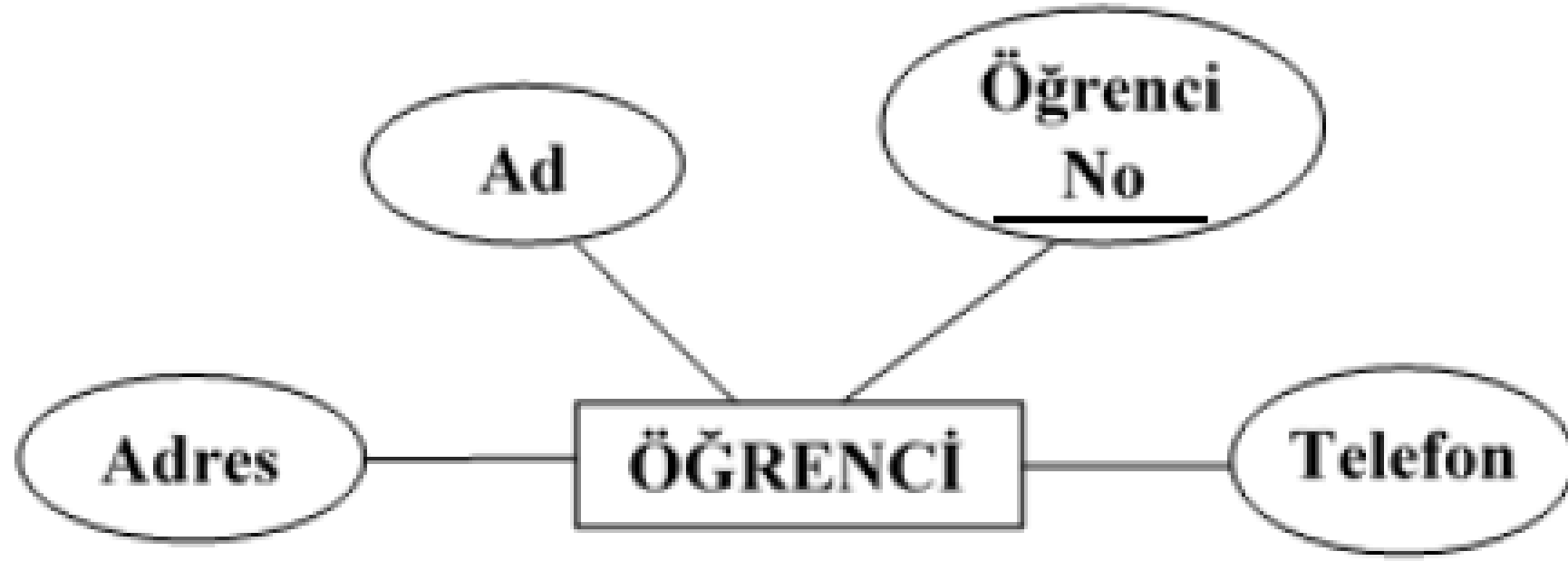
Gerçek hayatta diğerlerinden ayırt edilebilen nesnelere varlık denir.

Bir varlık, bir çevresel veri ya da bir kaynak olabilir. Bir varlık, kişi, araba, ev veya çalışan gibi fiziksel nesnelere olabileceği gibi, şirket, iş veya ders gibi fiziksel olmayan nesnelere de olabilir. Örneğin, kişi, yer, müşteri, depo, parça, ürün, ekipman, sipariş vb. hepsi birer varlıktır. Her varlık kendisini tanımlayan kendisine has özelliklere sahiptir.

Varlık kümesi, Veri tabanında benzer varlıklar ve özellik değerlerinden oluşan kümedir.



1. Varlık Örnekleri



- 1 - Ders
- 2 - Kitap
- 3 - Sipariş
- 4 - Ürün
- 5 - Müşteri

Varlık

Özellik

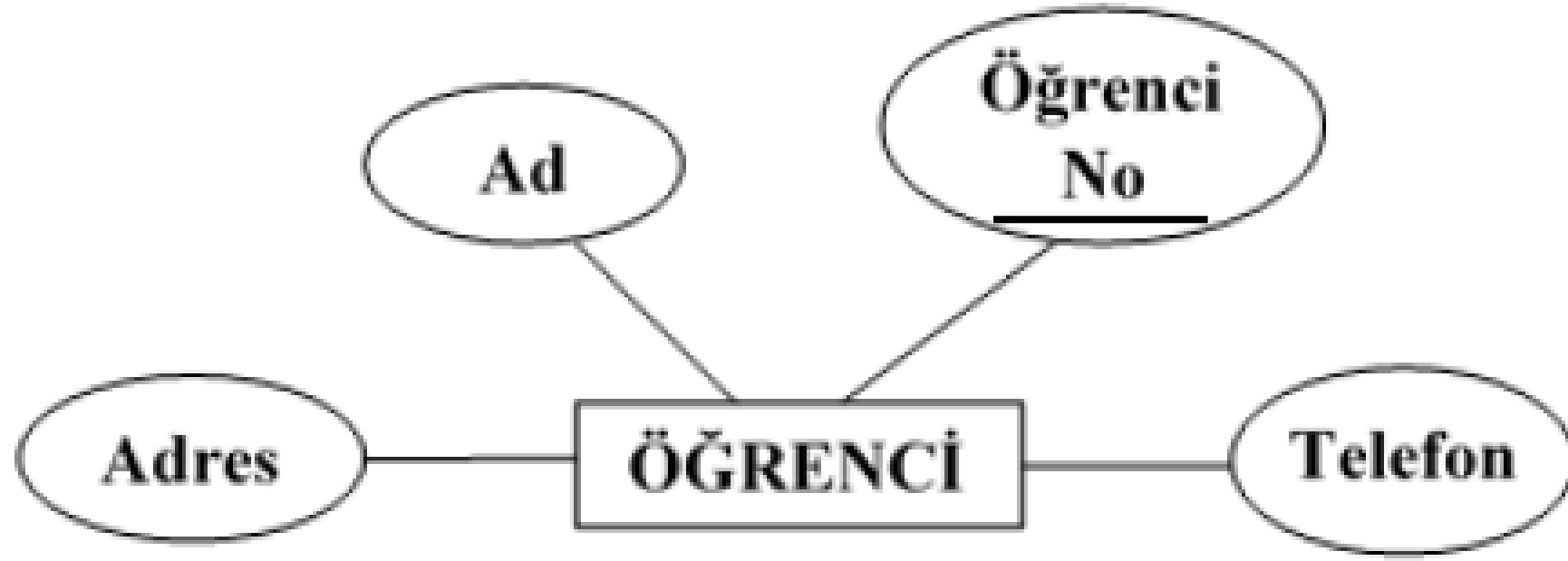
Türetilen
Özellik

İlişki

Anahtar

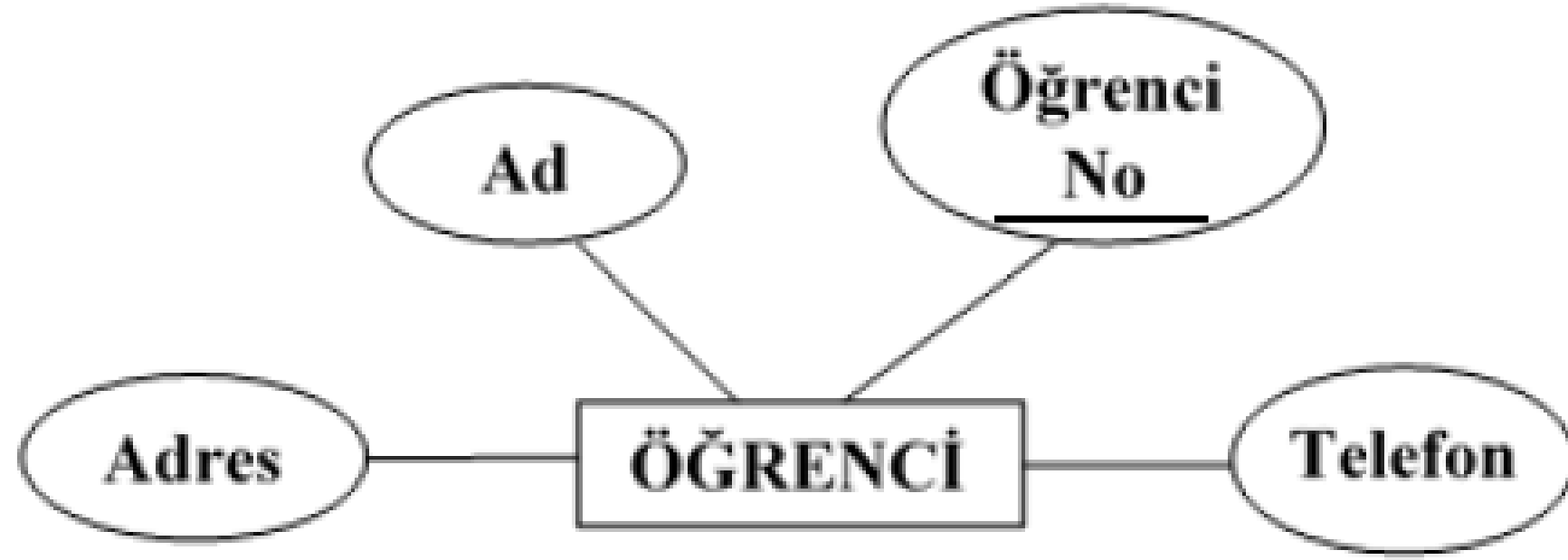
Çok değerli
özellik

2. Özellikler



Özellik, herhangi bir varlığa ait durumlardır. Örneğin müşteri no, müşteri adı, müşteri adresi vb. durumlar, müşteri varlığının özellikleri olarak ifade edilebilirler. Yada bir çalışan varlığı, çalışan adı, yaşı, adresi, geliri ve görevi özellikleri ile tanımlanır. Varlıklar Varlık-İlişki Diyagramı da dikdörtgen sembolüyle ifade edilirler. Özellikler o varlığı diğer varlıklardan ayırt etmeye yarar. Özellikler Varlık-İlişki Diyagramlarında elips şeklinde gösterilirler.





