– Fundamentals of Database Systems

Chapter 2:
Data Modeling using the Entity Relationship Model

eSPIN University

Overview

We learned that:

1. A DBMS will allow us to create and manage databases.
2. We can query the database (retrieve data from tables in the database).
3. We can update the database (change, add, or delete data from tables in the database).
4. We can use SQL to perform operations on the database

This chapter

- How to design a database using the ER-Model

- Three major components:

1. Entities
2. Attributes
3. Relationships

Database Design

- Objective: Decide on the structure of the database

- Three main steps:
1. Requirements collection and analysis
2. Conceptual design
3. Logical and physical design

1. Requirements collection and analysis

- **Decide on the requirements of the database system**

- **Ask questions such as:**

1. What is going to be stored?
2. How is it going to be used?
3. Who should access the data?

- **Result: data requirements**
- Functional requirements of the application

2. Conceptual design

- **A high-level description of the database**

- Detailed so that technical people can understand it
- But, not too complicated that non-technical people can't understand it

- From a set of requirements to a conceptual design.

Using the ER-Model

This chapter

3. Logical and Physical design

- **Logical design or data-model mapping:**
- Result is a database schema in implementation data model of DBMS

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

Part 2: Main components of the ER-Model

ER-Diagram *exam*

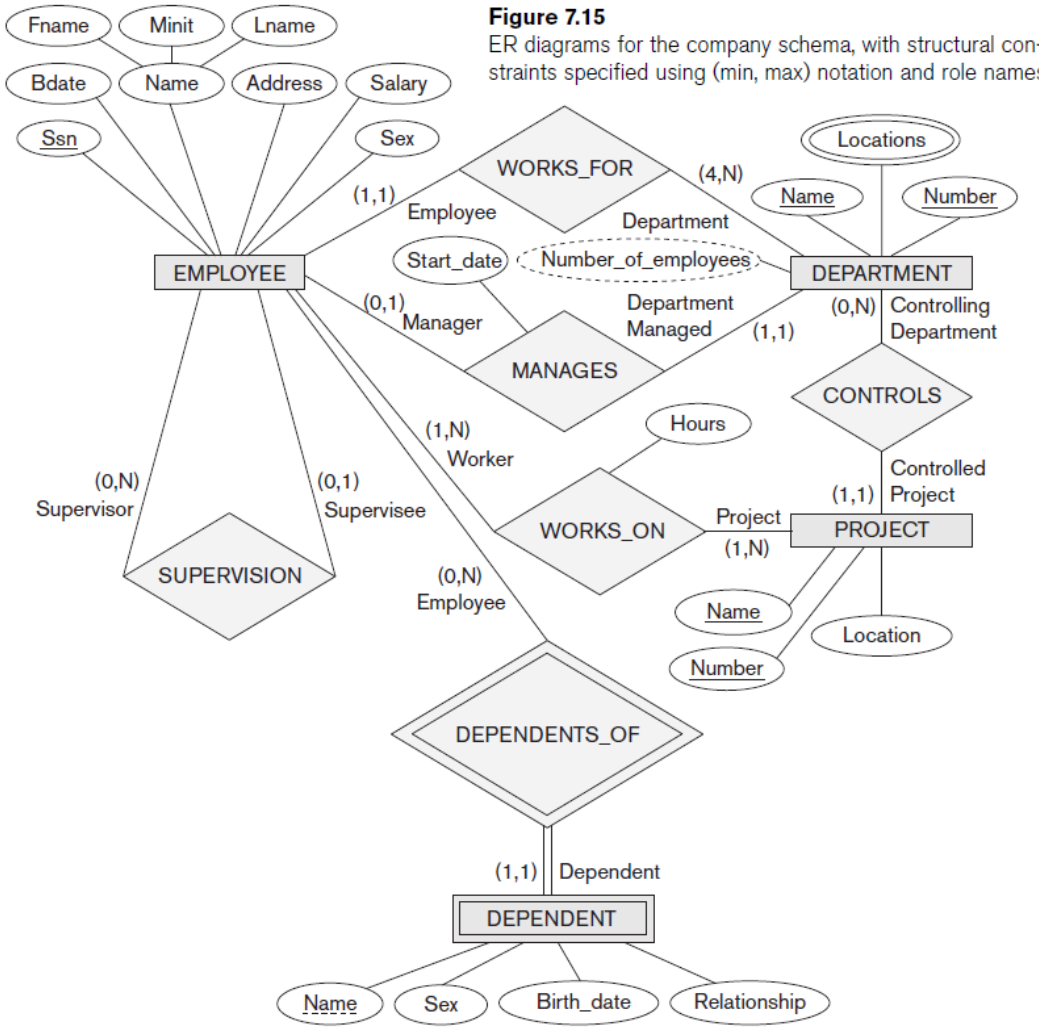


Figure 7.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

ER-Model components

- **Three major components:**

1. **Entities**
2. **Attributes**
3. **Relationships**

1. Entities

- **Basic objects in ER model**
- A “thing” in the real world with independent existence
- Physical existence: person, car, house, employee, ...
- Conceptual existence: company, job, course, ...
- Each entity must have a set of attributes

1. Entities, entity type, and entity sets

- **Entities:** The individual objects, which are members of entity sets
- Ex: A specific person or product

- **Entity type:** A collection of entities that have the same attributes. Should be a noun

