

Veritabanı Yönetim Sistemleri (335)

Dr. Öğr. Üyesi Ahmet Arif AYDIN

L9-

İlişkisel Veri Modeli
(Relational Data Model)

GÜZ -2022

Sorular

- ER Model
 - Ternary relationship, aggregation
 - composition, has-a
 - ER model neden önemlidir?
- UML(Unified Modeling Language) nedir ?
 - UML'in kullanım alanları nelerdir?
 - UML diagram çeşitleri nelerdir ? (Yapısal, Davranışsal)
 - UML Diagramlar: Activity, Class, Sequence, ..,

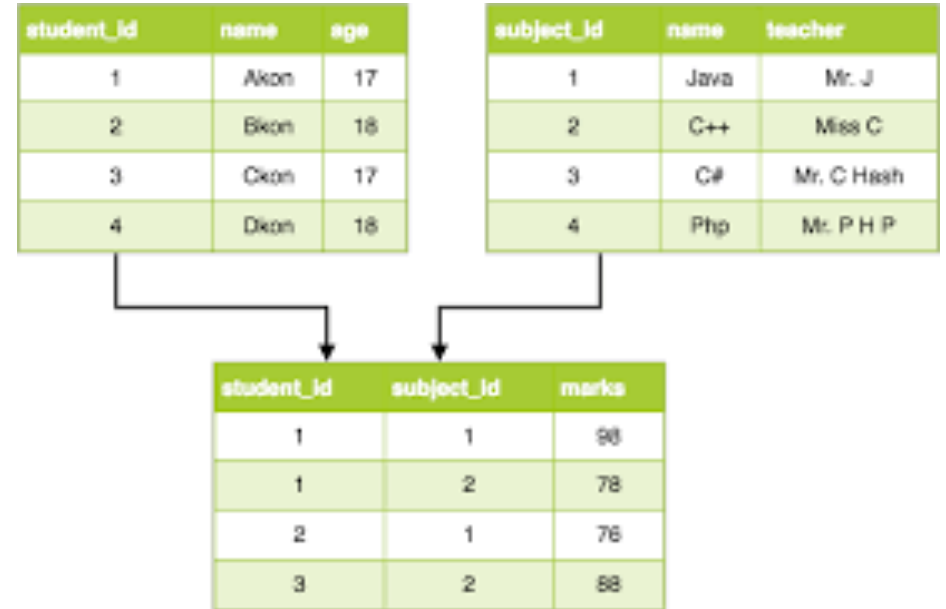
Sorular-2

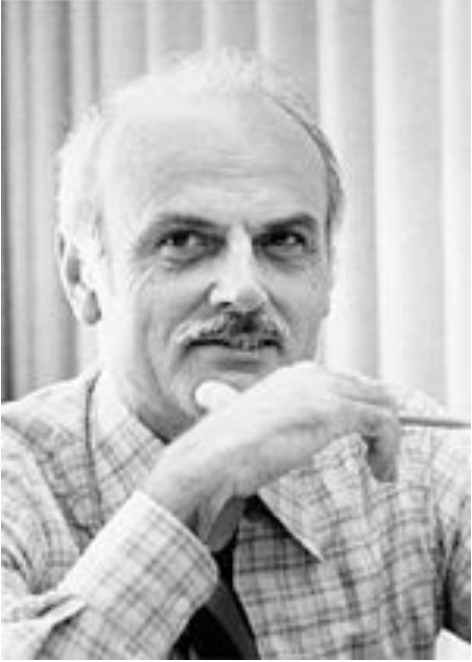
- Nesne Tabanlı Programlama Kavramları
 - Abstraction (soyutlama)
 - Encapsulation (kapsülleme)
 - Information hiding (bilgi gizleme)
 - Inheritance (kalıtım)
 - Polymorphism (çok biçimlilik)

İlişkisel Veri Modeli

1970'de Edgar Codd

- IBM's San Jose Research Lab
- İlişkisel veri modeli (relational data model) kavramını ortaya çıkarmıştır
- 1981 tarihinde ACM's Turing ödülünü almıştır.





A Relational Model of Data for Large Shared Data Banks

E. F. CODD

IBM Research Laboratory, San Jose, California

Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). A prompting service which supplies such information is not a satisfactory solution. Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information.

Existing noninferential, formatted data systems provide users with tree-structured files or slightly more general network models of the data. In Section 1, inadequacies of these models are discussed. A model based on n -ary relations, a normal form for data base relations, and the concept of a universal data sublanguage are introduced. In Section 2, certain operations on relations (other than logical inference) are discussed and applied to the problems of redundancy and consistency in the user's model.

KEY WORDS AND PHRASES: data bank, data base, data structure, data organization, hierarchies of data, networks of data, relations, derivability, redundancy, consistency, composition, join, retrieval language, predicate calculus, security, data integrity

The relational view (or model) of data described in Section 1 appears to be superior in several respects to the graph or network model [3, 4] presently in vogue for non-inferential systems. It provides a means of describing data with its natural structure only—that is, without superimposing any additional structure for machine representation purposes. Accordingly, it provides a basis for a high level data language which will yield maximal independence between programs on the one hand and machine representation and organization of data on the other.

A further advantage of the relational view is that it forms a sound basis for treating derivability, redundancy, and consistency of relations—these are discussed in Section 2. The network model, on the other hand, has spawned a number of confusions, not the least of which is mistaking the derivation of connections for the derivation of relations (see remarks in Section 2 on the “connection trap”).

Finally, the relational view permits a clearer evaluation of the scope and logical limitations of present formatted data systems, and also the relative merits (from a logical standpoint) of competing representations of data within a single system. Examples of this clearer perspective are cited in various parts of this paper. Implementations of systems to support the relational model are not discussed.

1.2. DATA DEPENDENCIES IN PRESENT SYSTEMS

The provision of data description tables in recently developed information systems represents a major advance toward the goal of data independence [5, 6, 7]. Such tables facilitate changing certain characteristics of the data representation stored in a data bank. However, the variety of data representation characteristics which can be changed

İlişkisel Veri Modeli (Relational Data Model)

Günümüzde bir çok veritabanı yönetim sistemi ilişkisel veri modelini kullanmaktadır.

☐ include secondary database models

143 systems in ranking, November 2020

Rank	Rank			DBMS	Database Model	Score		
	Nov 2020	Oct 2020	Nov 2019			Nov 2020	Oct 2020	Nov 2019
1.	1.	1.	1.	Oracle +	Relational, Multi-model i	1345.00	-23.77	+8.93
2.	2.	2.	2.	MySQL +	Relational, Multi-model i	1241.64	-14.74	-24.64
3.	3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	1037.64	-5.48	-44.27
4.	4.	4.	4.	PostgreSQL +	Relational, Multi-model i	555.06	+12.66	+63.99
5.	5.	5.	5.	IBM Db2 +	Relational, Multi-model i	161.62	-0.28	-10.98