Özellik Örnekleri

- 1 - Ders
- 2 - Kitap
- 3 - Sipariş
- 4 - Ürün
- 5 - Müşteri

Varlık

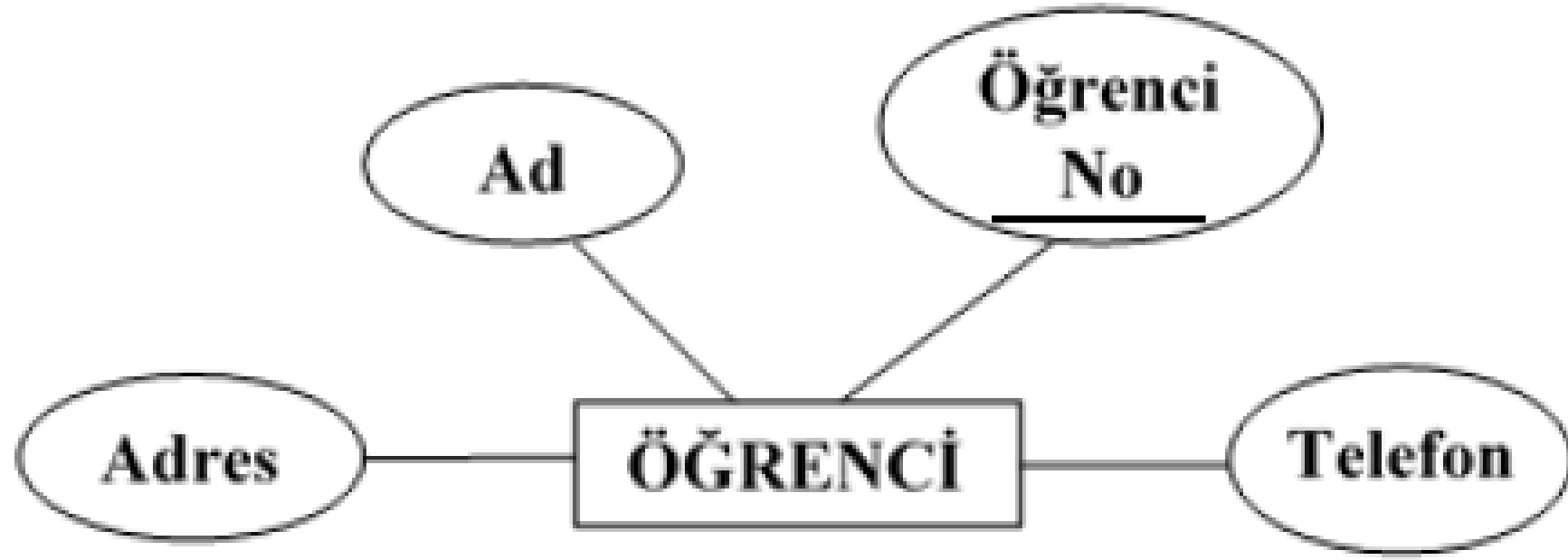
Özellik

Türetilen
Özellik

İlişki

Anahtar

Çok değerli
özellik



Özellik Örnekleri

- 1 - Ders
- 2 - Kitap
- 3 - Sipariş
- 4 - Ürün
- 5 - Müşteri

Ders Kodu

Ders Adı

Ders Kredisi

Ders
Dönemi

Ders Süresi

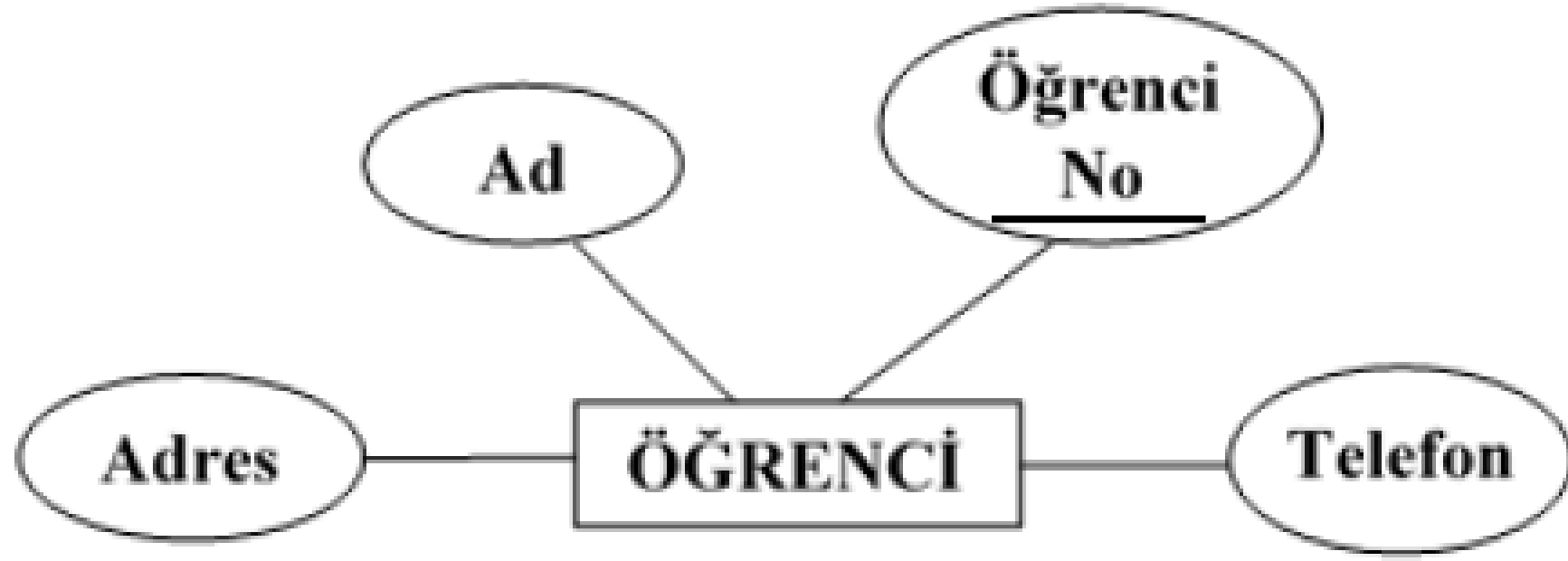
Varlık

Özellik

Türetilen
Özellik

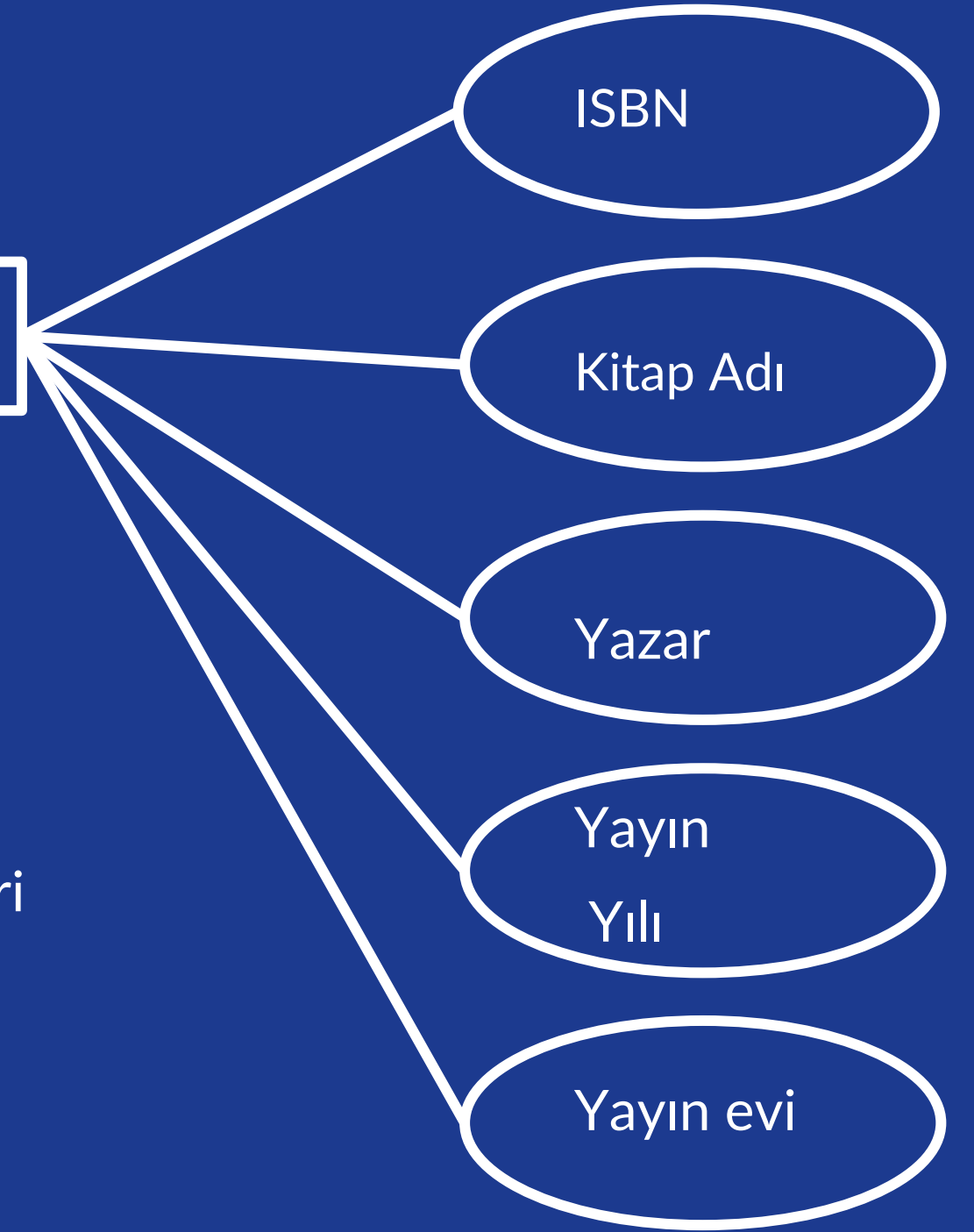
İlişki

AnahtarÇok değerli
özellik



Özellik Örnekleri

- 1 - Ders
- 2 - **Kitap**
- 3 - Sipariş
- 4 - Ürün
- 5 - Müşteri



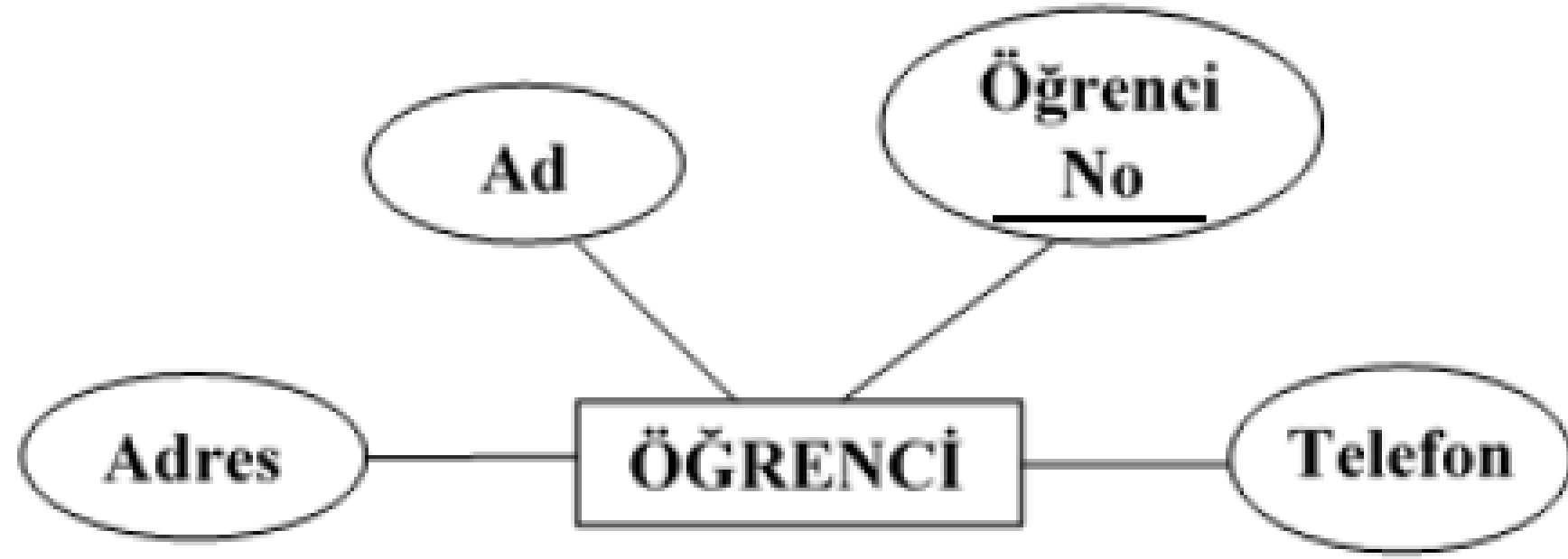
Varlık

Özellik

Türetilen
Özellik

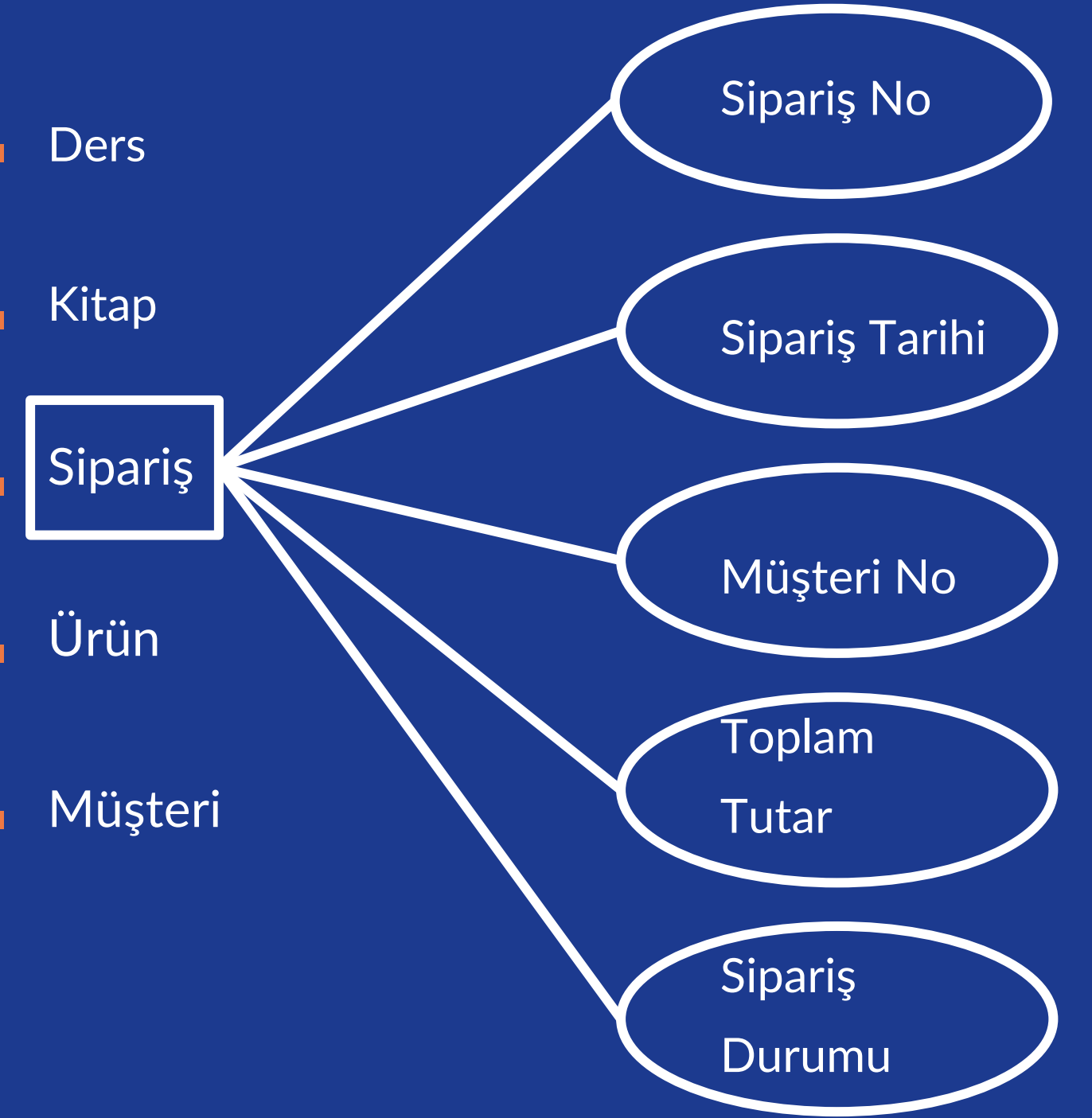
İlişki

AnahtarÇok değerli
özellik



Özellik Örnekleri

- 1 - Ders
- 2 - Kitap
- 3 - Sipariş
- 4 - Ürün
- 5 - Müşteri



Varlık