- **Entity sets:** Represent the sets of all possible entities

1. Entities

- **Entities *example*:**

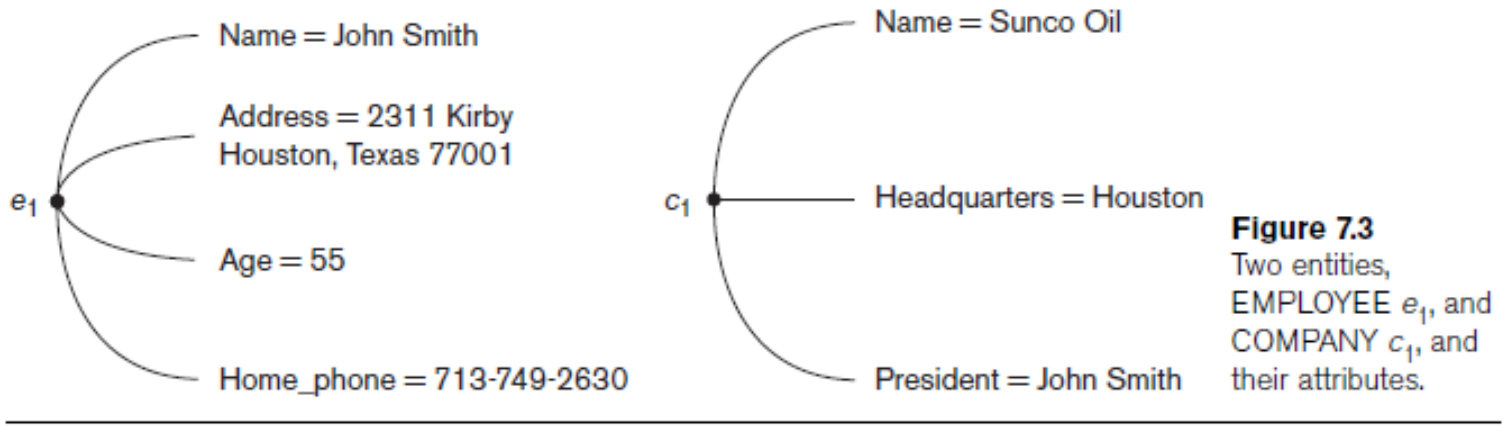


Figure 7.3
Two entities, EMPLOYEE e_1, and COMPANY c_1, and their attributes.

1. Entities

- **Entities, entity type and entity set example:**

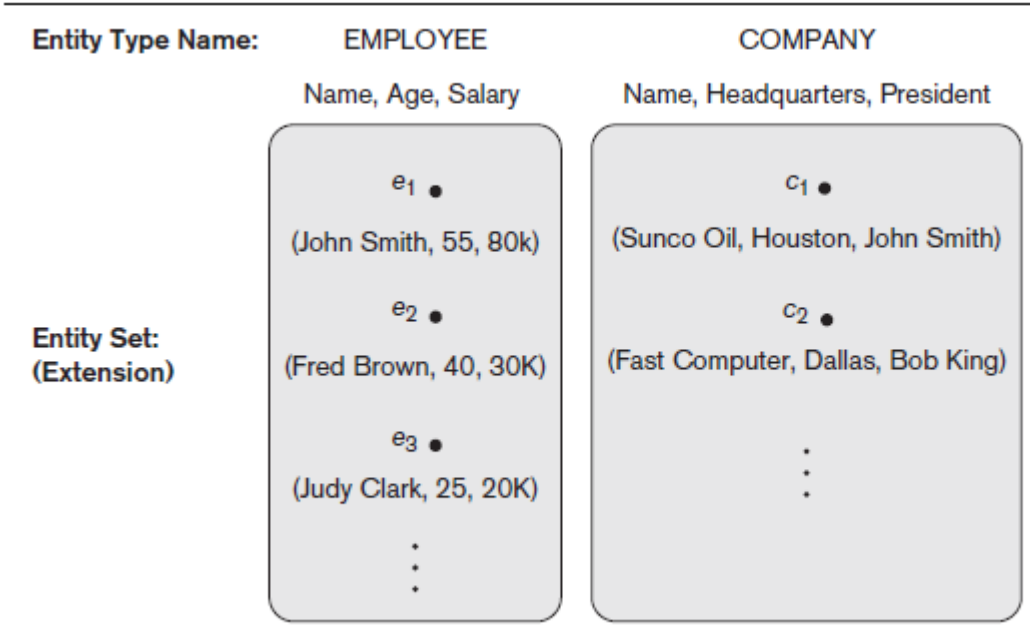


Figure 7.6
Two entity types, EMPLOYEE and COMPANY, and some member entities of each.

1. Entities

- **Basic objects in ER model**
- A “thing” in the real world with independent existence
- Physical existence: person, car, house, employee, ...
- Conceptual existence: company, job, course, ...

Each entity must have a set of attributes

2. Attributes

- **Properties used to describe entities**
- **Example #1:**
- Entity Name: EMPLOYEE
  Attributes:
- name, age, address, salary, job

- **Example #2:**
- Entity Name: CAR
  Attributes:
- name, maker, VIN number

2. Attributes

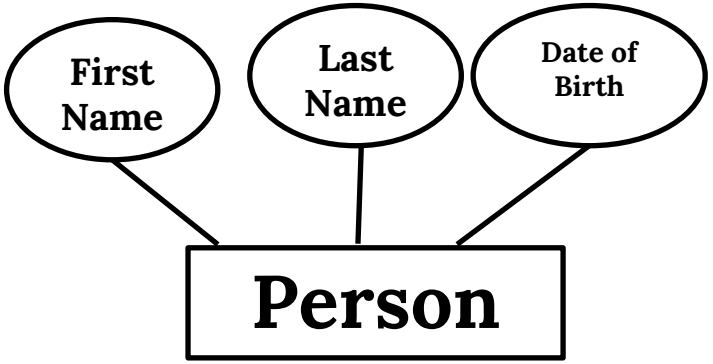
- **Properties used to describe entities**
- A particular entity will have a value for each of its attributes

  name: “John Smith”
  age: 55
  address: 2311 Kirby, Houston, TX 77001
  salary: $2000
  job: Accountant
- The values are the actual data stored in the database

Entities and Attributes

Entity

Graphical representation



An entity is represented by a rectangle

An attribute is represented by a circle

2. Attributes

- **Types of attributes:**

1. **Simple (Atomic) Vs. Composite Attributes**

Composite attributes can be divided into smaller subparts

- Subparts represent basic attributes with independent meanings
- Ex. Address, FullName, ...

Simple attributes cannot be divided

Ex. City, ZipCode, FirstName, LastName, ...

2. Attributes

- **Types of attributes:**

2. **Single-valued Vs. Multi-valued Attributes**

Single-valued attributes have a single value for a particular entity
- Ex. **Age**, **HairColor**, **Name**, ...

Multi-valued attributes may have a number of values for a particular entity
- Ex. **Skill**, **CollegeDegrees**,

2. Attributes

- **Types of attributes:**

3. Stored Vs. Derived Attributes

In some cases, one attribute (or more) is related to another

Ex. **Age** and **BirthDate**

A **derived attribute** can be determined from a **stored attribute**

Ex. **BirthDate** is a stored attribute, while **Age** is a derived attribute

2. Attributes

- **Types of attributes:**

4. **Null Values**

   Used in cases where attributes may not have values for a specific entity

   1. Because the value is not applicable

      Ex. **ApartmentNo**, **CollegeDegrees**, ...

