

Bölüm # 2

VERİ TABANI DERS NOTLARI

Veri Tabanı Tasarımı

Varlık Bağlantı ile veri modelleme (data modeling with ER)

Outline

- Veri tabanı tasarım aşamaları
- Temel Kavramlar
 - Varlık, Varlık kümesi ve nitelikleri, Bağıntı, Bağıntı kümesi, bağıntı sınırlamaları ve bağıntı türleri, Rol tanımlama
 - Anahtar nitelik
 - Zayıf Varlık Kümeleri
- Yardımcı Kavramlar
 - Genelleme
 - Kümeleme
- Örnek bir VT
- ER veri modeli ile örnekler

VT Tasarımı: 6 adım

1. Gereksinimlerin toplanması ve analizi (Requirements collection and analysis)

- VT Tasarımcısı, veri gereksinimlerini anlamak ve belgelemek için müstakbel veri tabanı kullanıcıları ile görüşmeler (interview) yapar
- Çıktı: **veri gereksinimleri**
- Uygulama için işlevsel gereksinimler (**Functional requirements**)

VT Tasarımı

2. Kavramsal şema

- Kavramsal bir tasarım
- Veri gereksinimlerini açıklamalı
- Varlık türleri nedir: açık gösterim (entity types)
- Bağıntıları (relationships) detaylı açıklama
- Kısıtlamalar (constraints): gösterim
- Üst-düzeyli veri modelinden gerçekleştirim veri modeline dönüşüm (Transformation from high-level data model into implementation data model)

VT Tasarımı

3. Mantıksal tasarım veya veri modeli eşlemesi (data model mapping)

- Result is a database schema in implementation data model of DBMS

4. Schema Refinement

- consistency, normalization

5. Fiziksel Tasarım (Physical design phase)

- Internal storage structures, file organizations, indexes, access paths, and physical design parameters for the database files specified