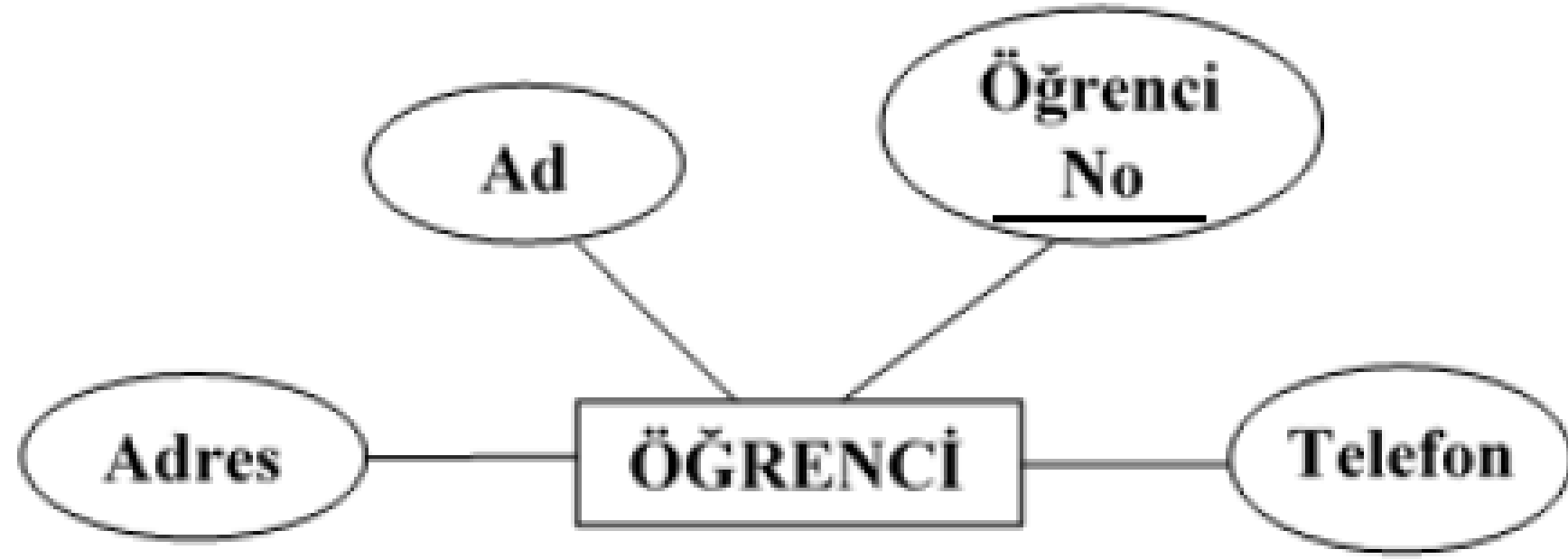
Özellik

Türetilen
Özellik

İlişki

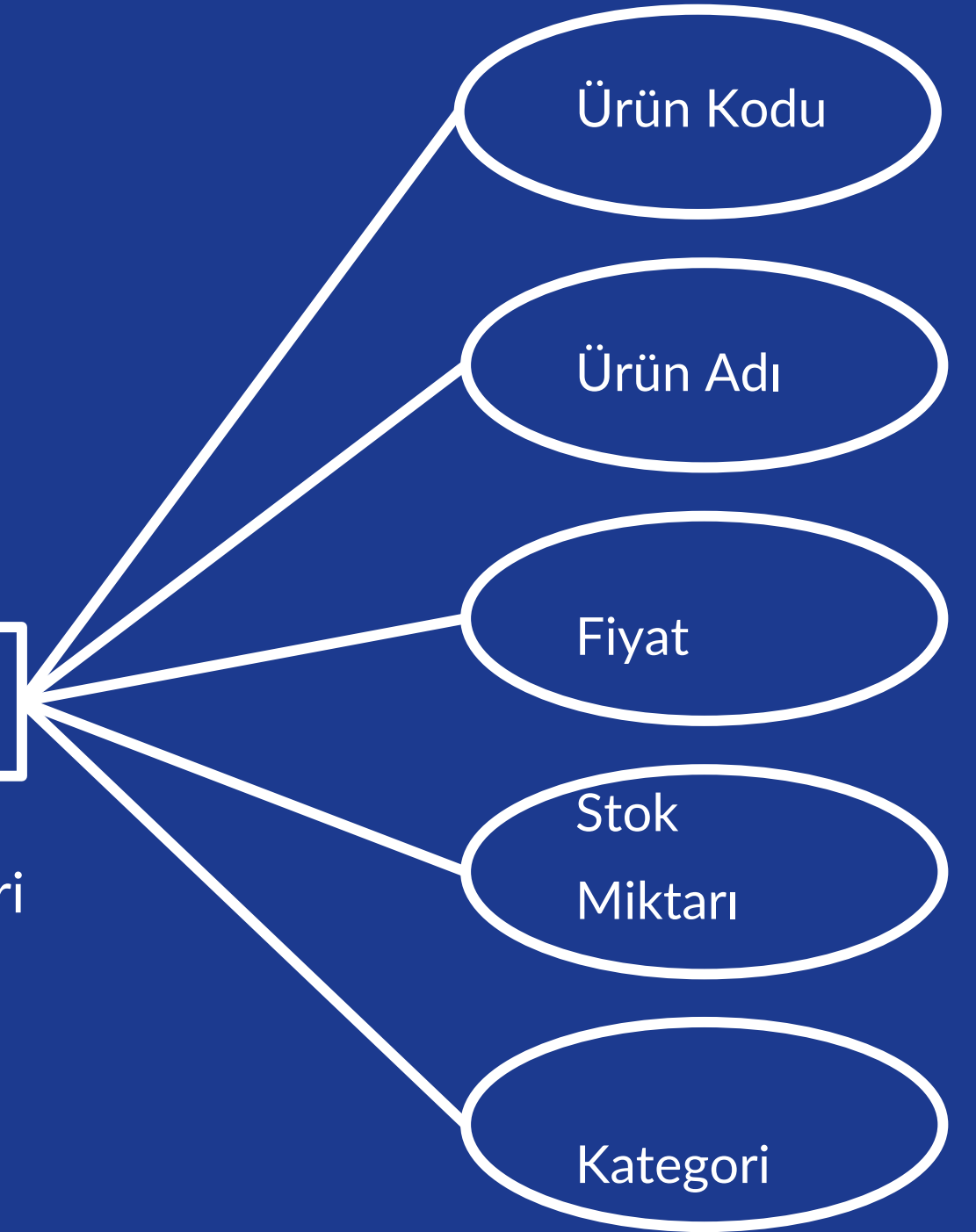
Anahtar

Çok değerli
özellik



Özellik Örnekleri

- 1 - Ders
- 2 - Kitap
- 3 - Sipariş
- 4 - **Ürün**
- 5 - Müşteri



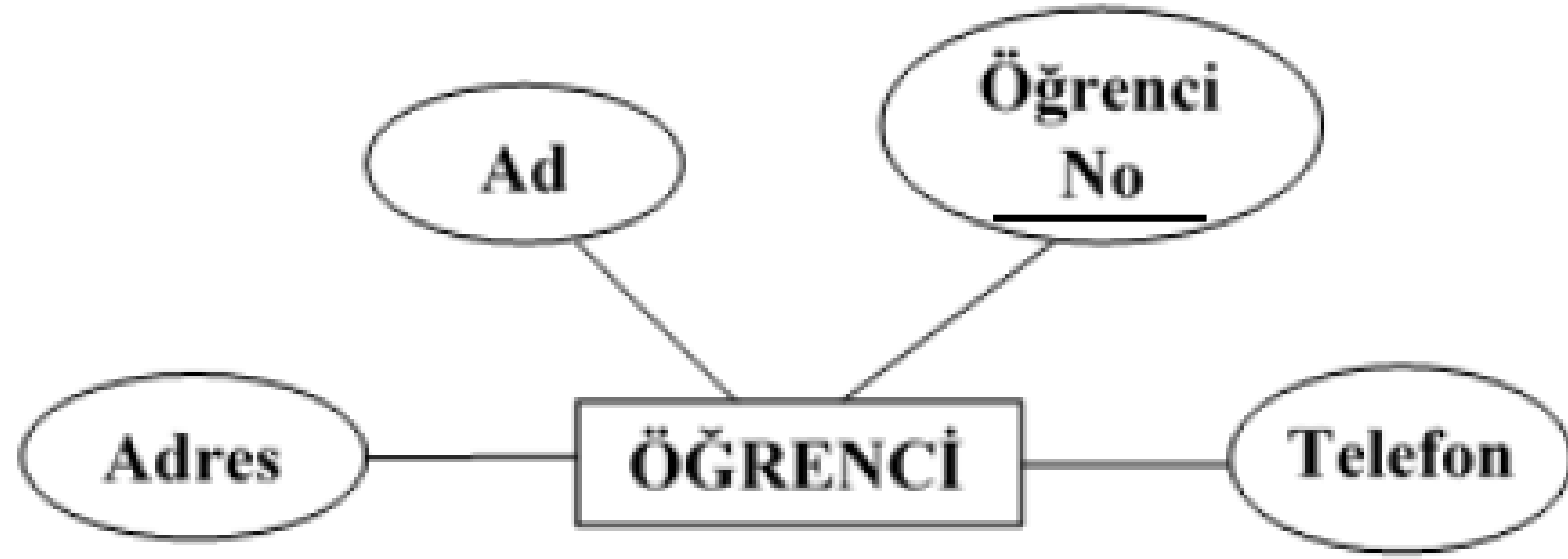
Varlık

Özellik

Türetilen
Özellik

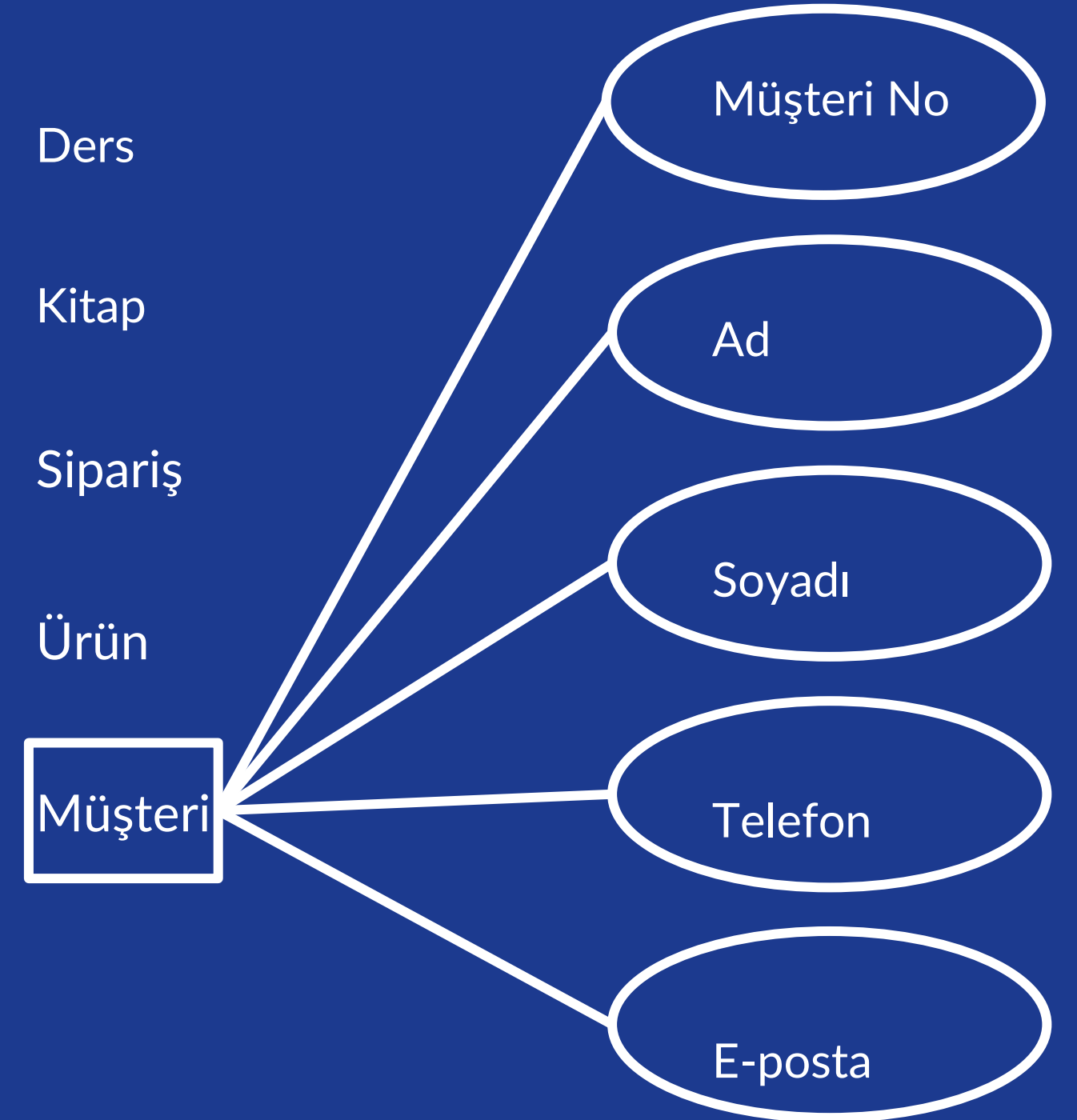
İlişki

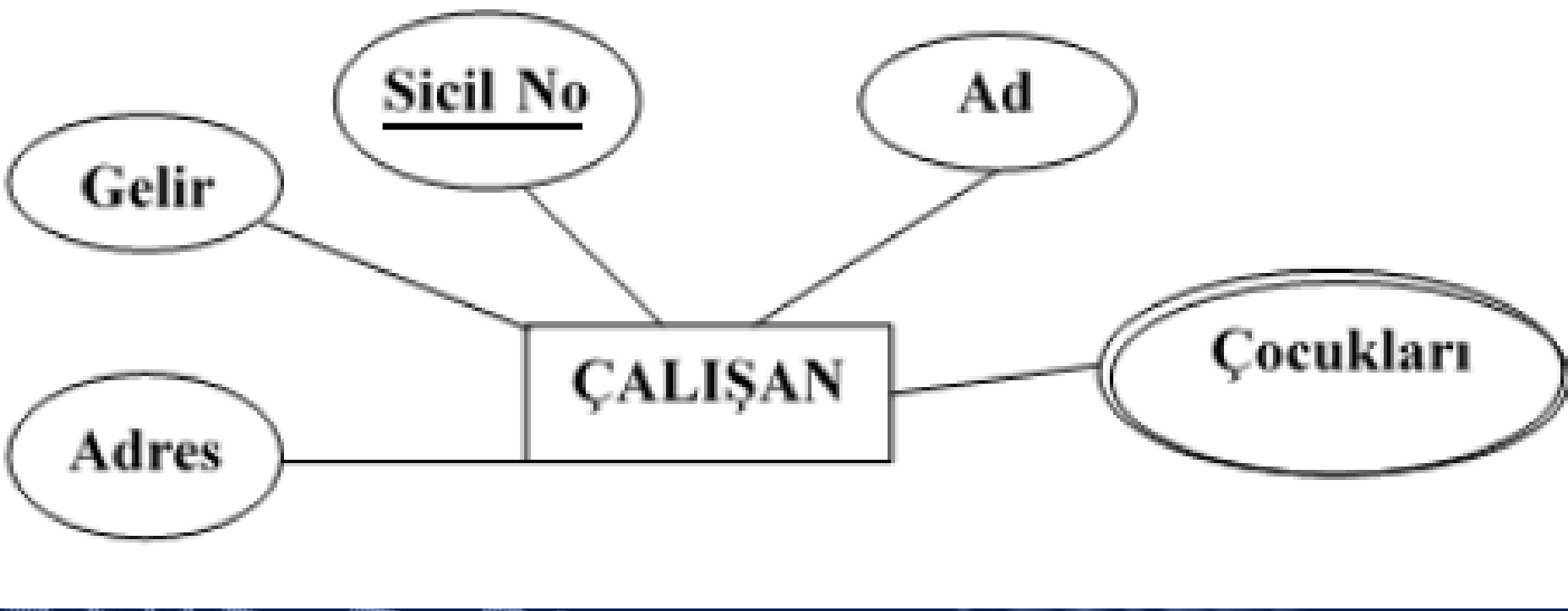
AnahtarÇok değerli
özellik



Özellik Örnekleri

- 1 - Ders
- 2 - Kitap
- 3 - Sipariş
- 4 - Ürün
- 5 - Müşteri

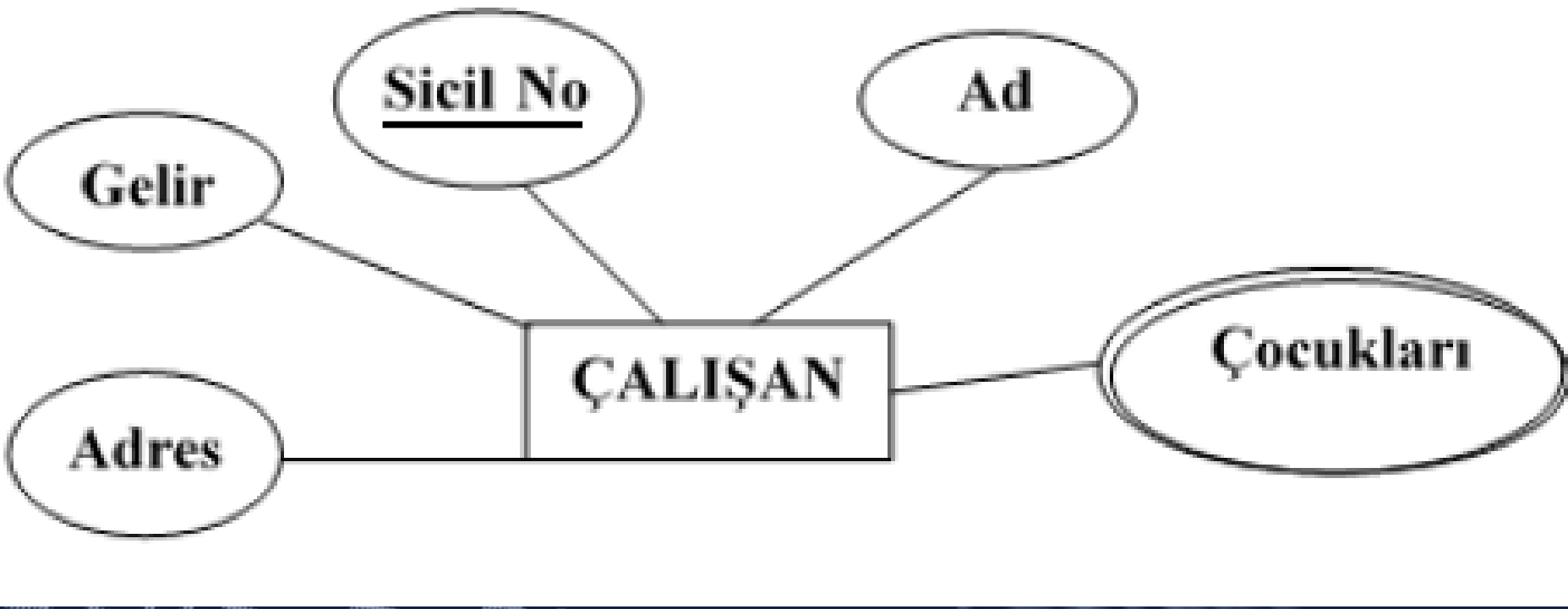




Tek Değerli Özellikler Ve Çok Değerli Özellikler

Bir varlığın sadece 1 değer alabilen niteliğine özellik denir. Örneğin bir iş yerindeki çalışanların tek sosyal sigorta numaraları vardır. Genellikle bir varlığın tek bir değeri vardır. Bir varlığın bir özelliğinin aldığı değer