   2. Or because the value is unknown

      Ex. **PhoneNumber** (may not exist)

      Ex. **Height** *of a person* (exists)

2. Attributes

- **Types of attributes:**

5. **Complex Attributes**

   A combination of composite and multivalued attributes

   ( ) are used to represent components of a composite attribute

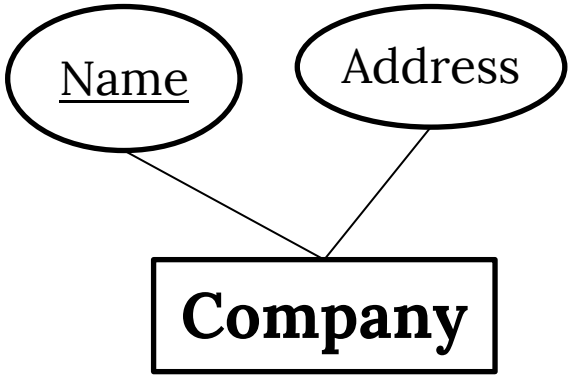
   { } are used to represent multivalued attributes

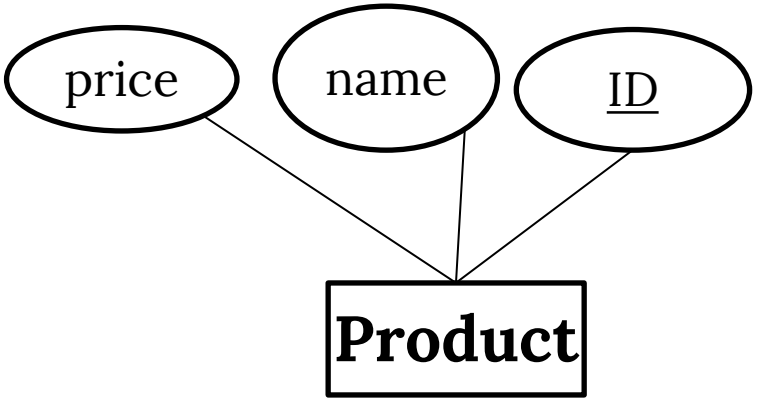
2. Attributes

- **Key constraint:**
- Attributes whose values are distinct for each individual entity in entity set
- **Key attribute:** An attribute that has to be unique and can be used to identify an entity (**Primary Key** in the database)

- **Value sets (or domain of values):**
- Specifies set of values that may be assigned to that attribute for each individual entity

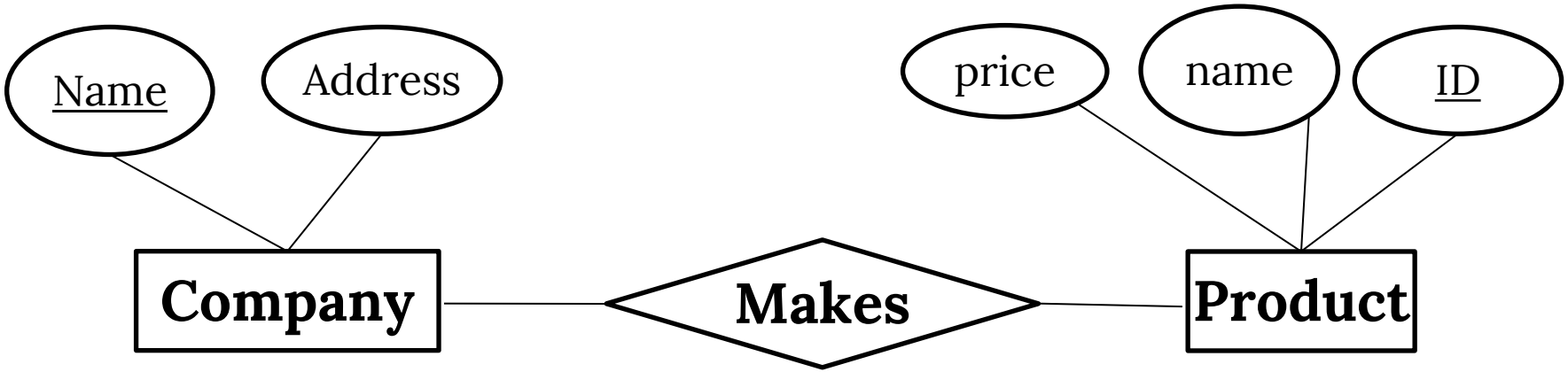
Entities and Attributes





- **Is there a relationship between company and product?**

3. Relationship



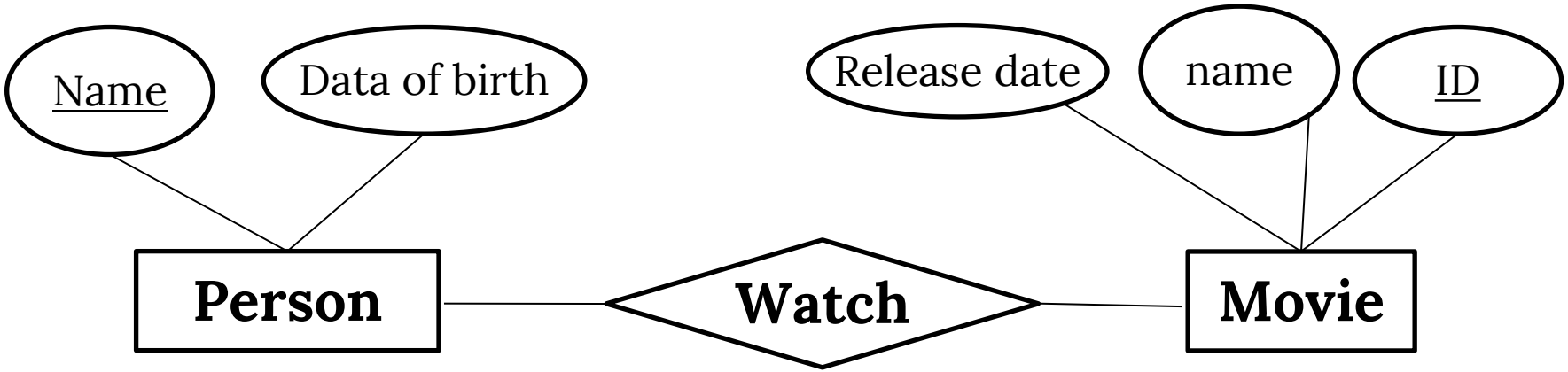
- Is there a relationship between company and product?
- Yes, a company makes products