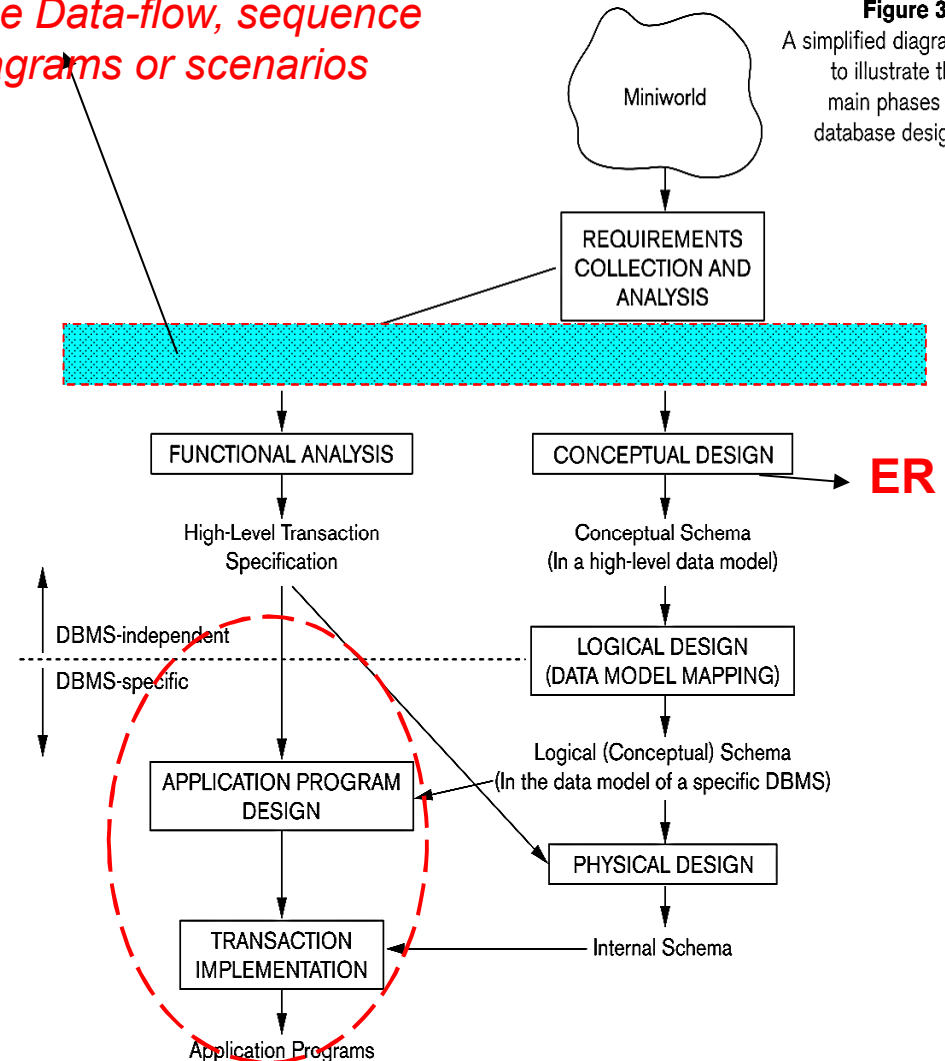
6. Security Design

- who accesses what, and how

VT Tasarım Aşamaları:Şekil ile

- VT tasarımı
 - Veri modelleme
 - Veri mühendisliği, bilişim uzmanlığı ile ilgili
- Uygulama tasarımı
 - Program akış ve arayüzlerin tasarımı
 - Yazılım mühendisliği ile ilgili

Use Data-flow, sequence diagrams or scenarios



Varlık Bağntı (Entity-Relationship, ER) Modeli

- ER model der ki: Bir veritabanı
 - Varlıklar (Entities) topluluğu ve
 - Varlıklar arası Bağntılar (Relationships) ile modellenenbilir.

Kavramsal Tasarım

- What are the **entities** and **relationships** in the enterprise?
- What information about these entities and relationships should we store in the database? **attributes - nitelikler**
- What are the **integrity constraints** or business rules that hold?
- A **database schema** in the ER Model can be represented pictorially (as an **ER diagram**).
- an ER diagram can be mapped into a relational db schema.

Gösterim

- ER modelinin gösterimi
 - Klasik ER diyagramı (*Chen* notasyonu)
 - UML diyagramı (*UML* notasyonu)
 - Yazılım geliştirme metodolojisinde kullanılır.
 - ER'daki varlık UML nesnesine karşılık gelir.
 - UML nesnesinde 3 kısım var: nesne adı, nitelikler ve operasyonlar.

ER model

- An **entity** is an object that exists and is **distinguishable** from other objects.
 - Ex: specific person, company, event, plant
- Entities have **attributes**
 - Ex: people have *names* and *addresses*
- An **entity set** is a set of **entities** of the same type that share the same properties.
 - Ex: set of all persons, companies, trees, holidays

Entity Sets: *instructor* and *student*

- instructor_ID instructor_name student-ID student_name

76766	Crick	98988	Tanaka
45565	Katz	12345	Shankar
10101	Srinivasan	00128	Zhang
98345	Kim	76543	Brown
76543	Singh	76653	Aoi
22222	Einstein	23121	Chavez
		44553	Peltier
<i>instructor</i>		<i>student</i>	

Varlık ve Varlık kümesi

- Var olan ve ayırdedilebilen her nesneye **varlık**, benzer varlıkların oluşturduğu kümeye **varlık kümesi** demiştik.
- Varlıklara ait **nitelikler** vardır. Bu nitelikler ile varlıklar birbirinden ayırdedilir.
 - *varlık kümesi: ÇALIŞAN*
 - *Ad-soyad: John Smith, bölüm:Research, maaş:40K*
- Her niteliğin bir **değer alanı**, yani olurlu değerlerinin tümünü içeren bir küme (**domain**) vardır.
- Niteliğin özellikleri: türü, değer aralığı, formatı.
- Bir varlık kümesindeki bütün varlıklar aynı niteliklere sahip

Key (Anahtar) Attribute(s)

- A **super key** of an entity set is a set of one or more attributes whose values **uniquely** determine each entity.
 - *{ ID, name }* is a super key of *instructor*
- A **candidate key** of an entity set is a **minimal** super key
 - *ID* is candidate key of *instructor*
 - *course_id* is candidate key of *course*
- Although several candidate keys may exist, one of the candidate keys is **selected** to be the **primary key**.