tek ise bu niteliğe tek değerli özellik denir. Ancak bazı özelliklerin birden çok değeri olabilir. Örneğin bir çalışanın birden fazla telefon numarası olabilir. Başka bir örnek olarak, bir çalışanın çocukları özelliği bir kişi de olabileceği gibi birden çok kişi de olabilir. Bu durumda Çalışan varlığının Çocukları özelliği çok değerli özelliktir. Varlık-İlişki şemasında çok değerli özellikler çift çizgili oval olarak gösterilir.





Çok Değerli Özellik Örnekleri

- 1 - Eposta Adresi
- 2 - Okunan Kitap
- 3 - Sipariş
- 4 - Aldığı Dersler
- 5 - Hobiler

```

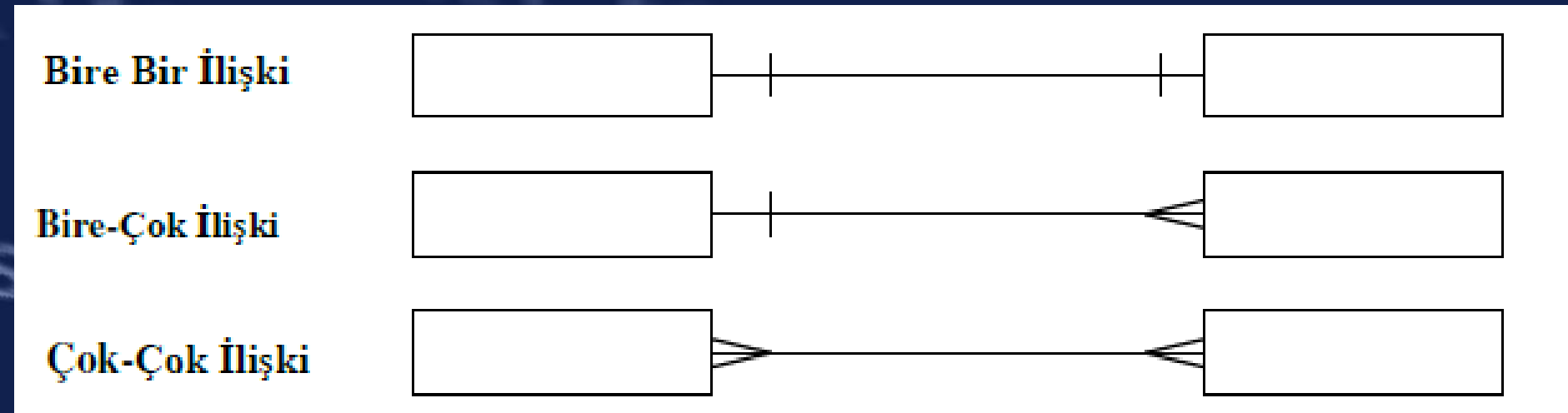
    Status(
    status_id=data.id, name=sta
    tuses[statu
  
```



3. İlişkiler

İlişki (Relationship): Varlıklar arasındaki bağlantılardır. İki veya daha fazla varlık kümesi arasında kurulan anlamlı bağlantılara ilişki denir.