3. Relationship

- A relationship between two (or more) entities

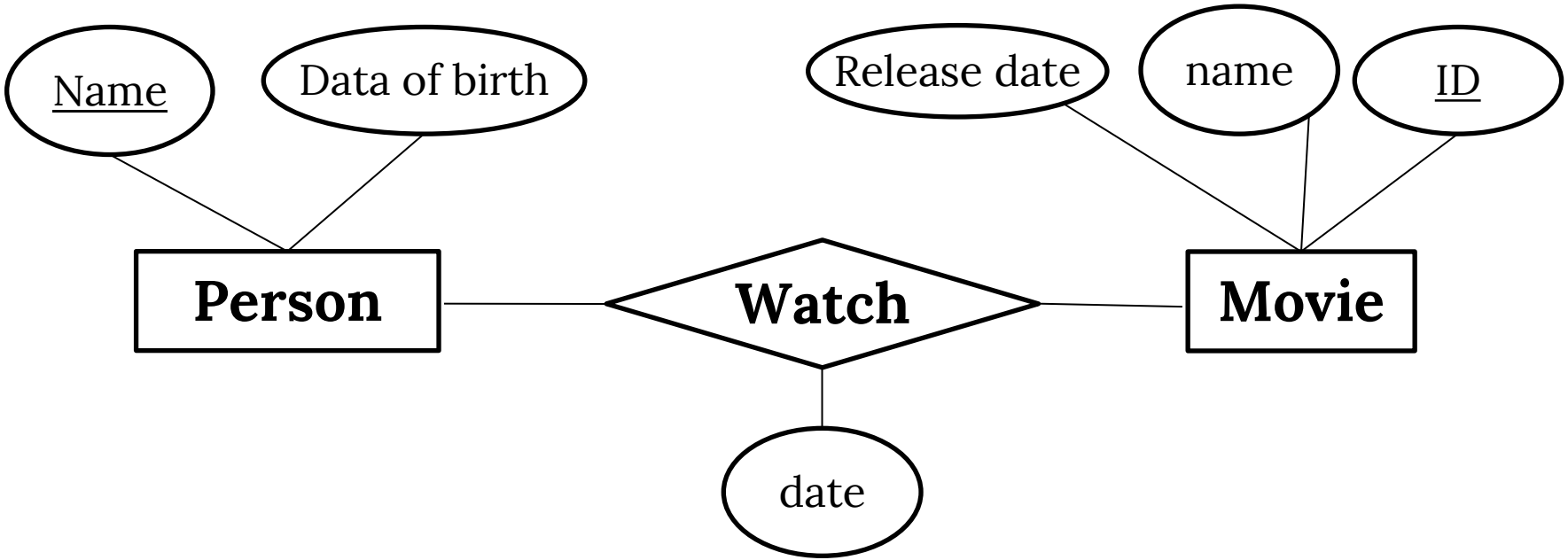
1. A relationship can have attributes

3. Relationship: A relationship can have attributes



- A person can watch a movie more than once
- What if we want to save the date each time a person watches a movie?

3. Relationship: A relationship can have attributes



3. Relationship

- **A relationship between two (or more) entities**

2. **Degree of relationship:**

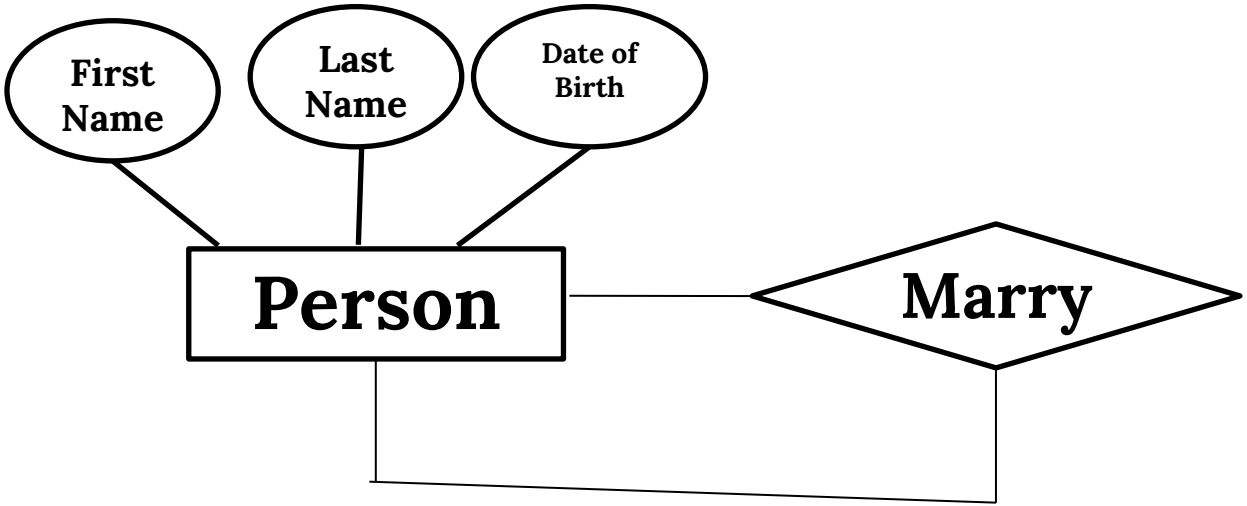
Number of participating entity types:
Binary: (two participating entity types) (of degree 2)
Ternary: (three participating entity types) (of degree 3)

3. Relationship

- **A relationship between two (or more) entities**

3. **Roll names:** Specify the role that an entity plays in each relationship instance. –Should to be a verb, and read from left to right
4. **Recursive relationship:** When an entity is in a relationship with another entity of the same entity type.