☐ include secondary database models

152 systems in ranking, October 2021

Rank	Rank			DBMS	Database Model	Score		
	Oct 2021	Sep 2021	Oct 2020			Oct 2021	Sep 2021	Oct 2020
1.	1.	1.	1.	Oracle +	Relational, Multi-model i	1270.35	-1.19	-98.42
2.	2.	2.	2.	MySQL +	Relational, Multi-model i	1219.77	+7.24	-36.61
3.	3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	970.61	-0.24	-72.51
4.	4.	4.	4.	PostgreSQL +	Relational, Multi-model i	586.07	+6.47	+44.57
5.	5.	5.	5.	IBM Db2	Relational, Multi-model i	149.66	-1.73	-16.30
6.	6.	6.	6.	SQLite +	Relational	138.17	-1.87	+21.79
7.	7.	7.	7.	Microsoft Access	Relational	137.80	-1.02	+8.43
8.	8.	8.	8.	MariaDB +	Relational, Multi-model i	109.31	-0.85	+6.71
9.	9.	9.	10.	Hive +	Relational	106.72	+3.22	+48.46
10.	10.	10.	11.	Microsoft Azure SQL Database	Relational, Multi-model i	84.96	+0.54	+5.24
11.	11.	11.	9.	Teradata +	Relational	80.60	+2.17	-4.14
12.	12.	14.	37.	Snowflake +	Relational	66.07	-0.51	-3.76
13.	13.	12.	13.	SAP HANA +	Relational, Multi-model i	1236.37	-1.88	-33.98
14.	14.	13.	14.	FileMaker	Relational, Multi-model i	1205.38	-7.09	-14.39
15.	15.	15.	12.	SAP Adaptive	Relational, Multi-model i	924.68	-1.62	-45.93
16.	16.	16.	15.	Google BigQuery	Relational, Multi-model i	622.72	+2.26	+35.75
17.	17.	17.	19.	Firebird	Relational	149.66	-1.73	-16.30
18.	18.	18.	16.	Amazon Redshift	Relational, Multi-model i	138.17	-1.87	+21.79
19.	19.	19.	17.	Informix	Relational	137.80	-1.02	+8.43
20.	20.	20.	20.	Spark SQL +	Relational, Multi-model i	109.31	-0.85	+6.71

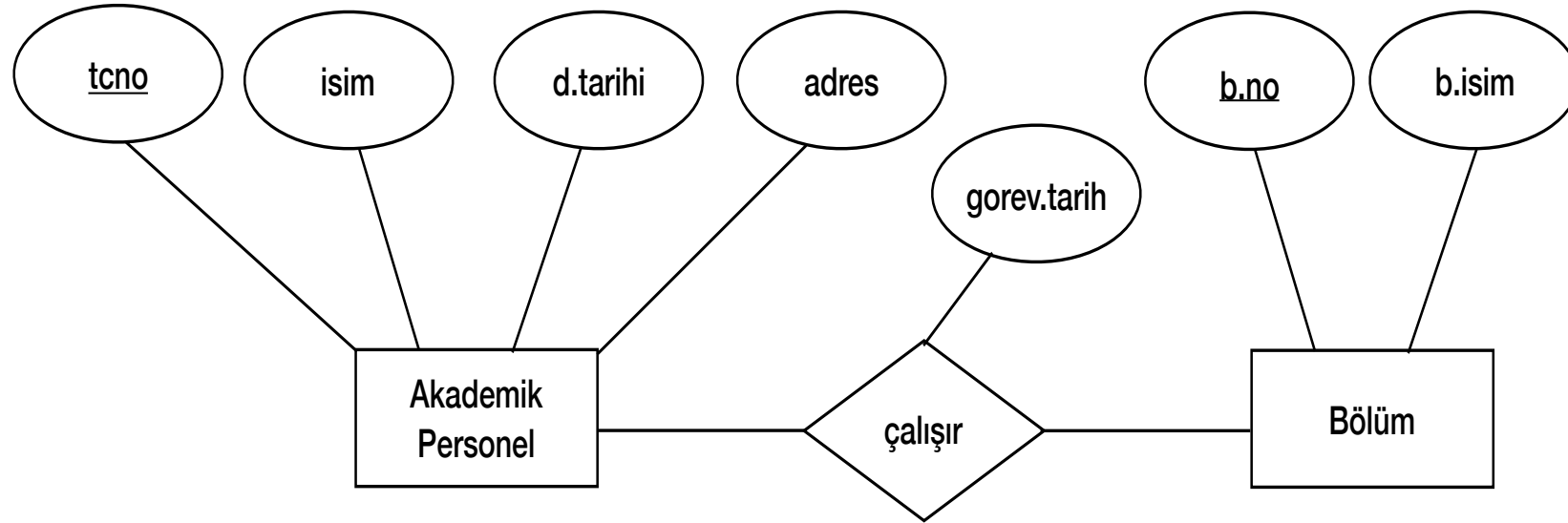
☐ include secondary database models

162 systems in ranking, October 2022

Rank	Rank			DBMS	Database Model	Score		
	Oct 2022	Sep 2022	Oct 2021			Oct 2022	Sep 2022	Oct 2021
1.	1.	1.	1.	Oracle +	Relational, Multi-model i	1236.37	-1.88	-33.98
2.	2.	2.	2.	MySQL +	Relational, Multi-model i	1205.38	-7.09	-14.39
3.	3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	924.68	-1.62	-45.93
4.	4.	4.	4.	PostgreSQL +	Relational, Multi-model i	622.72	+2.26	+35.75
5.	5.	5.	5.	IBM Db2	Relational, Multi-model i	149.66	-1.73	-16.30
6.	6.	6.	7.	Microsoft Access	Relational	138.17	-1.87	+21.79
7.	7.	7.	6.	SQLite +	Relational	137.80	-1.02	+8.43
8.	8.	8.	8.	MariaDB +	Relational, Multi-model i	109.31	-0.85	+6.71
9.	9.	9.	12.	Snowflake +	Relational	106.72	+3.22	+48.46
10.	10.	10.	10.	Microsoft Azure SQL Database	Relational, Multi-model i	84.96	+0.54	+5.24
11.	11.	11.	9.	Hive	Relational	80.60	+2.17	-4.14
12.	12.	12.	11.	Teradata	Relational, Multi-model i	66.07	-0.51	-3.76

<https://db-engines.com/en/ranking/relational+dbms>

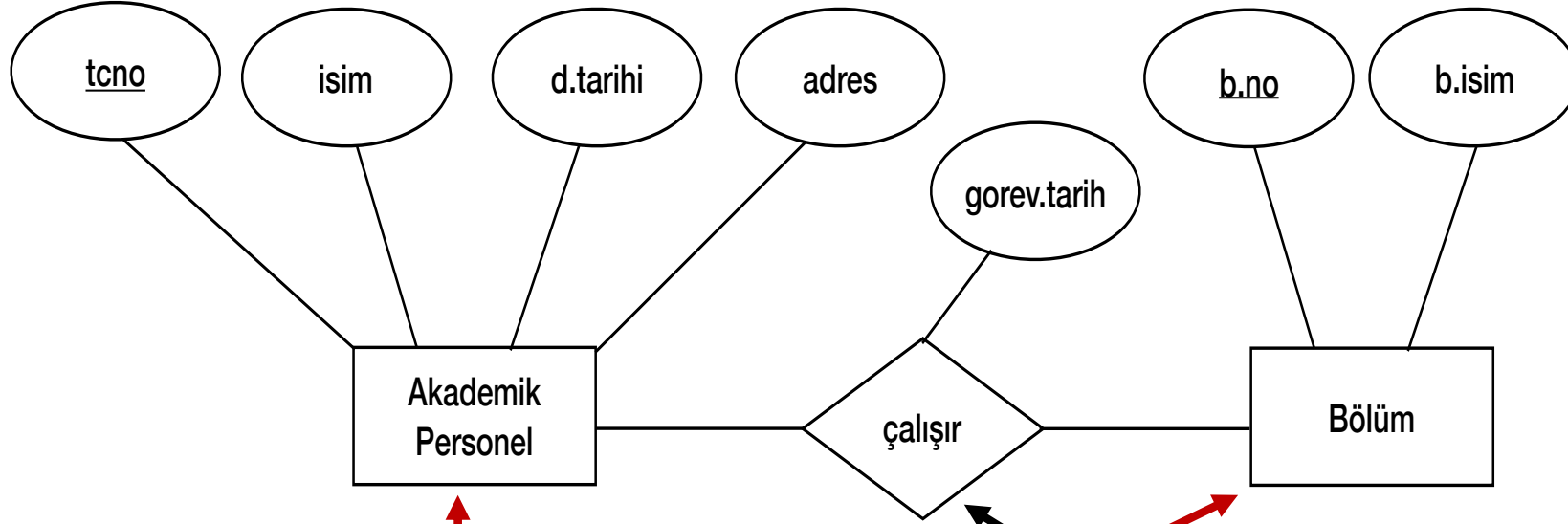
İlişkisel Veri Modeli (Relational Data Model)