Relationship (Bağlantı) Sets

- A **relationship** is an association among several entities
- Example:
 - 44553 (Peltier) *advisor* 22222 (Einstein)
 - *student* entity relationship set *instructor* entity
- A **relationship set** is a mathematical relation among $n \geq 2$ entities, each taken from entity sets
- $\{(e_1, e_2, \dots, e_n) \mid e_1 \text{ in } E_1, e_2 \text{ in } E_2, \dots, e_n \text{ in } E_n\}$
where $(e_1, e_2, \dots, e_n)$ is a relationship
 - Example:
- $(44553, 22222)$ in *advisor*

Bağıntı ve Bağıntı kümesi

- **Bağıntı**: iki ya da daha çok varlıklar arasındaki «olayı/ilişkiyi» tanımlayan kavram:
- Aynı **türdeki** bağıntıların tümüne birden **bağıntı kümesi** denilir.
- relationships can have their own attributes.
 - Öğrenci dersi alıyor ise
DERS ile ÖĞRENCİ arasında ALIYOR bağıntısı kurulur.
Notu niteliği, ALIYOR bağıntısına ait bir niteliktir.
- Same entity set can participate in different relationship sets, or in different “roles” in the same set.

Relationship Set *works_for*

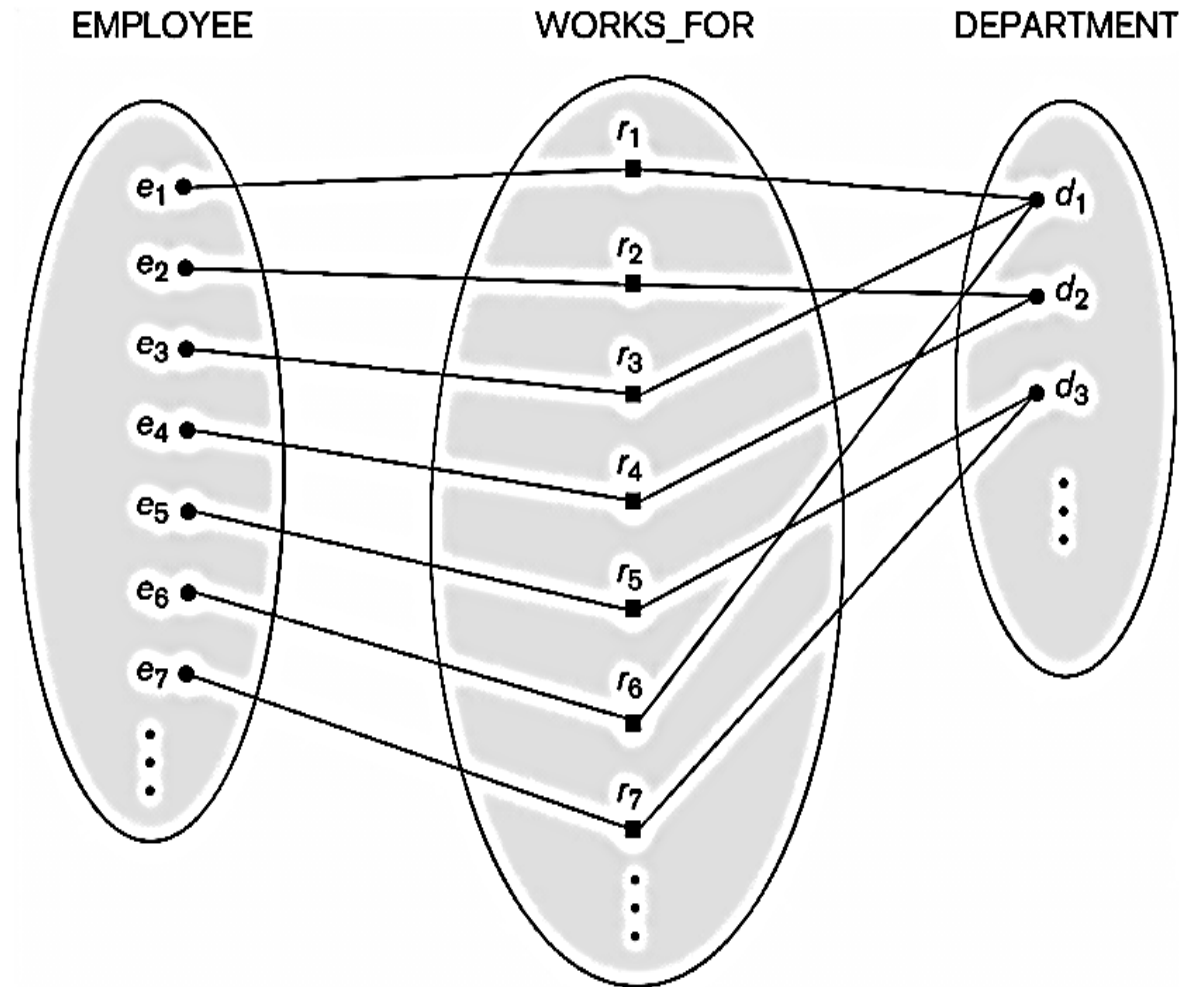


Figure 3.9

Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

Degree of a Relationship Set

- **binary relationship**
 - involve two entity sets (or degree two).
 - most relationship sets in a database system are binary.
- Relationships between more than two entity sets are rare. Most relationships are binary.
- Example: *students* work on research *projects* under the guidance of an *instructor*.