3. Relationship: Recursive relationship



3. Relationship: Recursive relationship

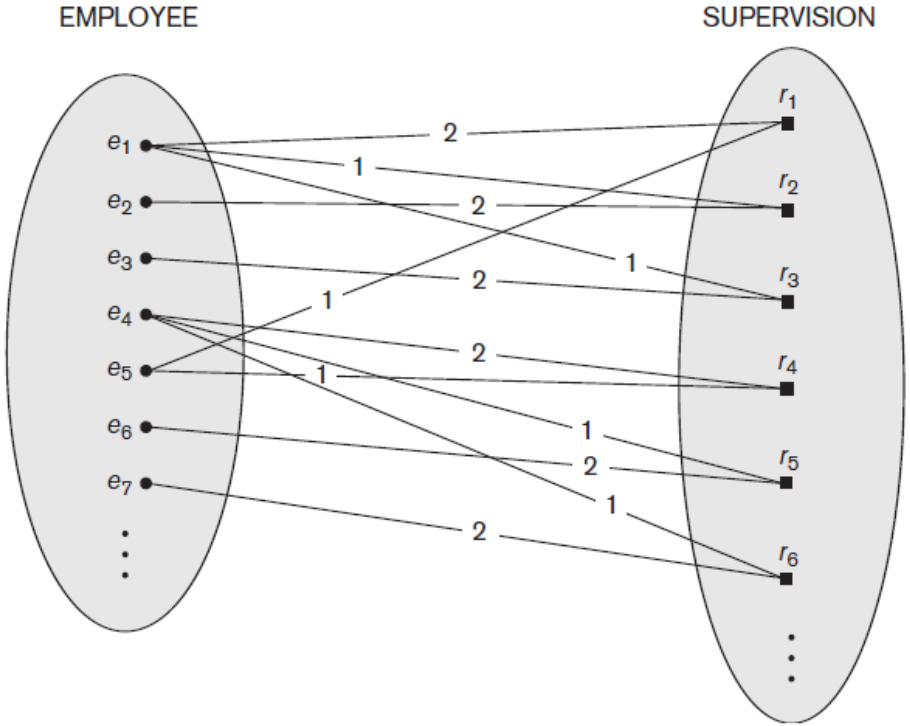


Figure 7.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

3. Relationship

- **A relationship between two (or more) entities**

5. **Participation constraint:**

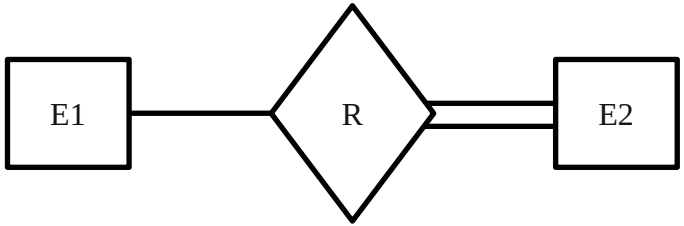
a. **Total participation:** Indicate if an entity in one side **must be** related to an entity in the other side

Ex: an EMPLOYEE must WORKS_FOR a DEPARTMENT

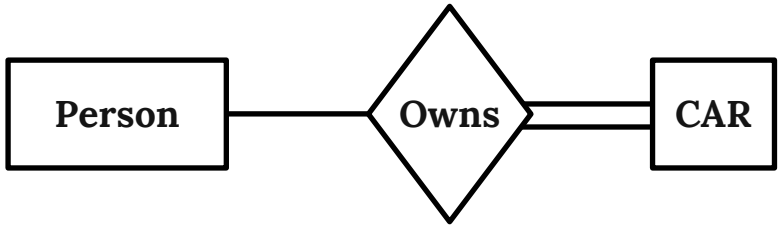
b. **Partial participation:** Indicate if an entity in one side **does not have to be** related to an entity in the other side

Ex: an EMPLOYEE does not have to MANAGE a DEPARTMENT

3. Relationship: Participation constraint

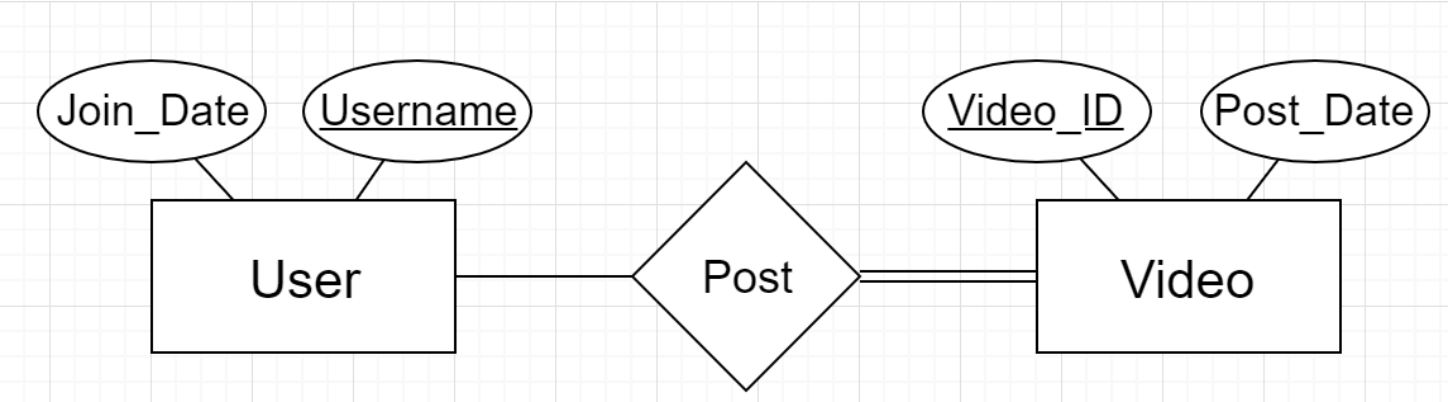


Total Participation of E2 in R, Partial Participation of E1 in R



Each car has to be owned by a person but not every person has to own a car

3. Relationship: Participation constraint



Each video has to be posted by a user, but each user does not have to post videos

3. Relationship

- **A relationship between two (or more) entities**

6. **Cardinality Ratio for Binary Relationship:**
- The number of entities to which another entity can be associated via a relationship set

- **Four Types:**

a. One-to-one (1:1)
b. Many-to-many (N:M)
c. One-to-many (1-M)
d. Many-to-one (M-1)

3. Relationship: Cardinality Ratio

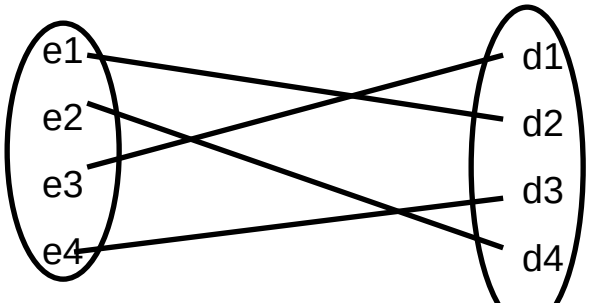
a. One-to-one (1:1):

An entity in one set is associated with at most one entity in another

Ex: EMPLOYEE MANAGES DEPARTMENT

An employee can manage one and only one department and the department is managed by one and only one employee