ER modelde tanımlanan varlık setleri ilişkisel modelde birer tablo olarak modellenir.

ER diagramında bulunan ilişki setlerinden **sadece many-to-many** olanlar ilişkisel modelde tabloya dönüştürülür

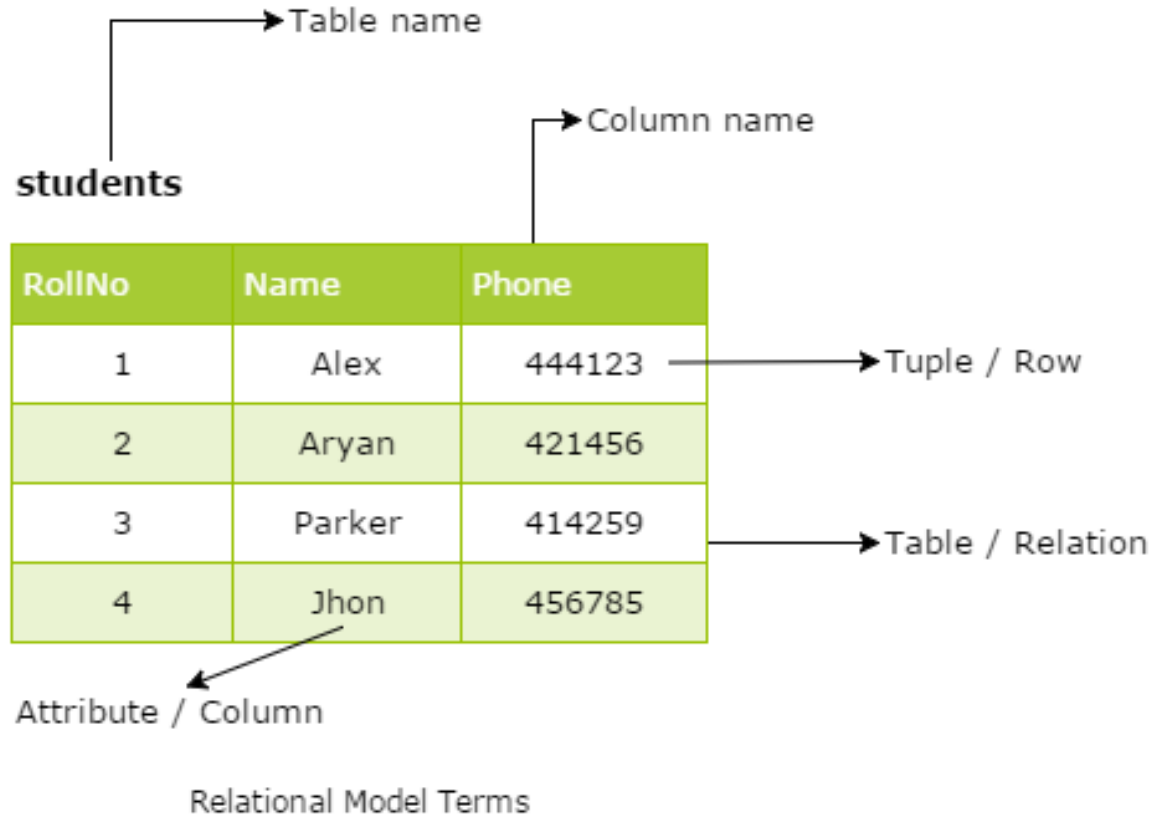
İlişkisel Veri Modeli (Relational Data Model)



ER modelde tanımlanan varlık setleri ilişkisel modelde birer tablo olarak modellenir.

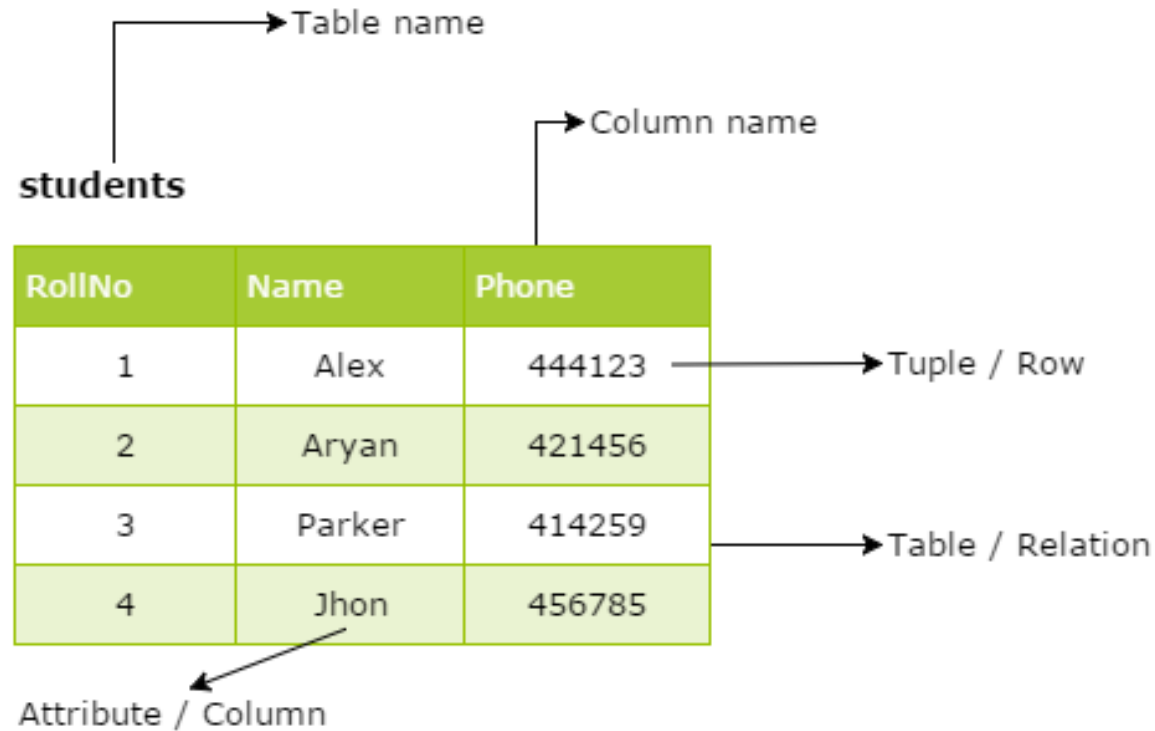
ER diagramında bulunan ilişki setlerinden sadece **many-to-many** olanlar ilişkisel modelde tabloya dönüştürülür

İlişkisel Veri Modeli (Relational Data Model)



- Tablo **relation** olarak adlandırılır.
- Tablola **satırlar ve sütunlar** dan oluşan iki boyutlu bir yapıdır.
- Her **bir satır** ER varlık setinde bulunan **bir varlığı** temsil eder.
- Her bir satır benzersiz olarak tanımlanamalıdır.
- Tablo yapısı kavramsal (logical) bir gösterimdir.
- Network ve hiyerarşik modellere göre kullanımı ve yönetimi daha kolaydır.

İlişkisel Veri Modeli (Relational Data Model)



Relational Model Terms

Avantajlar

- Tablolar verinin yapısının daha kolay bir biçimde anlaşılmasını sağlar.
- Tablo yapısı veri üzerinde gerçekleştirilecek sorgulamaları kolaylaştırır.
- Tablolar arasında ilişkiler oluşturma imkanı sağlar.
- Tablolar arasındaki ilişkiler ve kolon tipleri ve kısıtlamalar ile verinin tutarlılığı korunur.

İlişkisel Veri Modeli (Relational Data Model): Şema

İlişkisel tabloların yapısı hakkında bilgi veren gösterim biçimi **şema** olarak adlandırılır.

```
öğrenci (öğrencino: integer,  
        isim: varchar,  
        kullanıcıadı: varchar,  
        yaş: integer,  
        ortalama: float)
```

öğrenci
varlık setinin (tablo)
şeması

İlişkisel Veri Modeli (Relational Data Model): Şema

İlişkisel tabloların yapısı hakkında bilgi veren gösterim biçimi **şema** olarak adlandırılır.

```
öğrenci (öğrencino: integer,  
        isim: varchar,  
        kullanıcıadı: varchar,  
        yaş: integer,  
        ortalama: float)
```

öğrenci
varlık setinin (tablo)
şeması

Şema

- tablo adı (öğrenci)
- her bir satırdaki **sütun sayısı** (5)
- her bir sütuna kaydedilecek verinin tipi
- sütun sırası !!!

bilgilerini sağlamaktadır.

İlişkisel Veri Modeli (Relational Data Model): Tablo

kolonlar (nitelikler, alan)

öğrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

satır (kayıt, tuple)

İlişkisel veri modelinde bütün **satırlar (row, tuple)**

- aynı sırada olan alanları bulundurur
- bir öğrencinin bilgilerini içermelidir
- unique olmak zorundadır
- aynı şemayı içermek zorundadır.
- öğrenci tablosunda 5 satır bulunmakta ve her bir satır da 5 sütundan oluşmaktadır.

İlişkisel Veri Modeli (Relational Data Model): Tablo

kolonlar (nitelikler, alan)

öğrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

satır (kayıt, tuple)

- Öğrenci tablosunun
 - **derecesi** (degree of relation) kolon sayıdır 5
 - satır sayısı (**cardinality**) 5 dir
 - Kaydedilen verinin tipi sadece bir alan için değiştirilemez. (alter komutu ile hepsi değiştirilebilir)
- Bir veritabanında bulunan tablo isimleri unique olmalıdır
- Alter komutu ile yeni kolon eklenebilir

- Bütünlük kısıtlamaları bir veritabanında depolanan verilerin tutarlılığını (consistency) ve doğruluğunu (veracity) sağlar
- Tabloların tasarımı yapılırken ve oluşturulurken tanımlanan özellikler tablonun bütünlük kısıtlamalarını belirler.
- Veritabanı oluşturulan tabloların kısıtlama ve şartlarına uygun olan verileri tablolarda depolar.

İlişkisel Veri Modeli (Relational Data Model): Tabloların bütünlük kısıtlamaları

Öğrenci tablosunun kısıtlamaları :

- *isim* alanı sadece rakamlardan oluşamaz
- *yaş* sütununda integer tipinde sayı bulunması gerekmektedir (karakter bulunamaz)
- bilgileri depolanan her bir öğrencinin *öğrencino* alanı unique olmak zorundadır.

satır (kayıt, tuple)

kolonlar (nitelikler, alan)

öğrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

öğrenci (**öğrencino**:integer, **isim**:varchar(20), **kullanıcıadı**:
varchar(20), **yaş**:integer, **ortalama**:float)

İlişkisel Veri Modeli (Relational Data Model): Tabloların bütünlük kısıtlamaları

Bir varlık seti içerisinde (bir tabloda) bulunan nesneleri (satır) ve özelliklerini (kolonlar) diğer varlıklardan ayırt etmemizi sağlayan nitelik **anahtar (key)** olarak tanımlanır.

Anahtarlar tabloların ilişkilendirilmesini sağlar

satır (kayıt, tuple)

kolonlar (nitelikler, alan)				
öğrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

İlişkisel Veri Modeli (Relational Data Model): Tabloların bütünlük kısıtlamaları

Anahtar seçimi

- verinin tutarlılığı,
- veriye etkili bir biçimde erişim
- verinin sistematik bir biçimde depolanması

açısından çok önemlidir.

satır (kayıt, tuple)

kolonlar (nitelikler, alan)

öğrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

İlişkisel Veri Modeli: Birincil Anahtar (Primary Key)

primary key

Öğrenci Tablosu

öğrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

- Tablo tasarımında **birincil anahtar (primary key)** seçilmesi çok önemlidir.
- Öğrenci tablosunun öğrencino alanı primary key olarak tanımlanabilir
- Bu seçim ile birlikte **tabloya kaydedilecek her bir satırın benzersiz (unique) olması** sağlanır.

İlişkisel Veri Modeli: Birincil Anahtar (Primary Key)

primary key

Öğrenci Tablosu

öğrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

Primary key de **olması gereken** özellikler:

- NULL değer içermemelidir
- Kayıt işleminden sonra değişmemelidir
- Boyutu mümkün olduğu kadar az olmalıdır
- Şifrelenmiş değer içermemelidir.

İlişkisel Veri Modeli: İkincil Anahtar (Foreign Key)

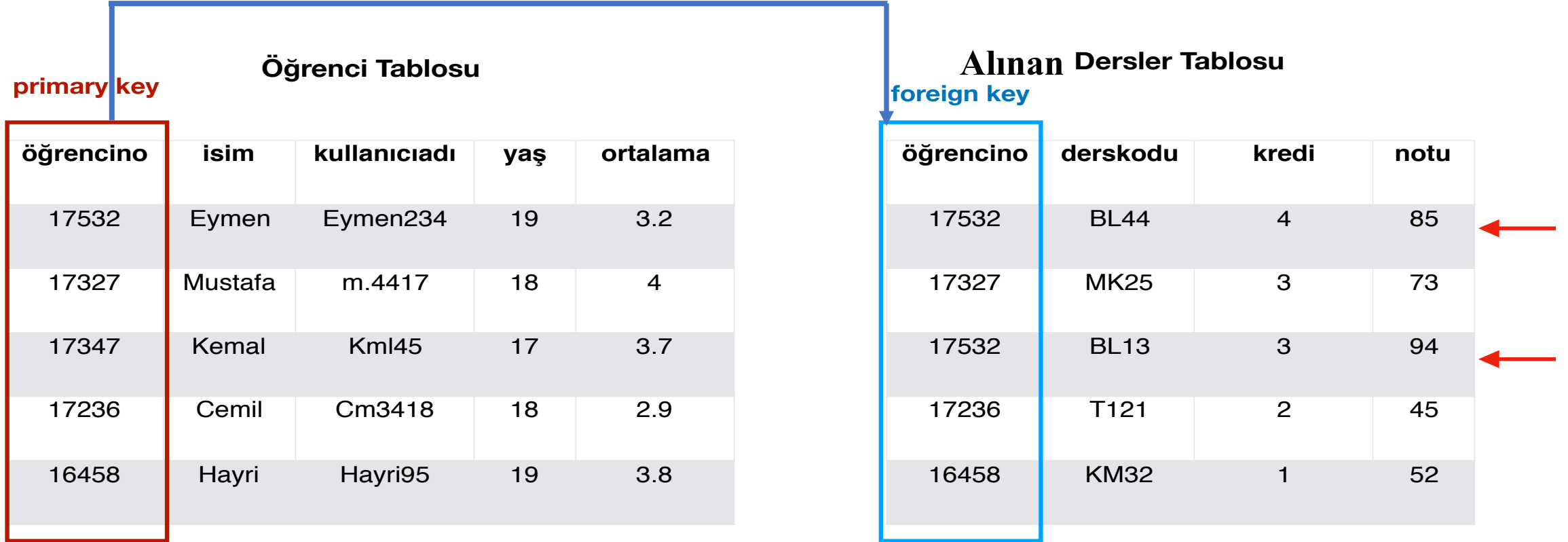
Bir tablodaki alanın (primary key)
başka bir tablodaki bir alanla ilişkilendirilmesiyle
yabancı anahtar (foreign key) oluşturulur.

Alınan Dersler Tablosu

foreign key

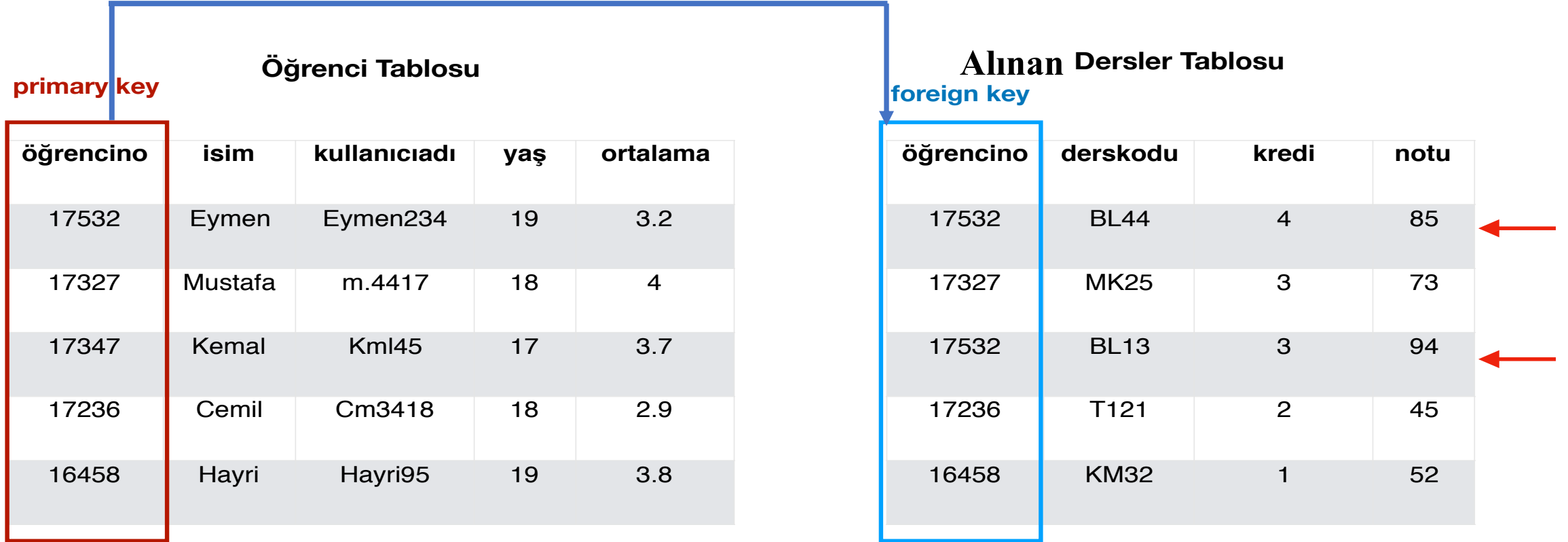
öğrencino	derskodu	kredi	notu
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52

İlişkisel Veri Modeli: İkincil Anahtar (Foreign Key)



- AlınanDersler tablosunun öğrencino alanı Öğrenci tablosunun öğrencino alanı ile ilişkilendirilip foreign key olarak tanımlanmıştır
- 17532 nolu öğrencinin farklı derslere ait iki adet kaydı bulunmaktadır


İlişkisel Veri Modeli: İkincil Anahtar (Foreign Key)



- AlınanDersler tablosuna öğrenci tablosundan bulunmayan bir kaydı ekleyemezsiniz
- AlınanDersler tablosunda kaydı bulunan bir öğrenciyi Öğrenci tablosundan silemezsiniz
- Bu kısıtlamalara ek olarak SQL dilini kullanarak tabloları oluştururken ek kısıtlamalar eklenebilir (yaş > 17)

İlişkisel Veri Modeli: Birleşik Anahtar (Composit Key)

Primary Key

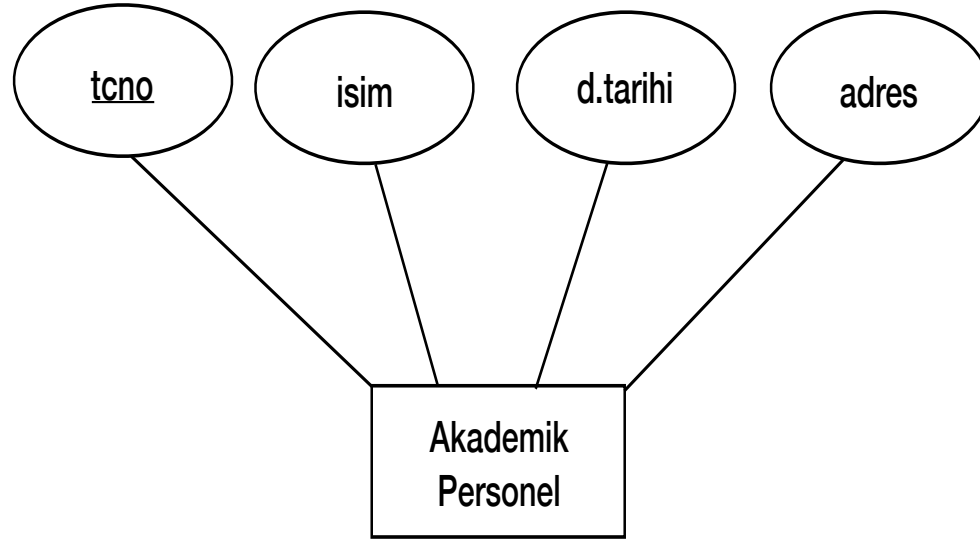


ProductID	VendorID	AverageLeadTime	StandardPrice	LastReceiptCost
1	1	17	47.8700	50.2635
2	104	19	39.9200	41.9160
7	4	17	54.3100	57.0255
609	7	17	25.7700	27.0585
609	100	19	28.1700	29.5785

ProductVendor table

Birden fazla alanın bir araya getirilerek oluşturulan birincil anahtara **composit key** denir.

ER Modelden İlişkisel Veri Modeline Geçiş

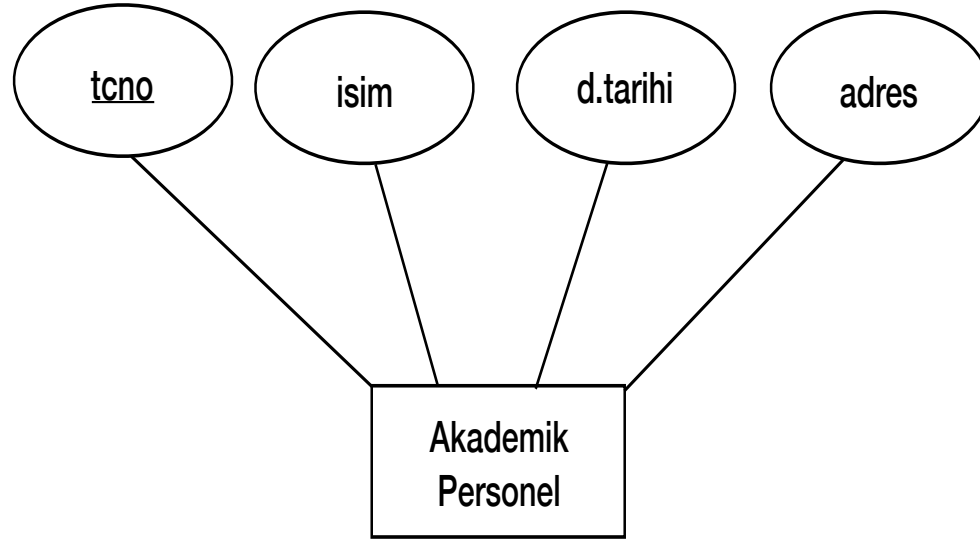


Akademik Personel
varlık setinin
ER diagram gösterimi



Akademik Personel
tablosunu oluşturalım

ER Modelden İlişkisel Veri Modeline Geçiş

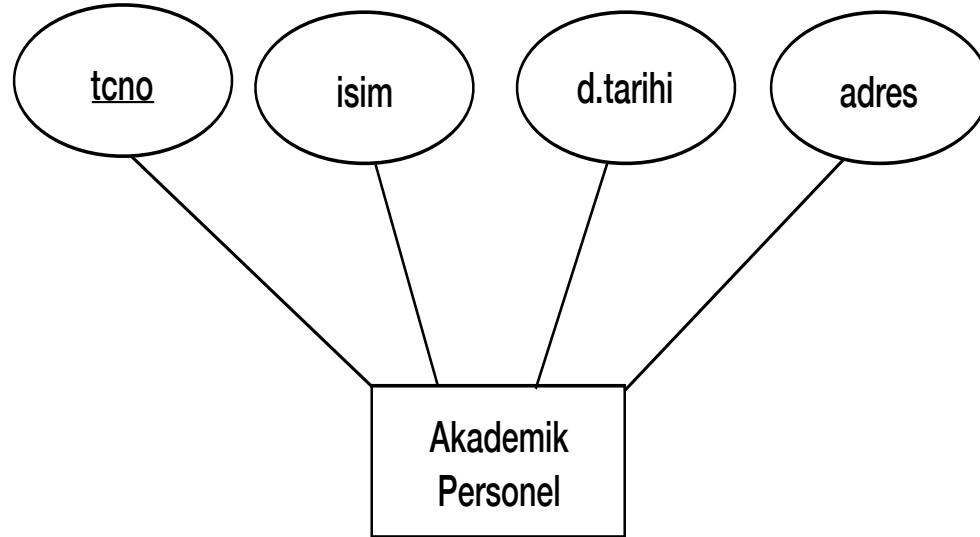


Akademik Personel
varlık setinin
ER diagram gösterimi

Akademik personel
(**tcno**:integer,
isim:varchar(20),
d.tarihi:varchar(20)
, **adres**: varchar)

şema

ER Modelden İlişkisel Veri Modeline Geçiş



Akademik Personel
varlık setinin
ER diagram gösterimi

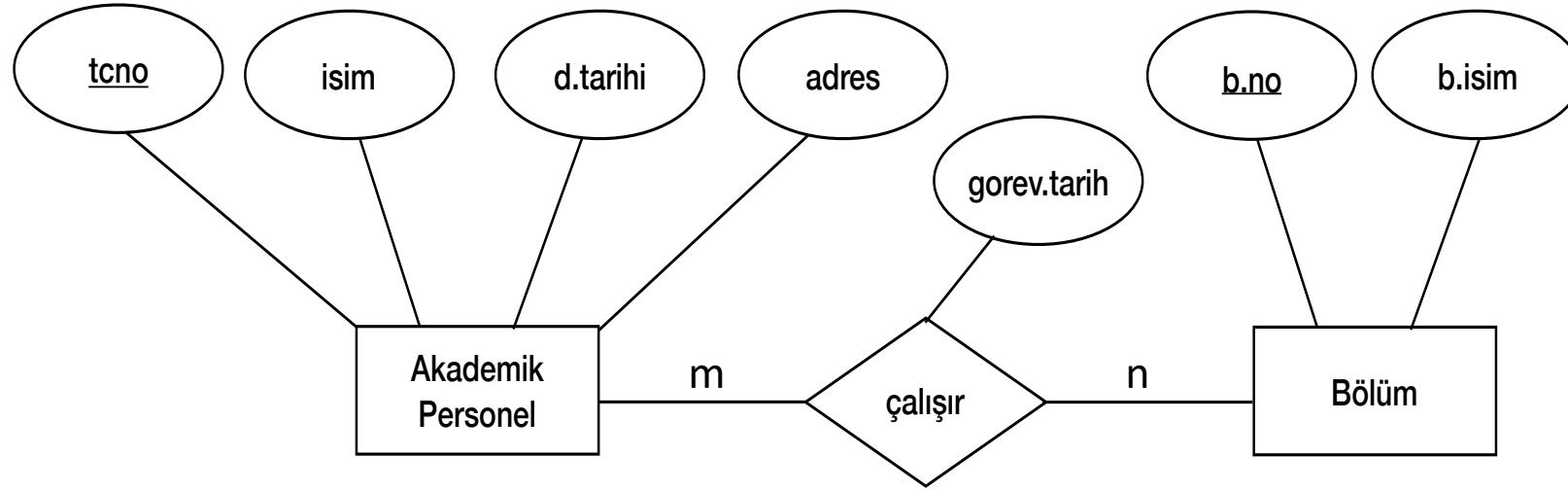
Akademik personel
(**tcno**:integer,
isim:varchar(20),
d.tarihi:varchar(20)
,adres: varchar)

şema

<u>tcno</u>	isim	d.tarihi	adres
335427	Mustafa	03.03.1993	Malatya
.	.	.	.

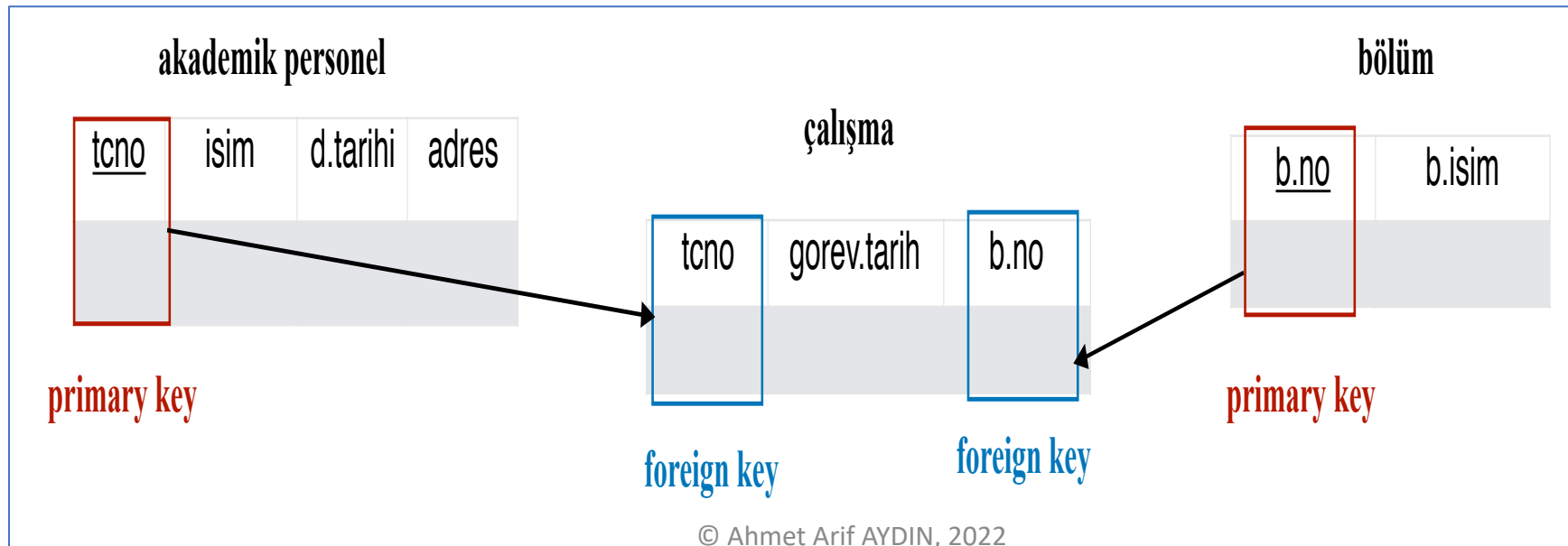
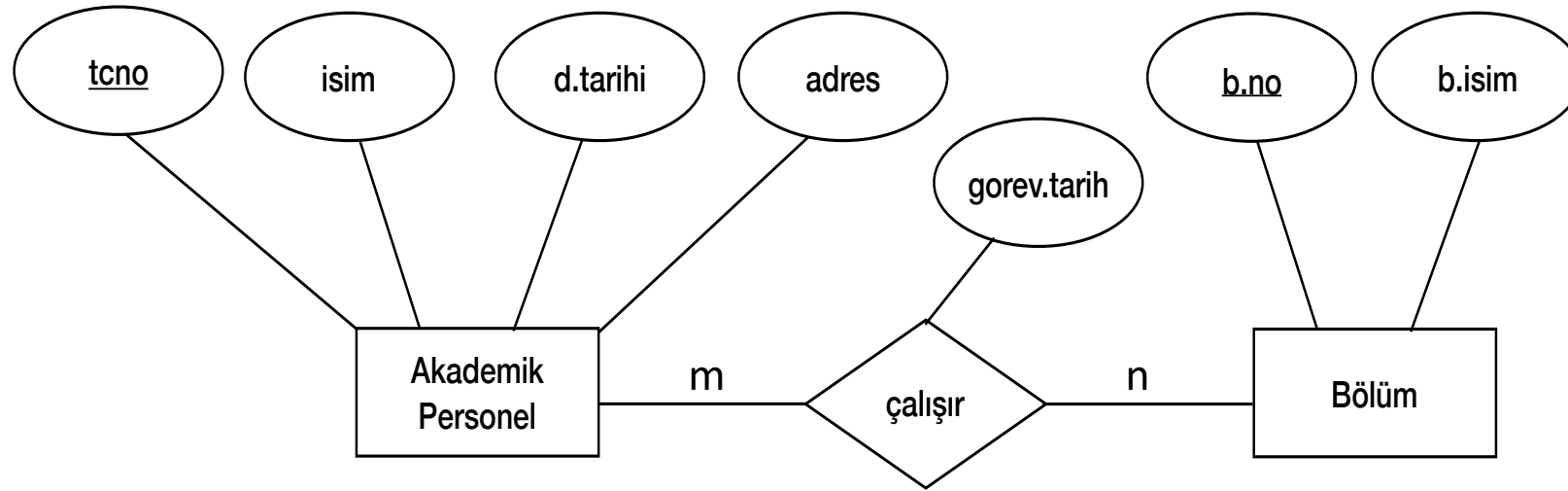
Akademik
Personel
tablosu

ER Modelden - İlişkisel Veri Modeline Geçiş

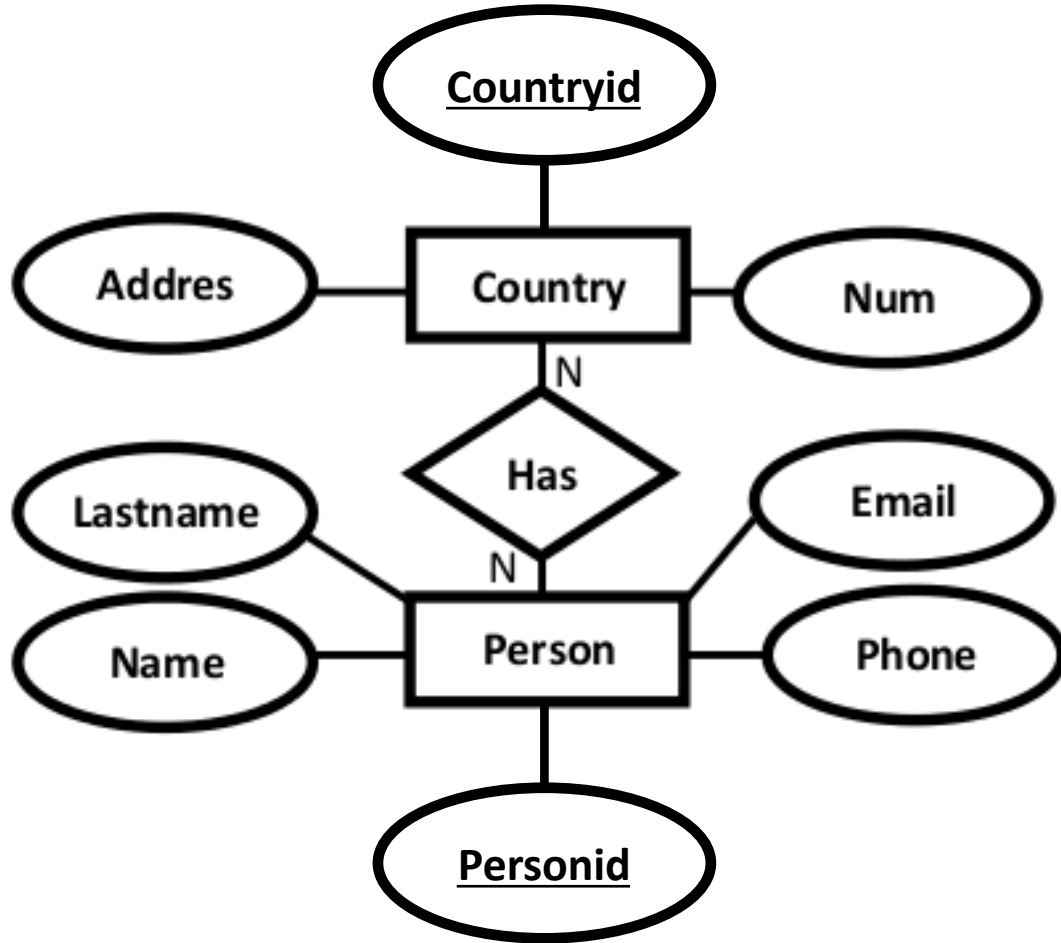


?

ER Modelden - İlişkisel Veri Modeline Geçiş



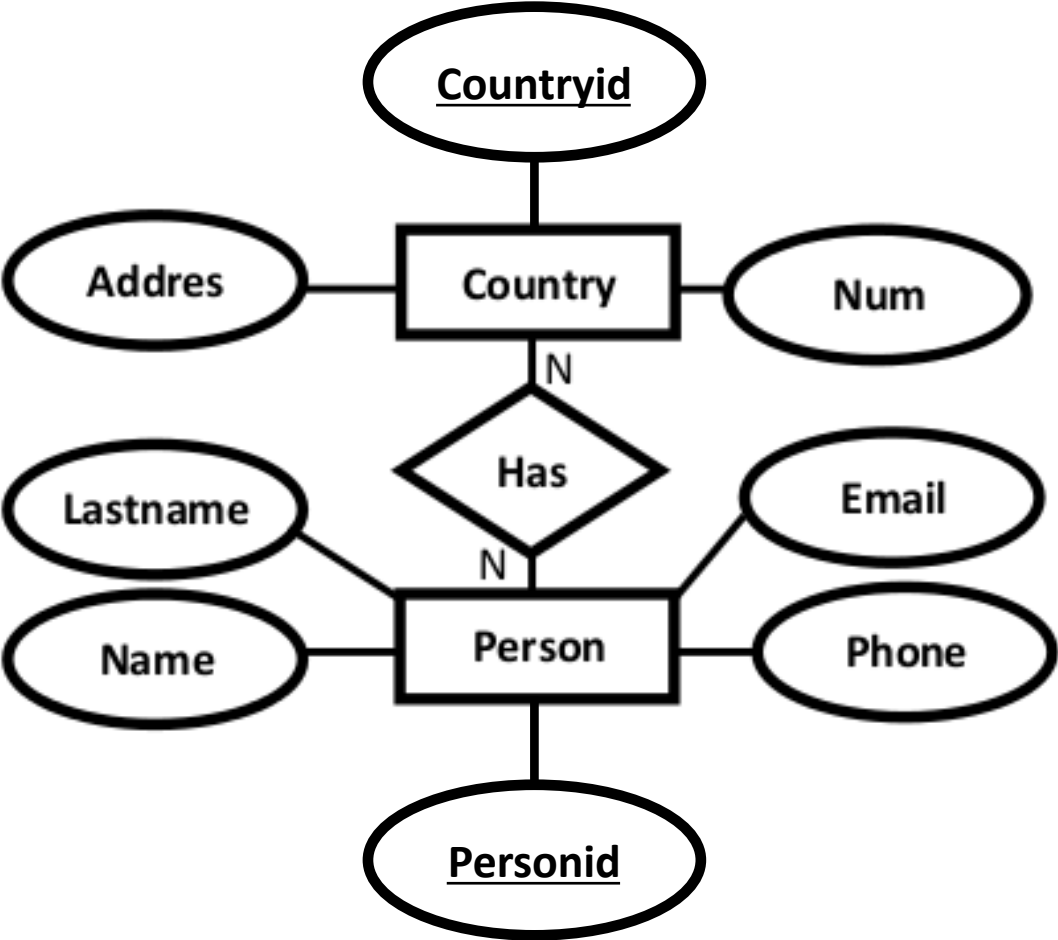
ER Modelden İlişkisel Veri Modeline Geçiş



Yandaki ER modelin şemasını oluşturalım

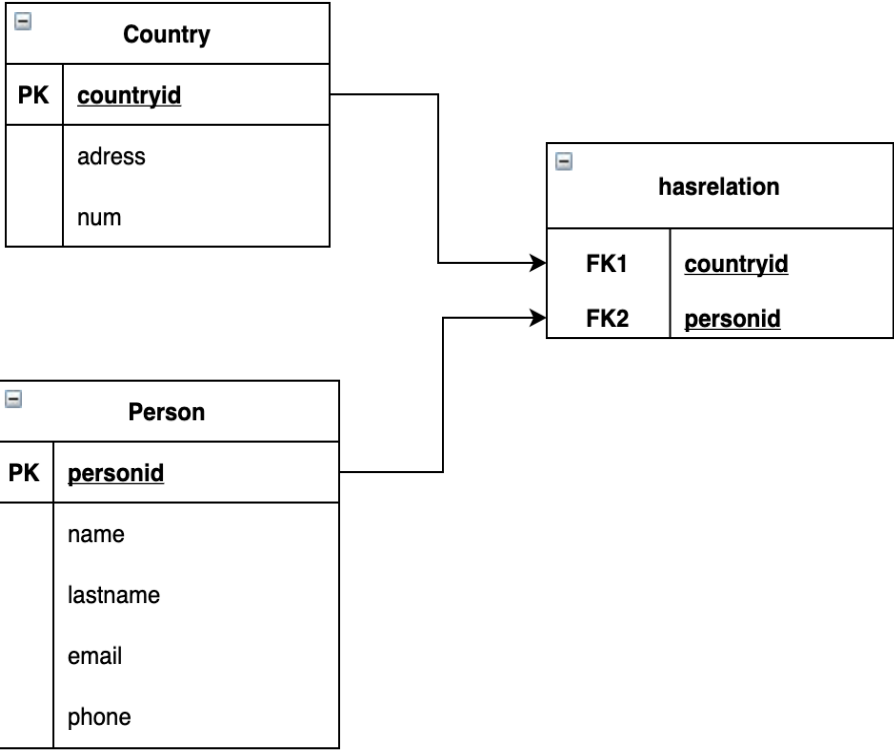
?

ER Modelden İlişkisel Veri Modeline Geçiş



ER model
ER Diagram

İlişkisel (relational)
model



Dinlediğiniz için
Teşekkürler...
İyi çalışmalar...