 - relationship *proj_guide* is a **ternary relationship** between *instructor*, *student*, and *project*

Attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.
 - Example:
instructor = (ID, name, street, city, salary)
course = (course_id, title, credits)
- **Domain** – the set of permitted values for each attribute
- **Attribute types:**
 - **Simple** and **composite** attributes.
 - **Single-valued** and **multivalued** attributes
 - Example: multivalued attribute: *phone_numbers*
 - **Derived** attributes: Can be computed from other attributes
 - Example: age, given *date_of_birth*

Composite Attributes

- Name: fname, middle initial, last name
- Address: **street**, city, state, postal-code
- **street**: street_number, street name, apt_no

Mapping Cardinality Constraints

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- For a binary relationship set, the mapping cardinality must be one of the following types:
 - One to one
 - One to many
 - Many to one
 - Many to many

Bağıntı Küme sınırlamaları

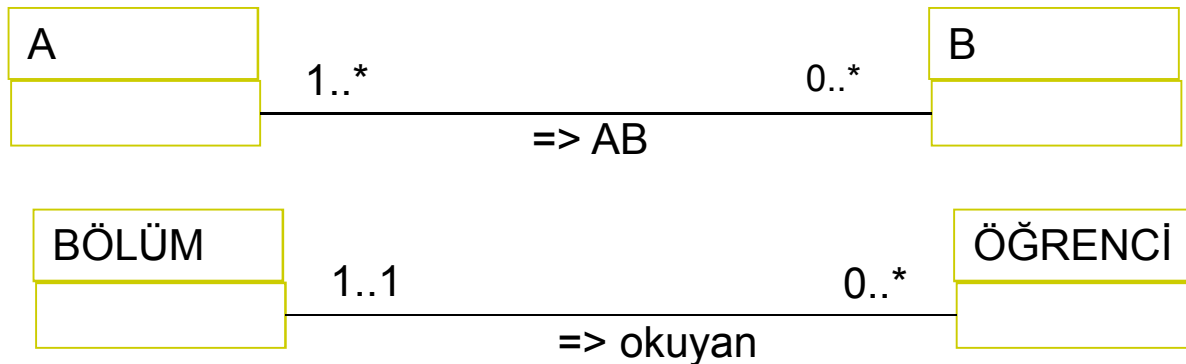
Bağıntı ismi (*relationship name*): manaya uygun bir isim

Bağıntındaki varlıklar (*participating entity types*): 2 veya daha çok, aynı veya farklı varlıklar arasında

Bağıntının derecesi (*degree of a relationship*): 2,3,...

Kaç-kaçlık olduğu (*cardinality ratio*): 1-1, 1-N, N-1, N-N

ROL tanımlama: Bağıntındaki varlıkların bağıntındaki işlevlerini belirleyen bir rolleri vardır. Bunun açık, anlaşılır olmadığı durumlarda belirtilmesi gerekir.