EMPLOYEE MANAGES DEPARTMENT

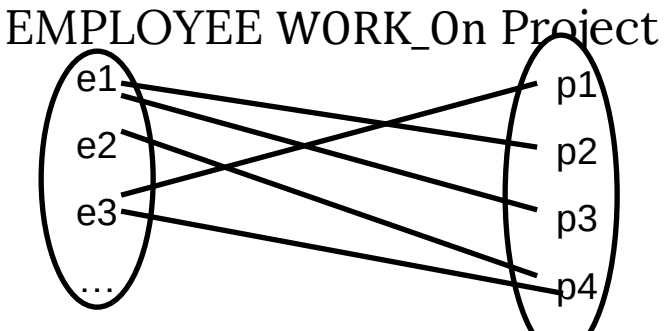


3. Relationship: Cardinality Ratio

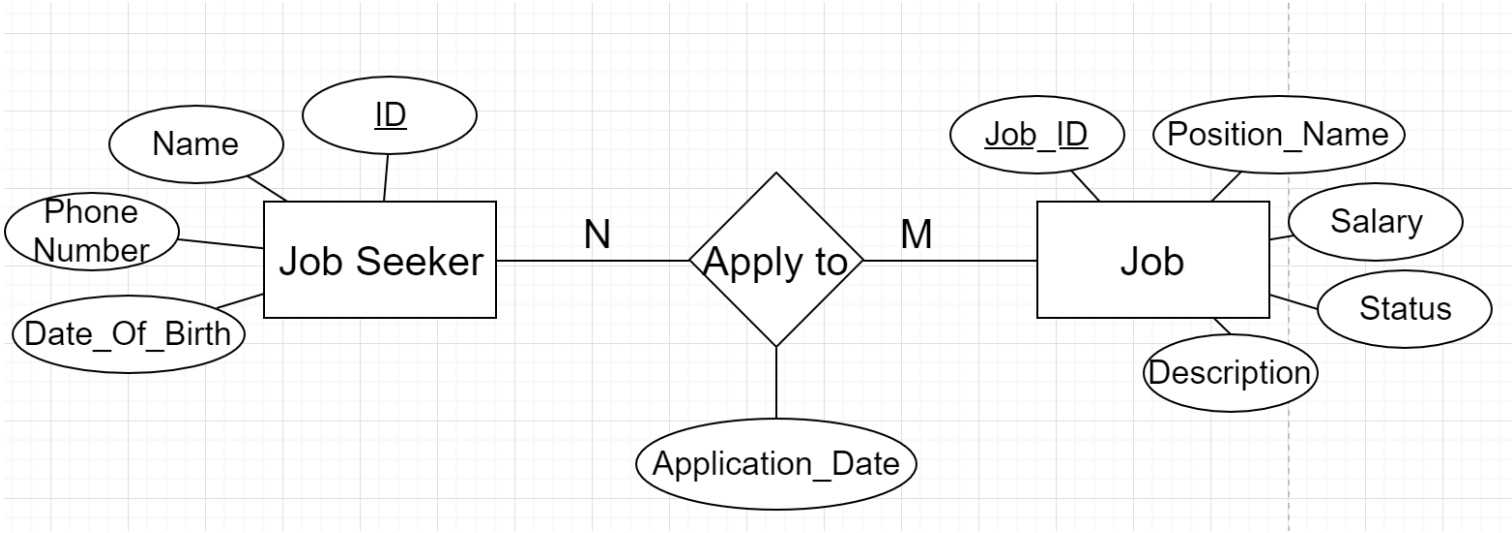
b. Many-to-many (N:M):
entities of either set may be associated with any number of entities in the other

Ex: EMPLOYEE WORK_On Project

An employee can work in many projects while each project can have many employees working on it.



3. Relationship: Cardinality Ratio (N-M)



Each job seekers can apply to many jobs and each job can be applied to by many job seekers

Partial participation on both sides

3. Relationship: Cardinality Ratio

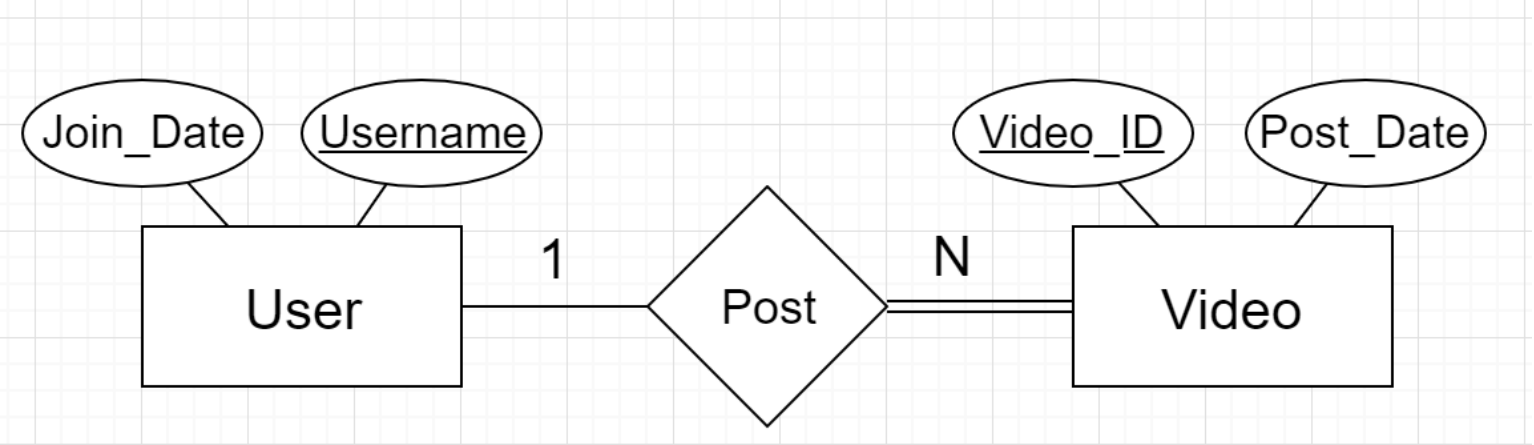
c. One-to-many (1:M):
An entity in the first set is associated with 0 or more entities in the second set, but an entity in the second set can be associated with at most one entity in the first

Ex: CUSTOMER COMPLETES ORDER

A customer complete many orders but each order is completed by one order

d. Many-to-one (M:1): Similar to one-to-many but the reversal

3. Relationship: Cardinality Ratio (1-N)

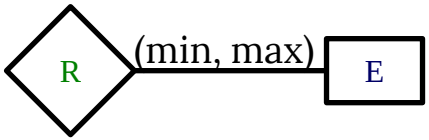


A user can post many video but each video can only by posted by one user *Notice the "1" and "N"*

A video has to be posted by a user but a user does not have to post a video (See the partial and total participation slide)