Aynı varlıklar arasında farklı türden ilişkiler olabilir veya bir ilişki bir ya da daha fazla varlık arasında olabilir. Bir ilişki, iki ya da daha fazla veri varlığı arasındaki mantıksal etkileşimdir. Varlık-ilişki, hem varlıkları hem de ilişkileri göstermektedir.



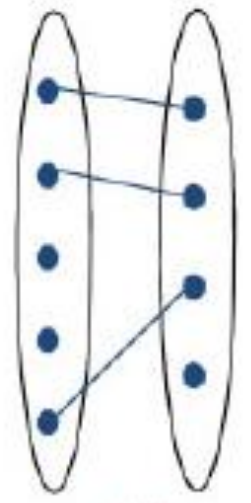
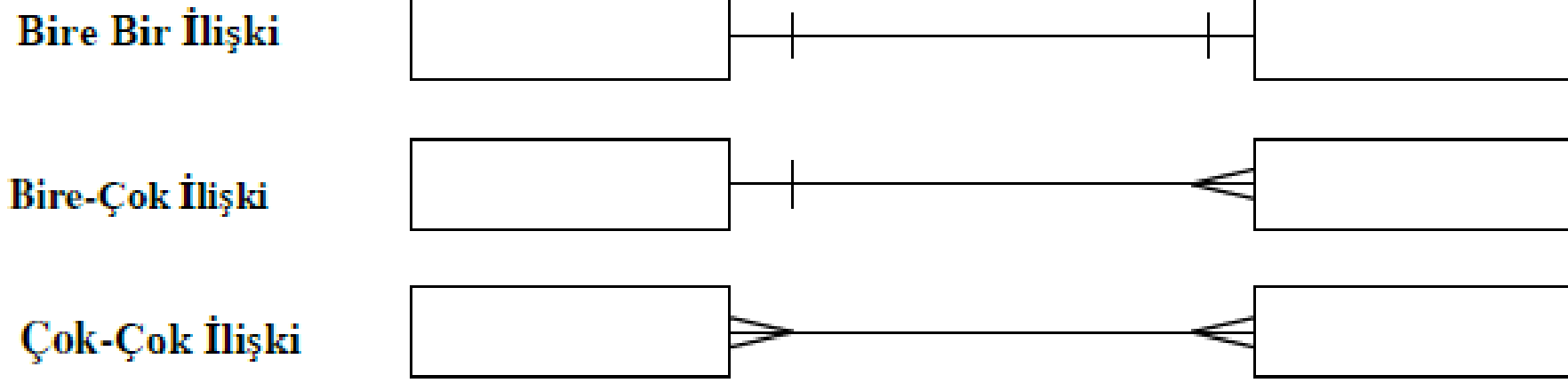
```
for data in resp_iter:
    status = Status(
        status_id=data.id, name=...
```



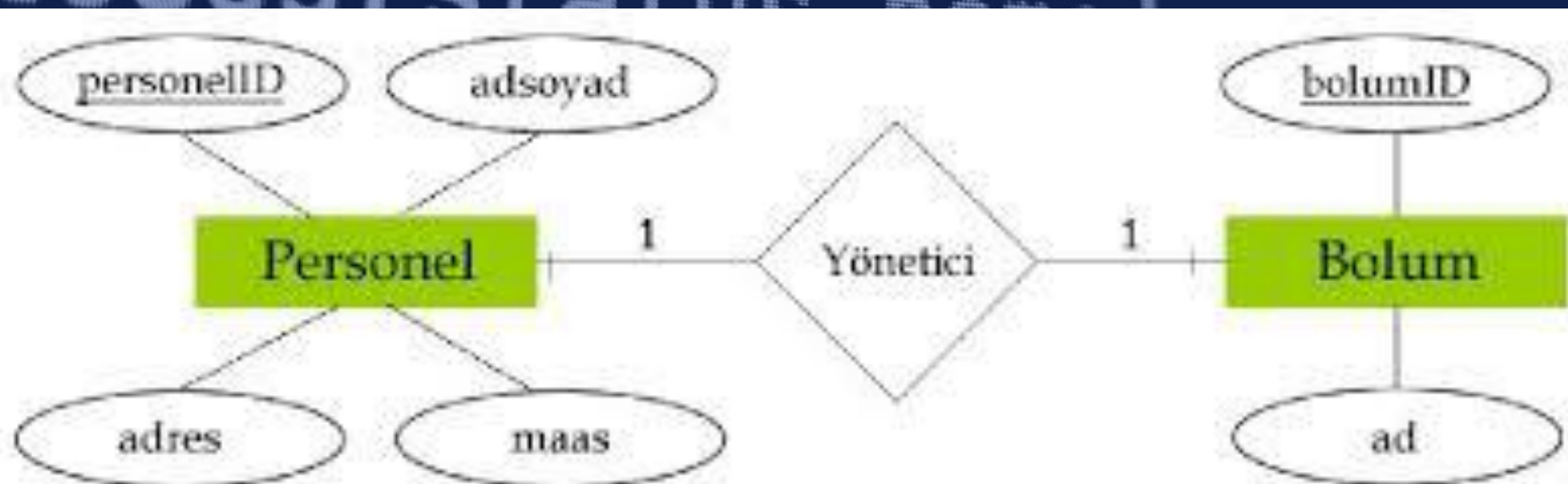
Bire-Bir (1-1) İlişki

Herhangi bir varlık kümesindeki her varlık diğer varlık kümesinin en çok bir varlığı ile ilintilidir.

ÖRN: A varlık kümesinin bir elemanı, B varlık kümesinin bir elemanı ile ilişki kurabiliyorsa bu ilişki türüne bire-bir ilişki denir.

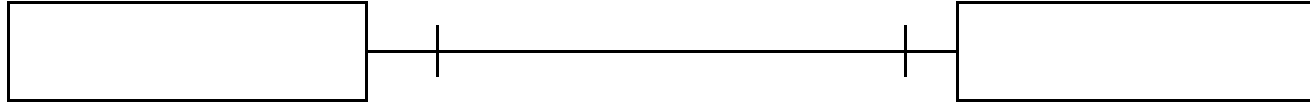


1-1

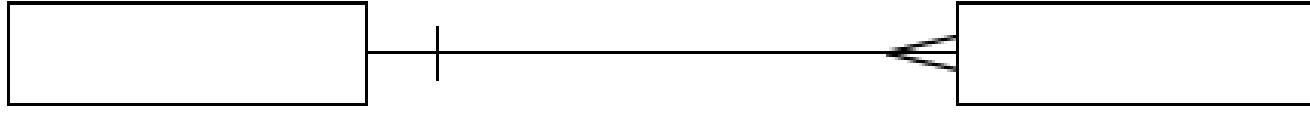


Bire-Çok (1-N) İlişki

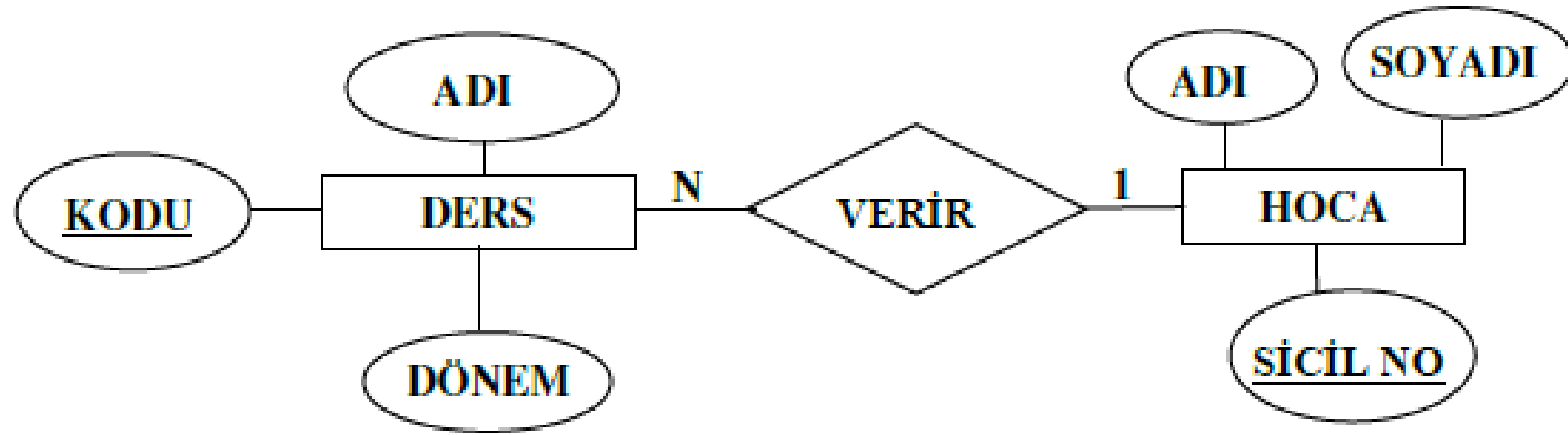
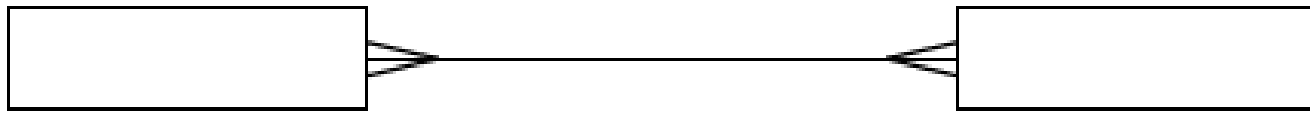
Bire Bir İlişki



Bire-Çok İlişki

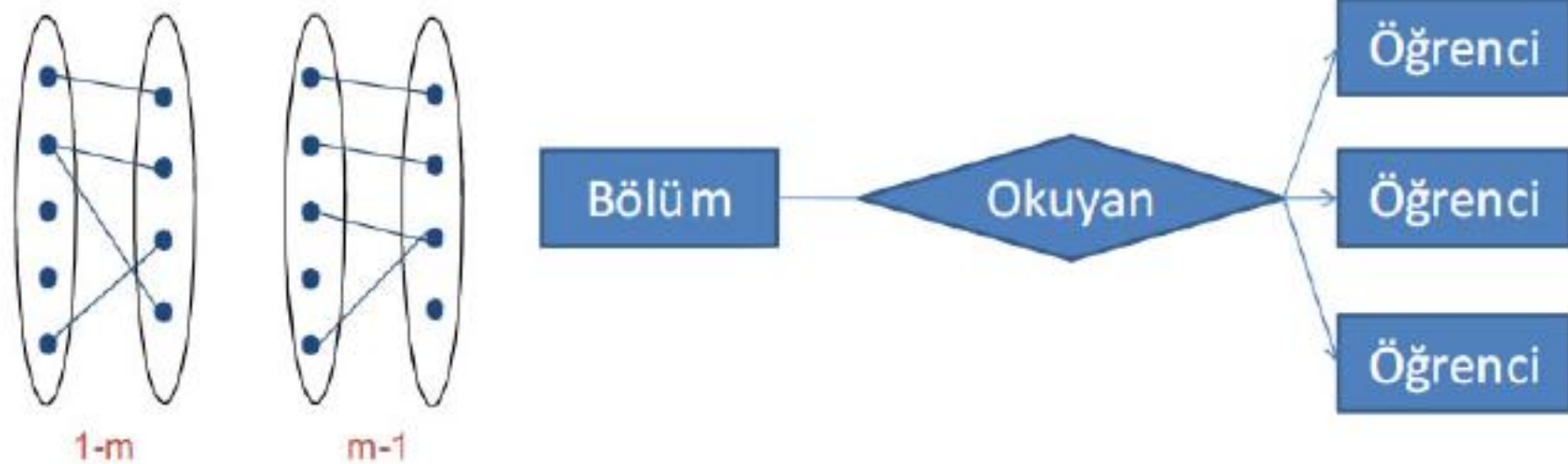


Çok-Çok İlişki



İlk kümedeki her varlık diğer kümenin en çok bir varlığına ilintidir.
En çok kullanılan ilişki şeklidir.