Bağıntı Örnekleri: Many to one, Many to many

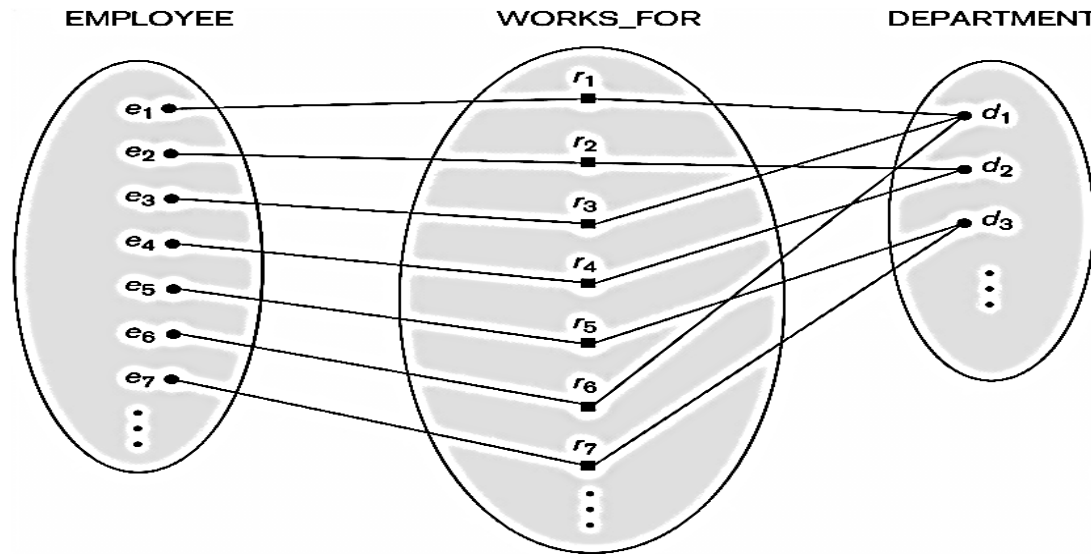


Figure 3.9
Some instances in the WORKS_FOR relationship set, which represents a relationship type WORKS_FOR between EMPLOYEE and DEPARTMENT.