Part 3: Additional Concepts

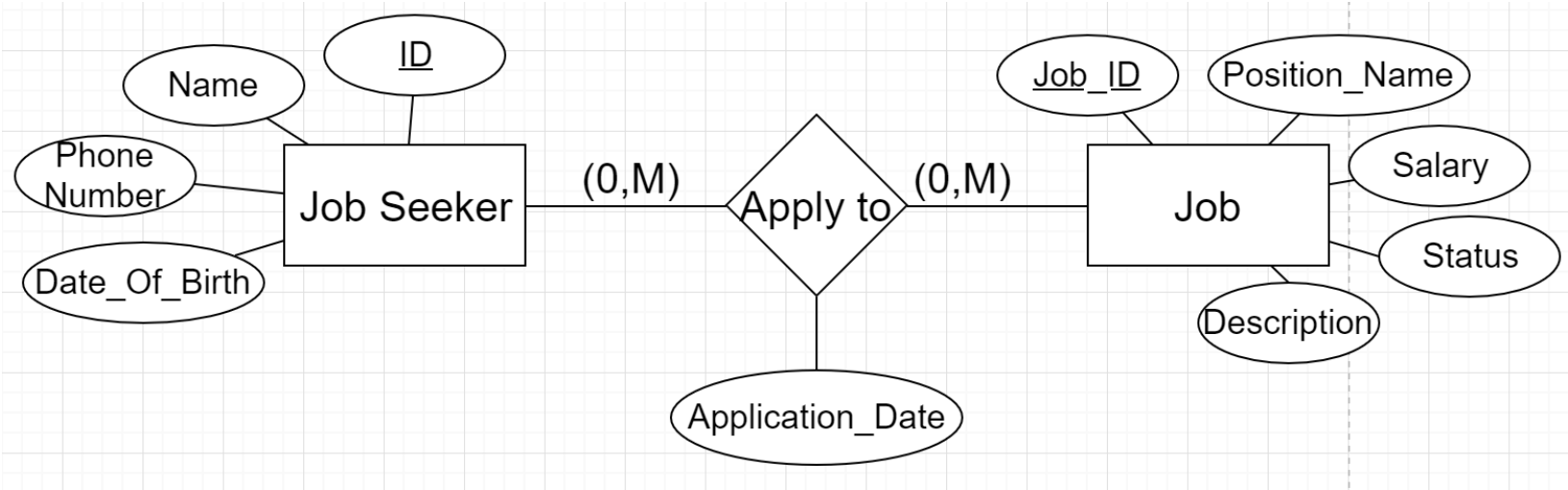
Structural constraints (min, max)



Structural Constraint
(min, max) on Participation of E in R

Replaces: 1) cardinality ratio (1:1, 1:N, M:N) and 2) single/double line notation for participation constraints

Structural constraints (min, max)



A job seeker can be to either “zero” or many jobs and the job can be applied to by zero or many job seekers.

Weak Entity Types

- **Do not have key attributes of their own**
- Identified by being related to specific entities from another entity type

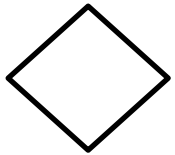
- **Identifying relationship:** Relates a weak entity type to its owner

Always has a total participation constraint

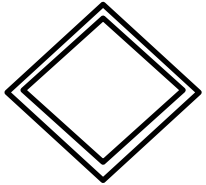
ER Diagram Notations

Entity

Weak Entity



Relationship



Identifying Relationship

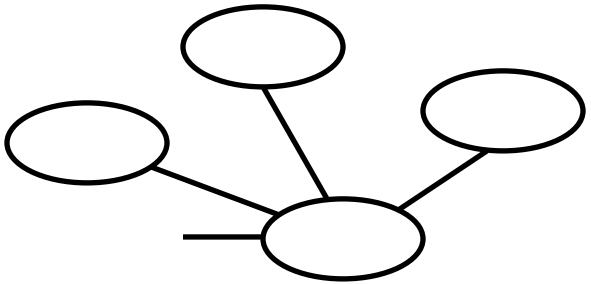
Attribute

Key Attribute

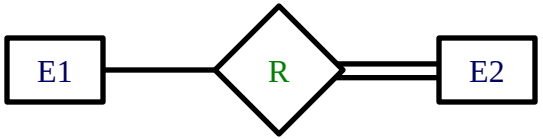
Derived attribute

Multivalued

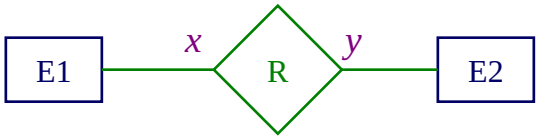
ER Diagram Notations



Composite attributes



Total Participation of E2 in R, Partial Participation of E1 in R



Cardinality Ratio *x*:*y* for E1, E2 in R

Examples

1. ER diagram for a Company schema

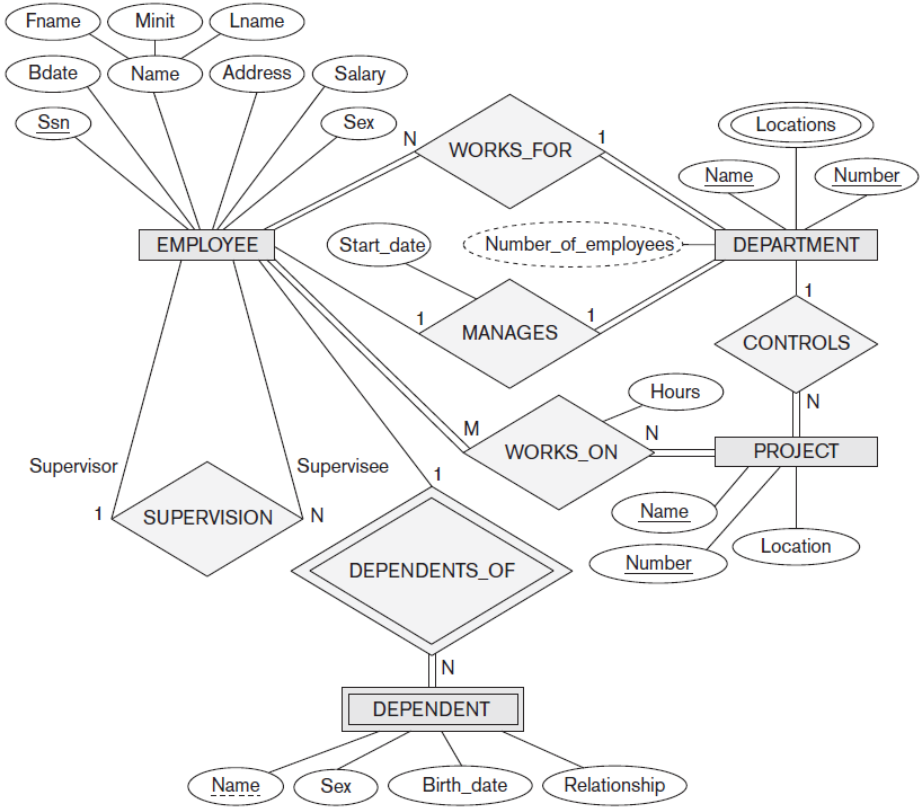


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.

Examples

2. Company schema with structural constraints specified using (min, max) notations

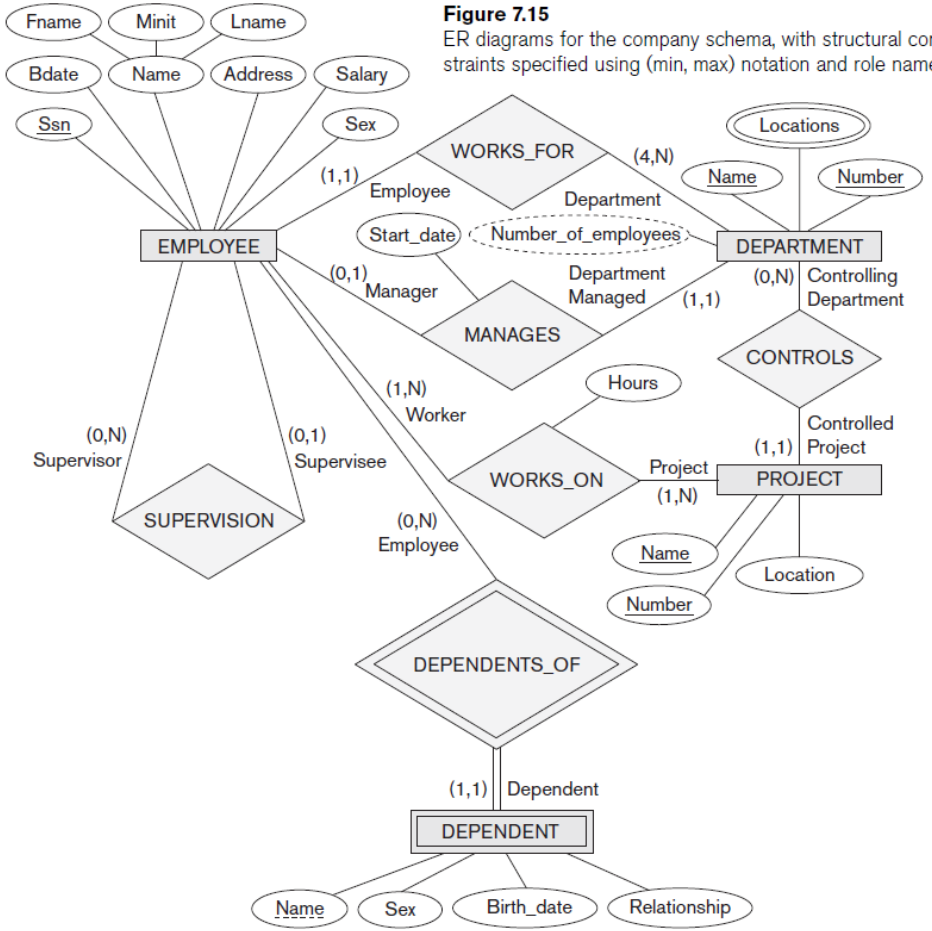
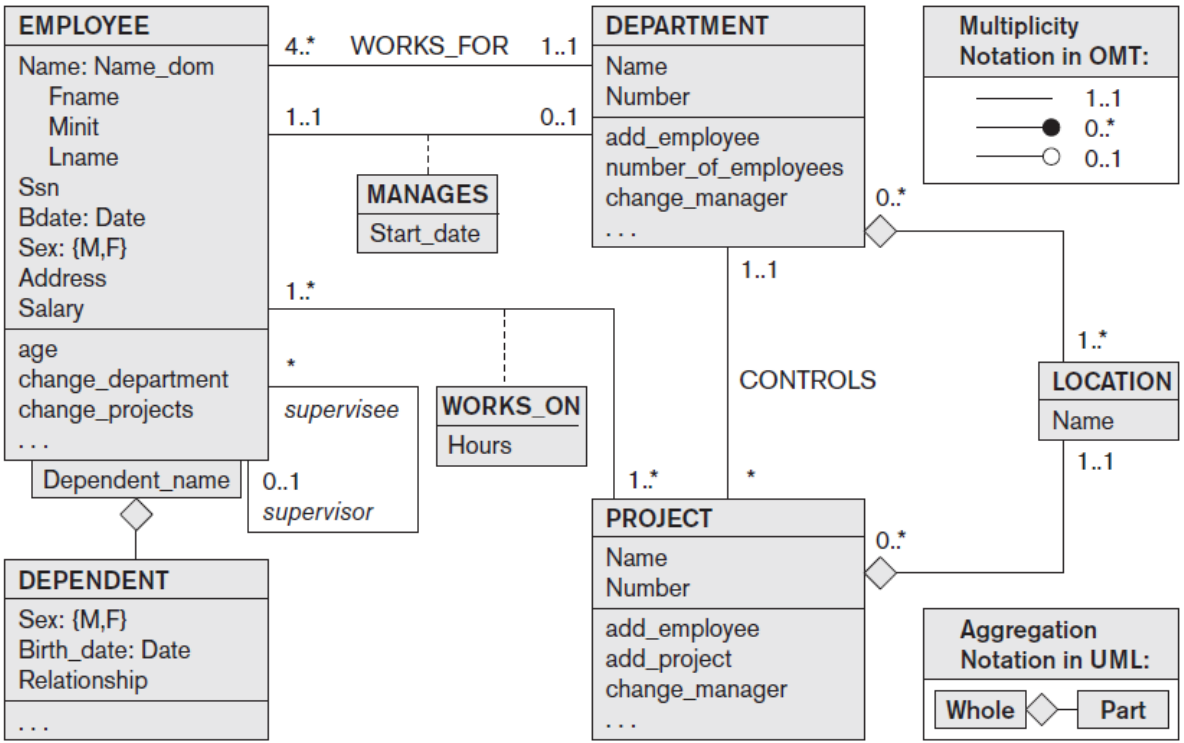


Figure 7.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

UML Class diagram

- An alternative to the ER model.

Figure 7.16
The COMPANY conceptual schema in UML class diagram notation.



Exercise

- Design an ER diagram for a website that sells book -use structural constraints (min, max) and participation constraints (total or partial).
- Website has information about the books such as the name of the book, the ISBN of the book, number of copies sold, and the publication year.
- The website also save the information about the customers who have bought books from the website. The information include: the full name, an ID for the customer, and the phone number
- Each time a customer buys a book, the date of the sale is saved.
- The website now wants to save the information about the publishers.
- The publishers information are: name, unique id, location (can be more than one).
- A publisher can publish zero or more books but each book can be published by one and only one publisher.