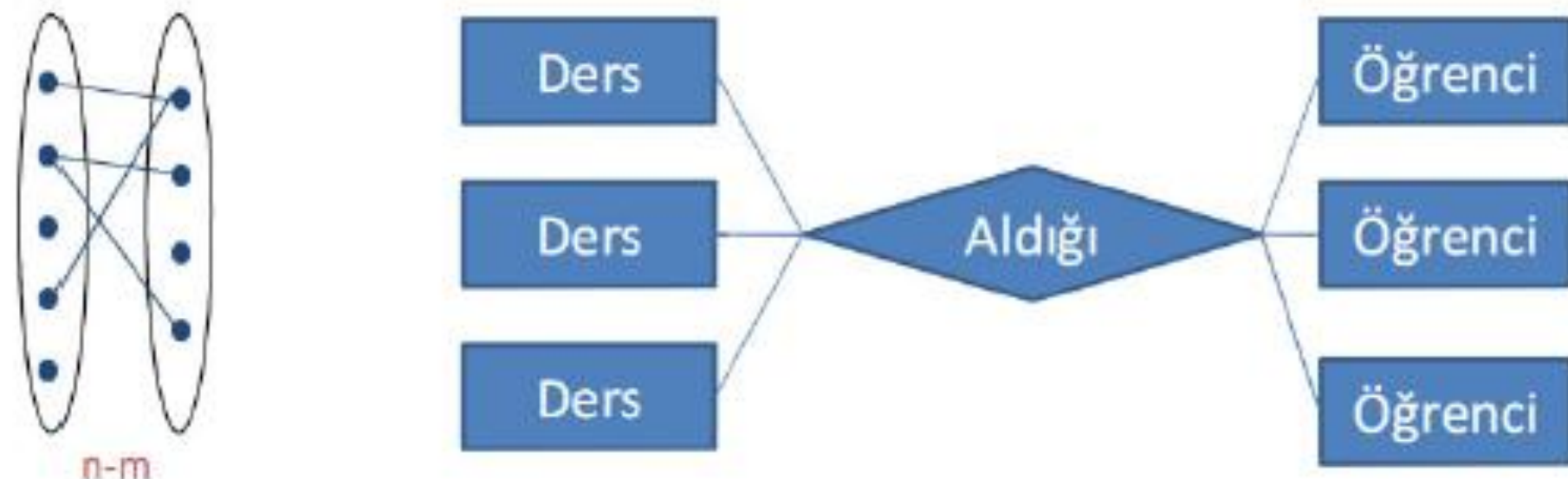
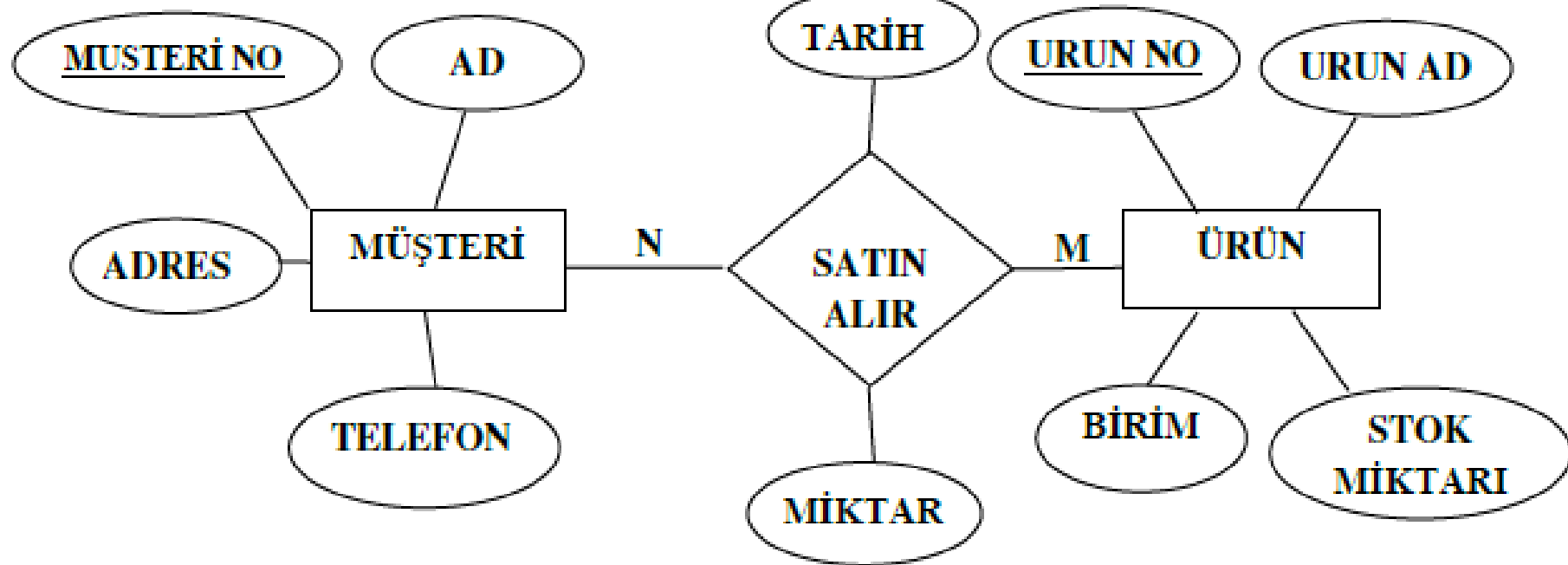
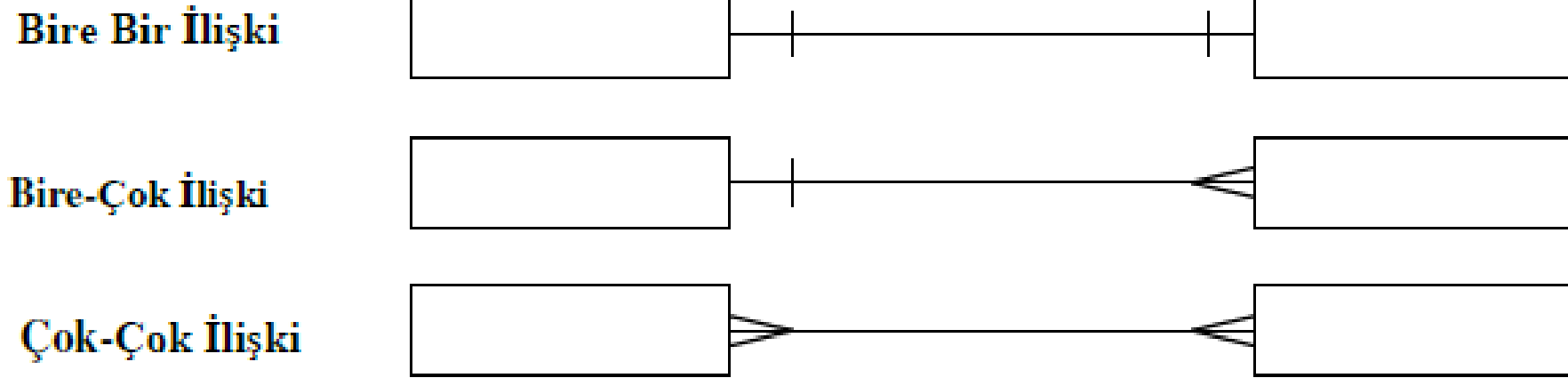
ÖRN: A varlık kümesinin bir elemanı B varlık kümesinin birden çok elemanı ile ilişki kurabiliyorsa bu ilişki bire-çok ilişkidir.



Çok-Çok İlişki (N-N yada N-M)

Bir ilişkide herhangi bir kümede bulunan varlıklardan her biri diğer kümede bulunan birçok varlıkla ilintilidir.

ÖRN: A varlık kümesinin bir elemanı, B varlık kümesinin sıfır ya da birçok eleman ile bu ilişki türü çoğa çok ilişki türüdür.



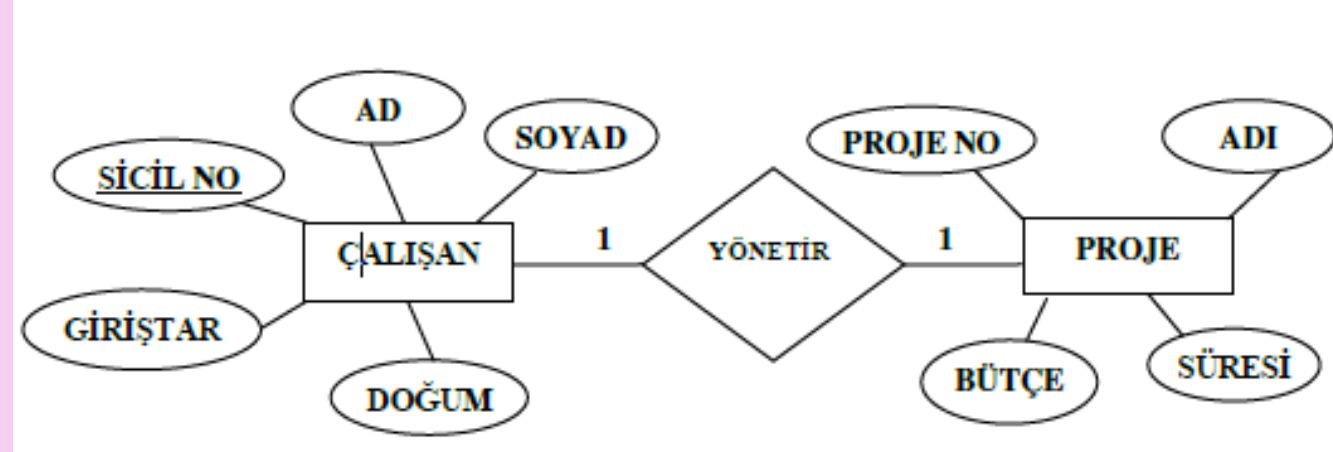
Varlık-İlişki Şemasının Tablolara Dönüştürülmesi

Not: Varlık-İlişki şemasının tablolara dönüştürme ya birinci anahtar alanı diğer tablonun içerisine koyarak ya da ikinci tablonun anahtar alanını birinci tablonun içerisine koyarak yapılır.

Bire-Bir İlişkilerin Dönüştürülmesi

VTYS 2. Hafta

Öğr. Gör. Furkan DURMUŞ



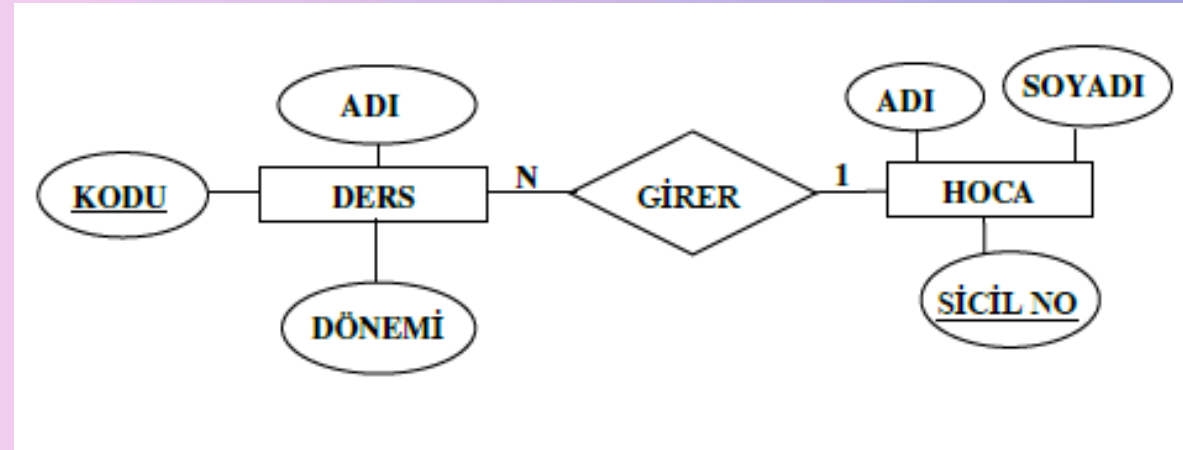
Proje			
Proje No	Proje Adı	Bütçe	Süresi
1	P1	200	10 yıl
2	P3	100	3 yıl
3	P16	100	4 yıl

Çalışan					
Sicil No	Adı	Soyadı	Giriş Tarihi	Doğum Tarihi	Proje No
1	Veli	Tane	06/03/2004	15/10/1970	1
2	Ayşe	Şahin	15/10/2003	01/08/1966	2
3	Murat	Al	29/05/2002	10/02/1971	3

Bire-Çok İlişkilerin Dönüştürülmesi

VTYS 2. Hafta

Öğr. Gör. Furkan DURMUŞ



DERS			
Kodu	Adı	Dönemi	Sicil No
D1	İşletme	1	
D2	Veritabanı	1	
D3	Programlama	2	
D4	Pazarlama	2	

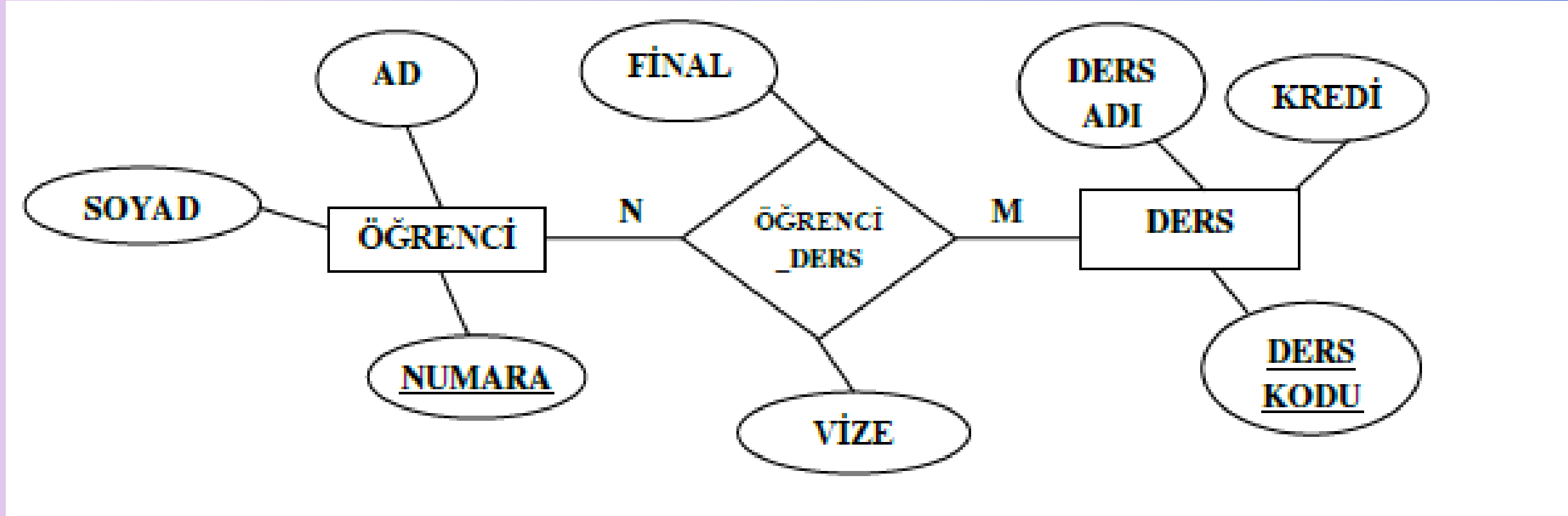
HOCA		
Sicil No	Adı	Soyadı
101	Ayşe	Kar
203	Ahmet	Can
607	Ali	Kaya
301	Veli	Dal

N

Çok-Çok İlişkilerin Dönüştürülmesi

VTYS 2. Hafta

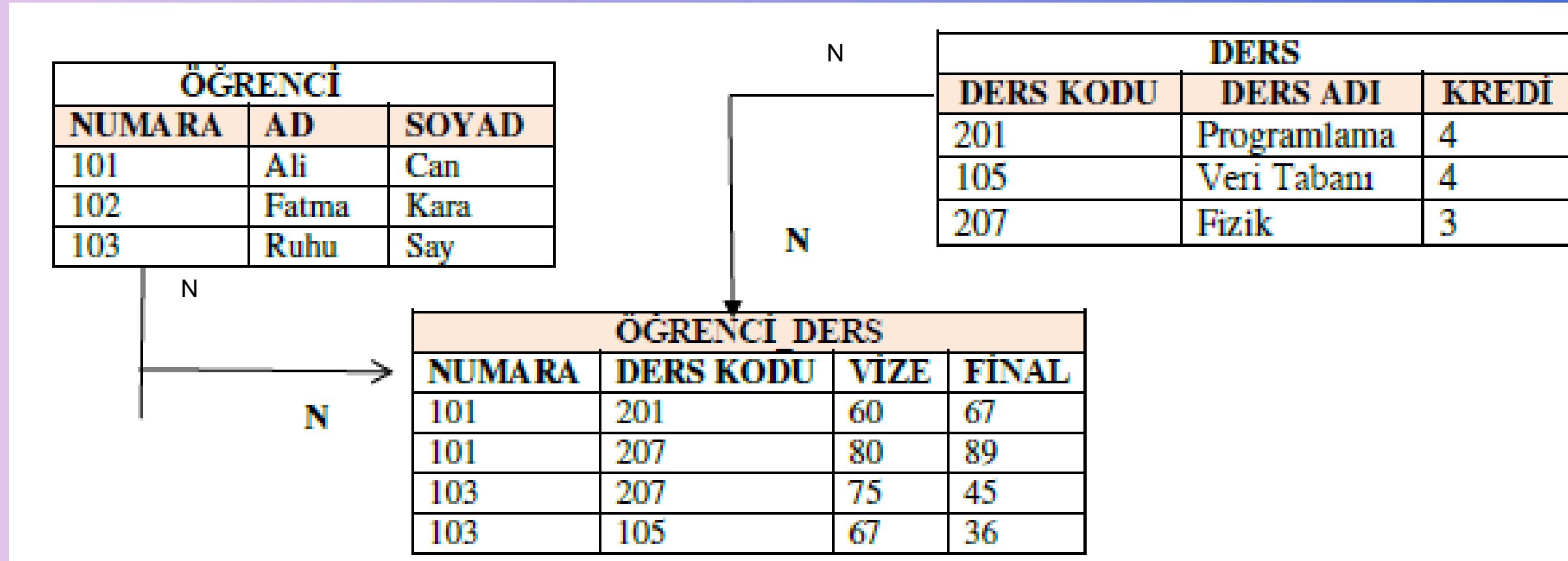
Öğr. Gör. Furkan DURMUŞ



Çok-Çok İlişkilerin Dönüştürülmesi

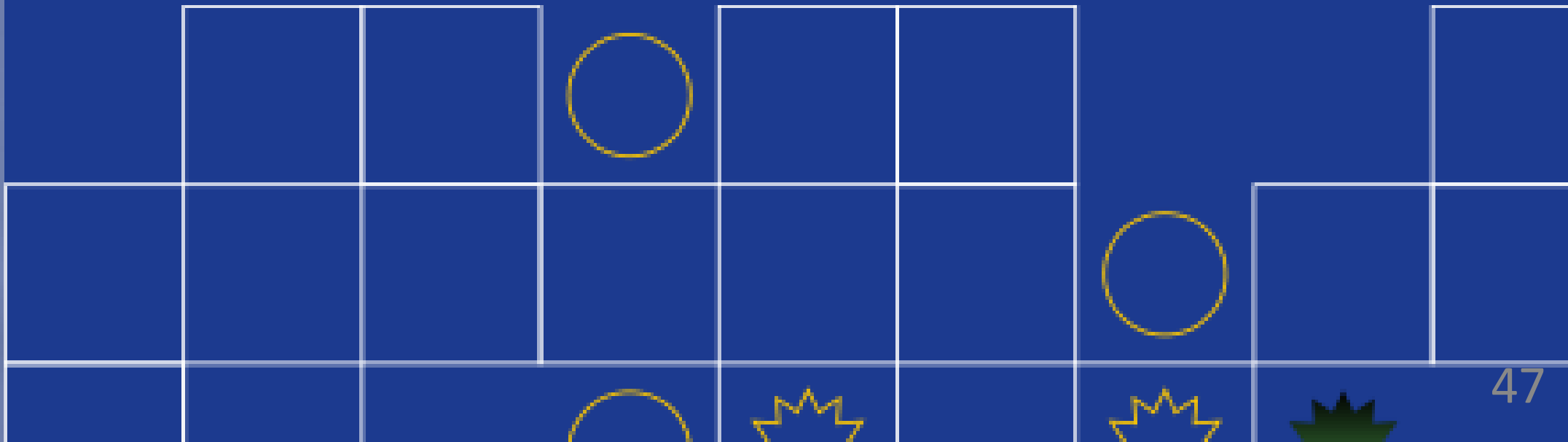
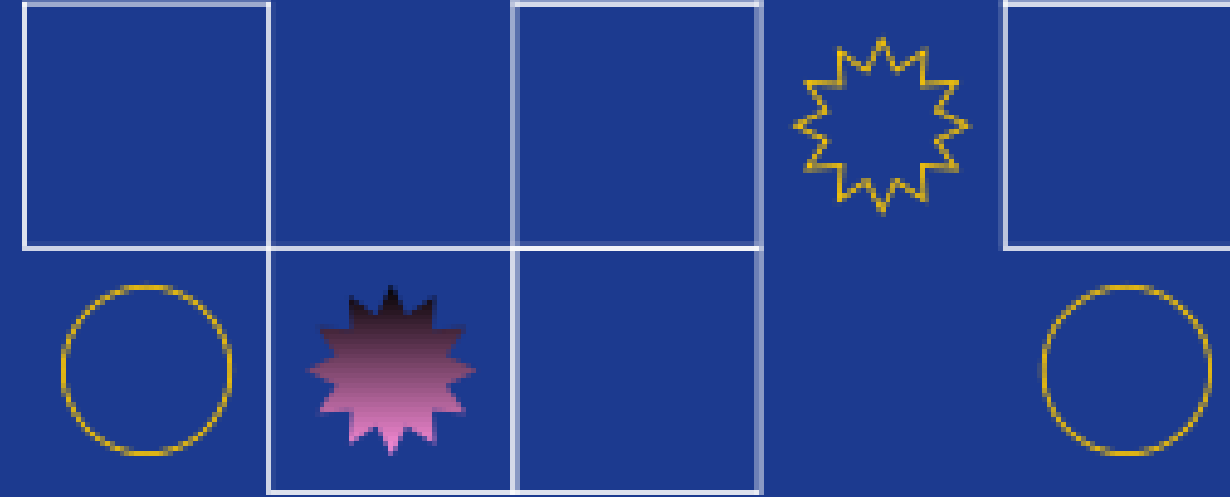
VTYS 2. Hafta

Öğr. Gör. Furkan DURMUŞ





Uygulama



Uygulama

Veritabanı Gereksinimleri (ŞİRKET)

Şirket, BÖLÜM'lerden oluşmaktadır. Her bölümün numarası ve adı vardır.

Her BÖLÜM bir veya daha fazla PROJE'yi kontrol eder, ancak her PROJE yalnızca bir BÖLÜM'e bağlıdır.

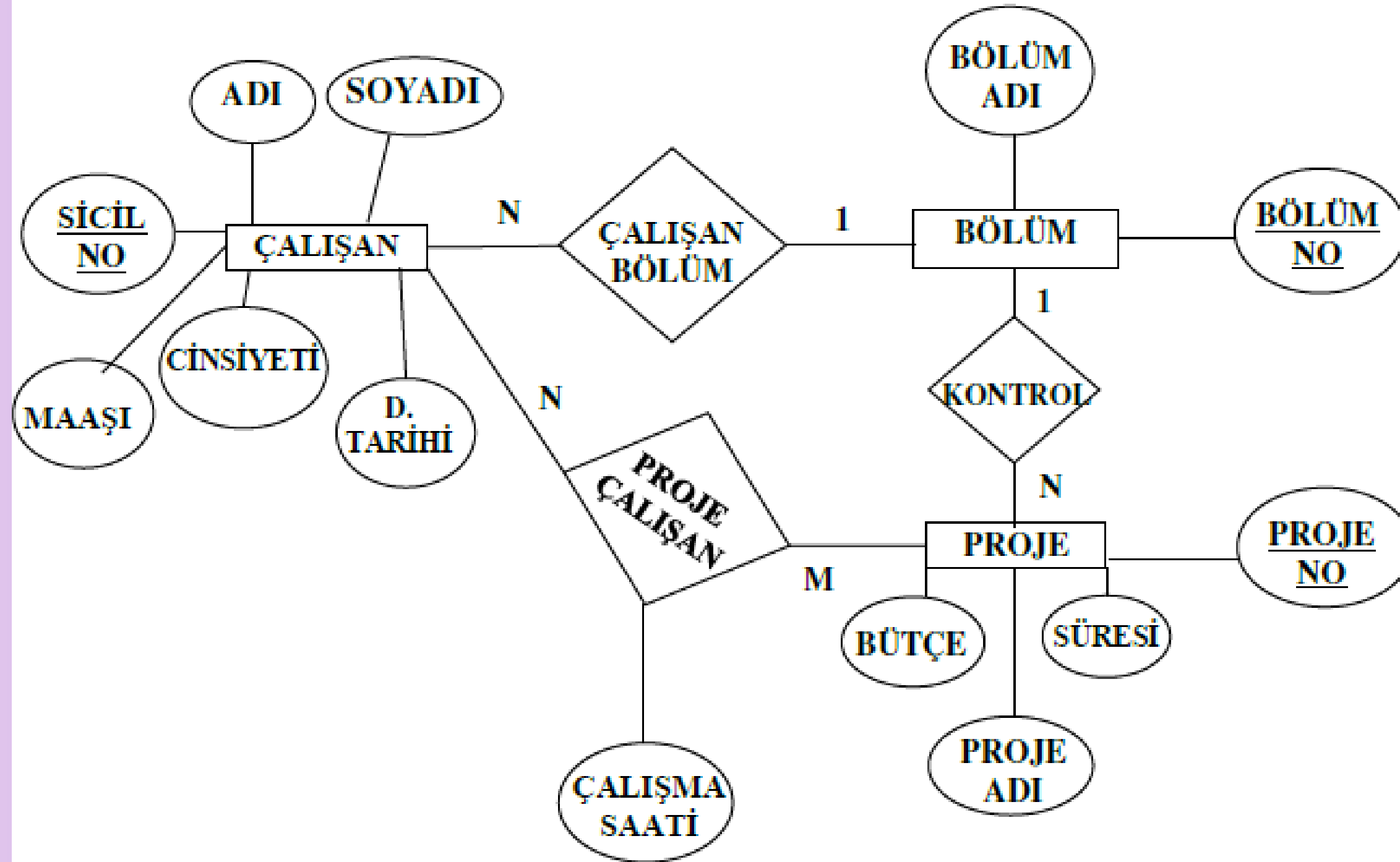
Her projenin kendine ait bir numarası, adı, bütçesi ve süresi bulunur.

Veritabanında şirket ÇALIŞAN'larının sicil numarası, adı, soyadı, maaşı, cinsiyeti ve doğum tarihi saklanacaktır.

Her çalışan yalnızca bir BÖLÜM'de yer almaktadır, fakat birden fazla PROJE'de çalışabilir. Yine bir projede birden çok çalışan vardır.

Her çalışanın, projelerde haftalık çalışma saatleri kaydedilecektir.

Varlık-İlişki Diyagramı

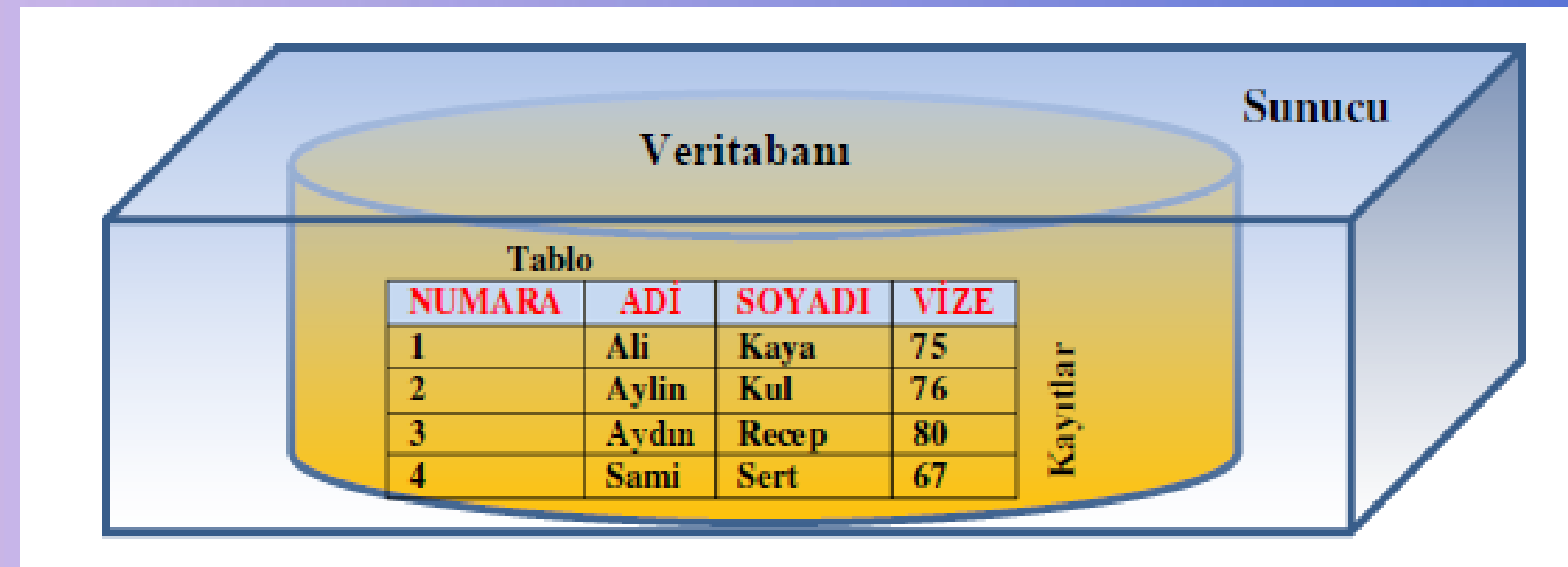


İlişkisel Veri Tabanının Genel Yapısı

VTYS 2. Hafta

Öğr. Gör. Furkan DURMUŞ

Tablolar veri tabanlarının temelidir, çünkü tablolar bilgileri organize olmuş bir şekilde tutar ve bizim bilgilere ulaşmamızı sağlar. Bir veri tabanı en az bir tablodan oluşur. Tablo, verileri düzgün kayıtlar şeklinde belirli bir düzene göre kalıcı ortamlarda bir veri tabanında bir isim altında saklayan, veriler üzerinde SQL yardımıyla ekleme, silme, güncelleme ve listeleme yapılabilen bir veri tabanı nesnesidir.

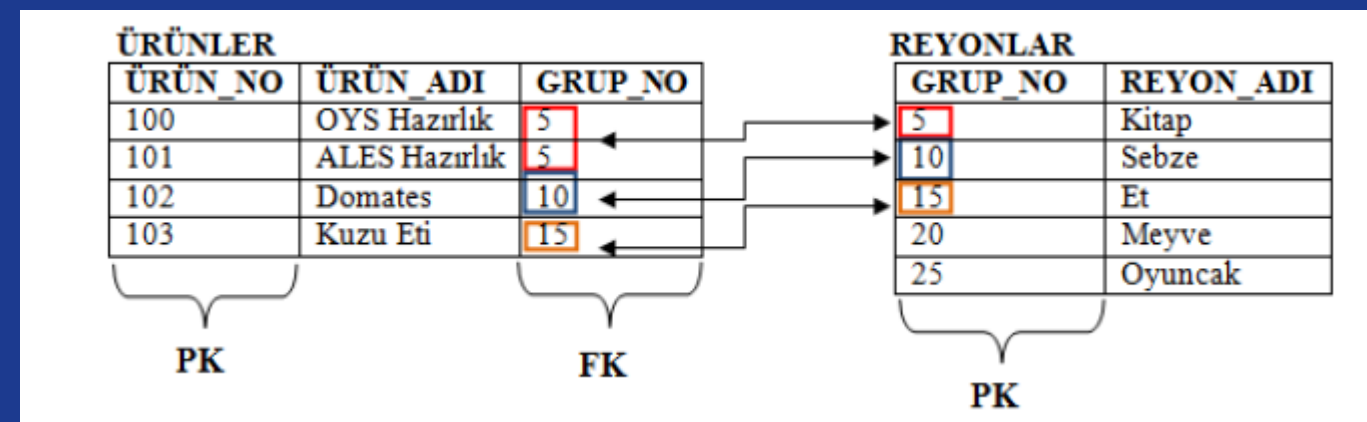


Birincil Anahtar (Primary Key)

Bir tablonun birincil anahtarı, tabloda depoladığınız her satırı benzersiz şekilde tanımlayan bir veya daha çok alandan oluşur. Genellikle, birincil anahtar olarak işlev gören bir TC kimlik numarası gibi benzersiz bir tanımlayıcı vardır. Günlük hayatımızda da, TC Kimlik Numaramız, illerin plaka ve posta kod numaraları bağlı buldukları ülkelerde benzersiz numaralardır birincil anahtarları başlarındaki sayısal ifadelerdir. Birincil anahtarlar hiçbir zaman NULL(boş) veya birbiri ile aynı olan değerleri içeremez.

Yabancı Anahtar (Foreign Key)

Tablo içerisindeki verilerin birbirleri ile iletişim kurabilmeleri amacıyla kullanılan anahtarlardır. Birincil anahtarlar hiçbir zaman NULL(boş) veya birbiri ile aynı olan değerleri içeremezken, yabancı anahtarlar birbirleri ile aynı olan değerler içerebilirler. Bir tabloda birden fazla yabancı anahtar kullanılabilir. Kısacası yabancı anahtar, bir tabloya girilebilecek verileri başka bir tablonun herhangi bir alanında yer alabilecek veriler ile sınırlandırmak ve ilişkilendirmek için kullanılır. Yabancı anahtara, başka bir tablonun birincil anahtarı da denilebilir.



Unique Key (Tekil Anahtar)

Unique Key olarak tanımlanan alan için bir değer sadece bir kere girilebilir. Bir başka satıra daha aynı verinin girilmesine izin verilmez. Primary Key 'den farklı olarak Unique Key, NULL (boşluk) değerini alabilir. Örneğin programda her personele ait bir sicil numarası olacağı için bu alan Unique key olarak tanımlanabilir. Ama isim alanı birden fazla aynı isme sahip personel olabileceği için bir Unique key olarak tanımlanamaz. Ali isimli birden fazla personel olabileceği gibi.

Veri Kısıtlamaları

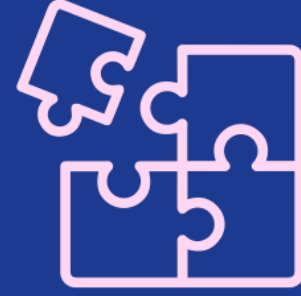
Tablo tasarlarken kullanılan verilerin tutarlılığını sağlamak ve ne tür değerlere sahip olabileceğini belirlemek için de kısıtlamalar getirilebilir. Veri kısıtlamaları sırasında kullanılan bazı kısıtlamalar şunlardır:



01

Not Null Kısıtlaması

Veri girişi yapılacak bir tablodaki sütunun değer alıp (NULL) almaması (NOT NULL) gerektiğini belirlemek için kullanılan kısıtlamadır.



02

Default Kısıtlaması

Veri girişi sırasında bir alanın alabileceği varsayılan bir değer atamak için kullanılır.



03

Unique Kısıtlaması

Tablodaki bir alana girilen verinin tekrarsız olmasını sağlamak için kullanılır.



04

Check Kısıtlaması

Kontrol kısıtlayıcı olarak da adlandırılır. Veri girişlerinin belirtilen kriterlere göre yapılmasını sağlar. Örneğin kişinin T.C. Kimlik numarası girilirken 11 haneden fazla değer girilmesi engellenebilir.