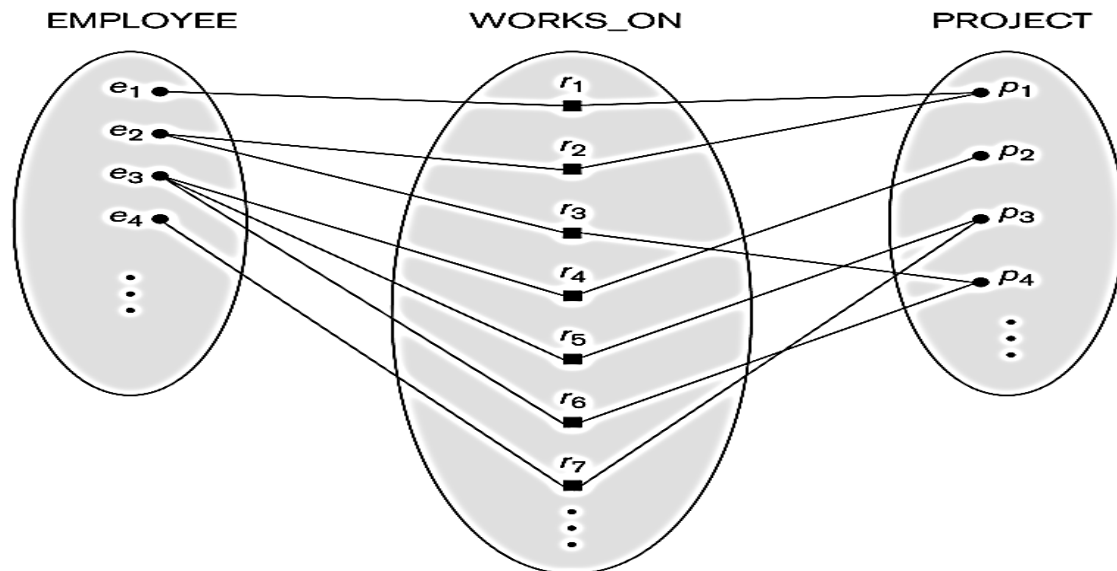
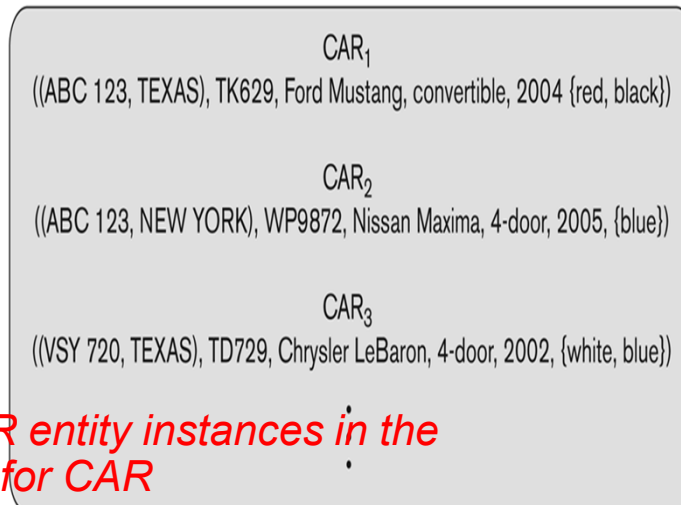
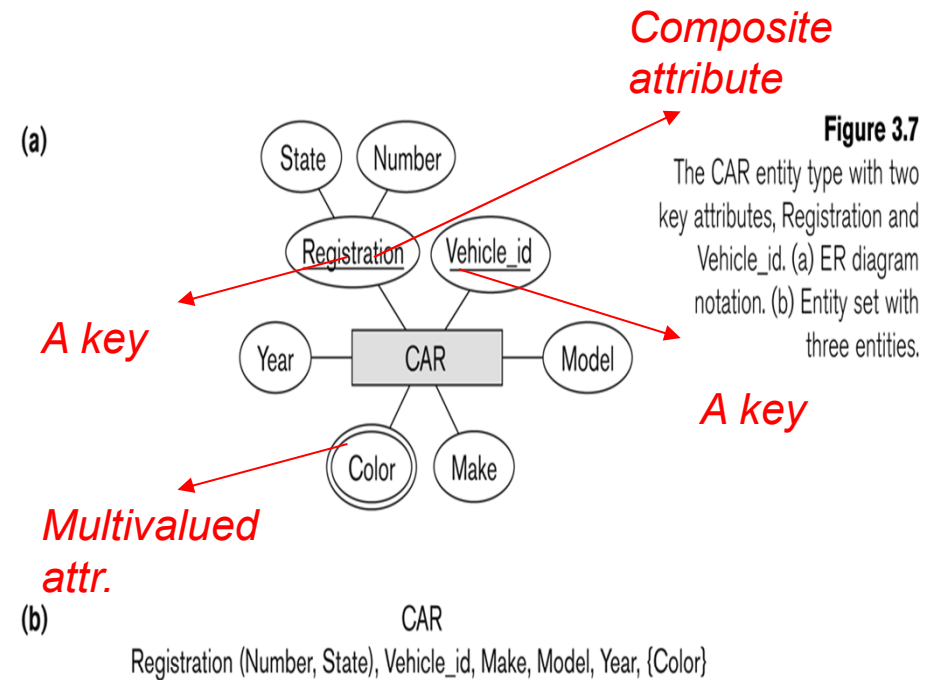
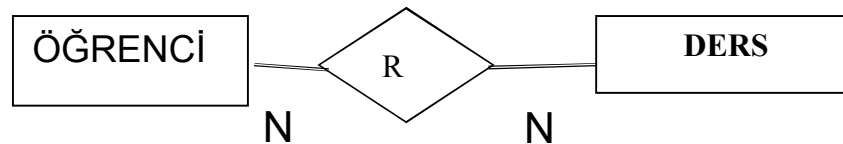
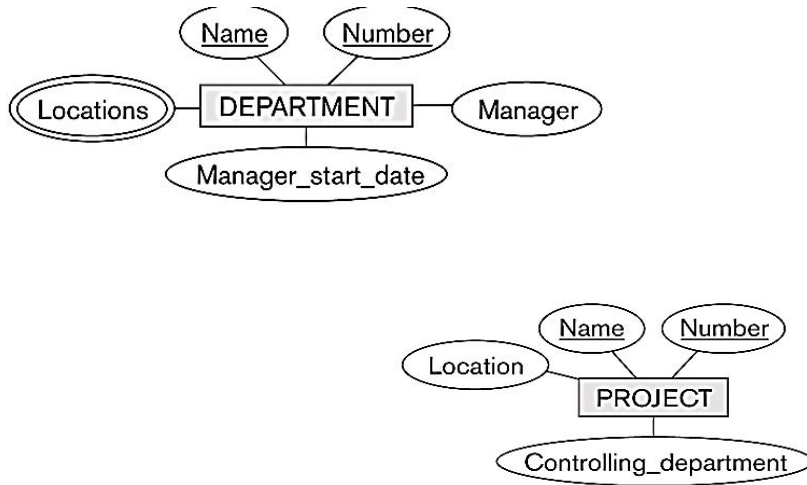


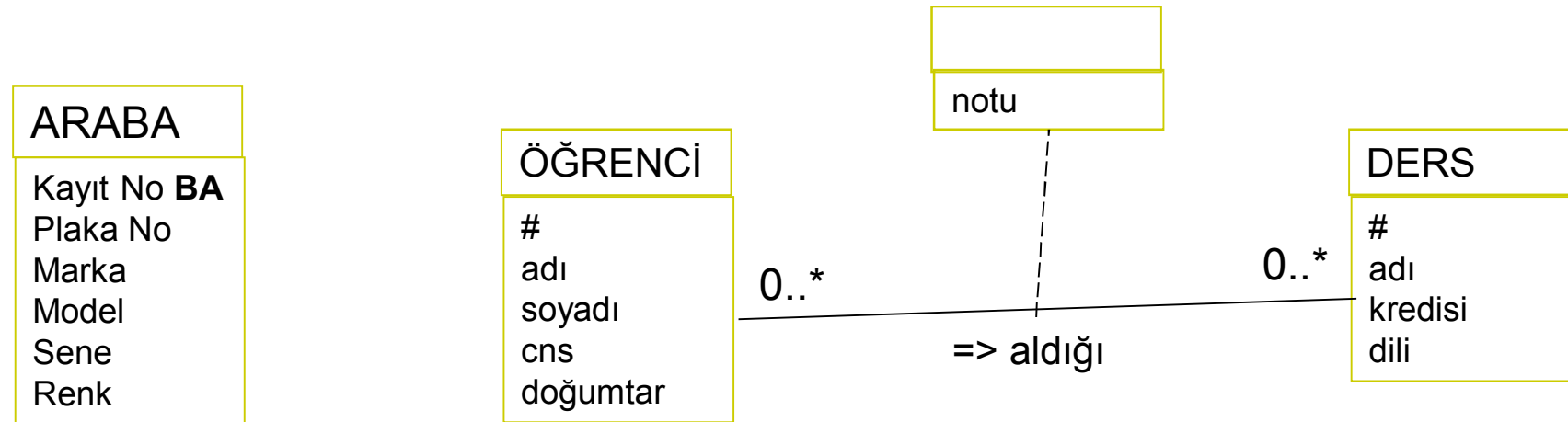
Figure 3.13
An M:N relationship, WORKS_ON.

Chen notasyonu



three CAR entity instances in the entity set for CAR

UML ile Varlık-Bağıntı Örnekleri



ER design

● <u>Entity 1</u>	<u>Card. ratio</u>	<u>Entity 2</u>
● STUDENT	-----	TC-KİMLİK-KARTI (ID-CARD)
● STUDENT	-----	TEACHER
● CLASSROOM	-----	DESK (SIRA)
● COUNTRY	-----	CURRENT-PRESİDENT
● COURSE	-----	TEXTBOOK
● ITEM	-----	ORDER
● STUDENT	-----	COURSE
● COURSE	-----	INSTRUCTOR
● INSTRUCTOR	-----	DEPARTMENT
● CITY	-----	COUNTRY

How about doing an ER design interactively on the board?

- Applications to be modeled: