

Database Systems

Lecture #03

Sang-Wook Kim
Hanyang University

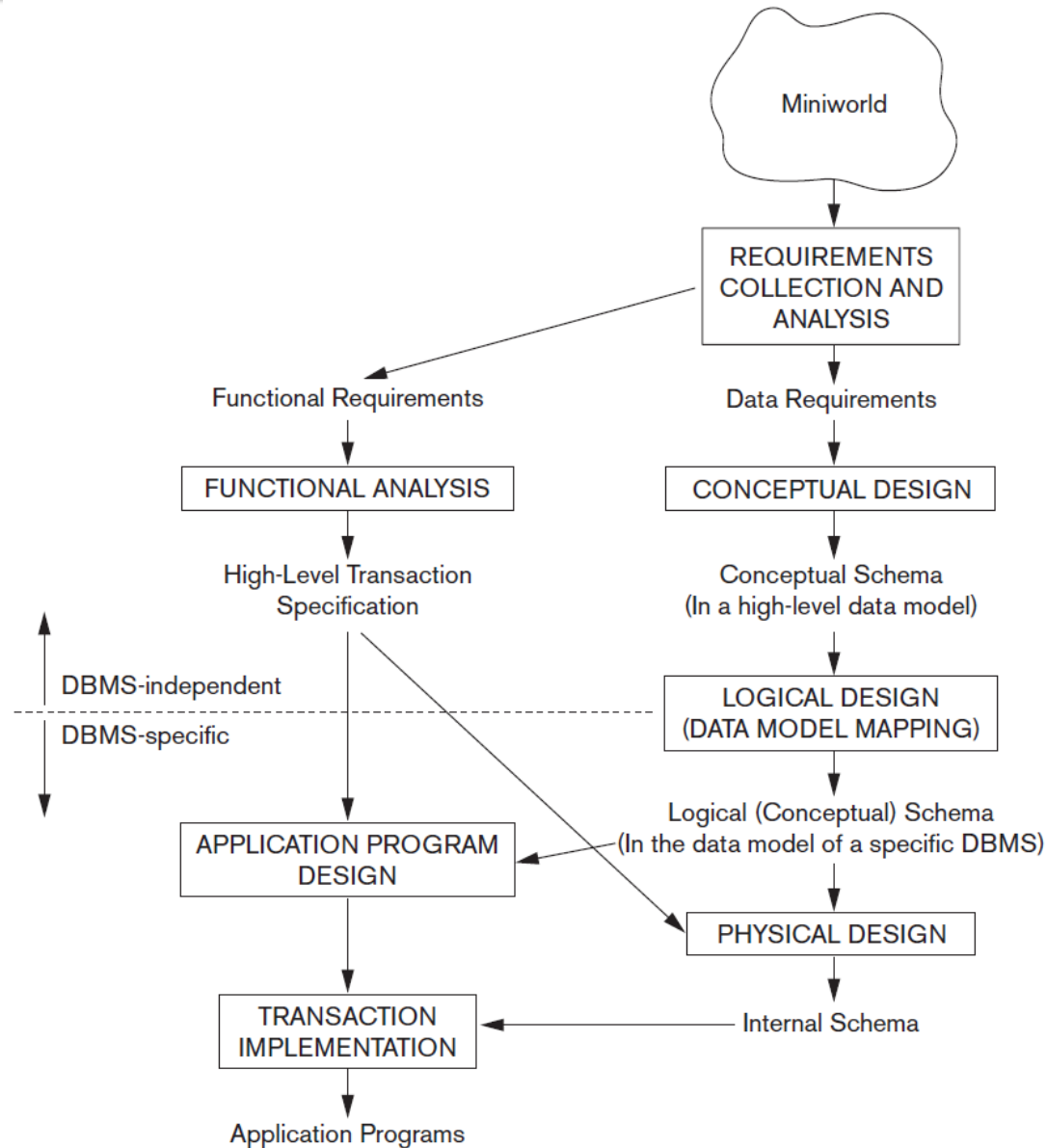
Objectives



- ◆ To learn conceptual modeling using the entity-relationship (ER) model
 - Database design process
 - Entity-relationship model
 - Conceptual data modeling

- ◆ Database Design Process
- ◆ Example: COMPANY Database
- ◆ Entities and Attributes
 - Types of Attributes
 - Entity Type
 - Key Attribute
- ◆ Initial Conceptual Design of COMPANY Database
- ◆ Relationship
 - Relationship Degree
 - Recursive Relationships

Database Design Process



◆ Requirements

- Employees, departments, and projects
- Company is organized into departments
- Department: has unique name, unique number, locations, manager and start date when manager began managing the department

◆ Requirements (cont'd)

- Department controls a number of projects
- Project: has unique name, unique number, and location

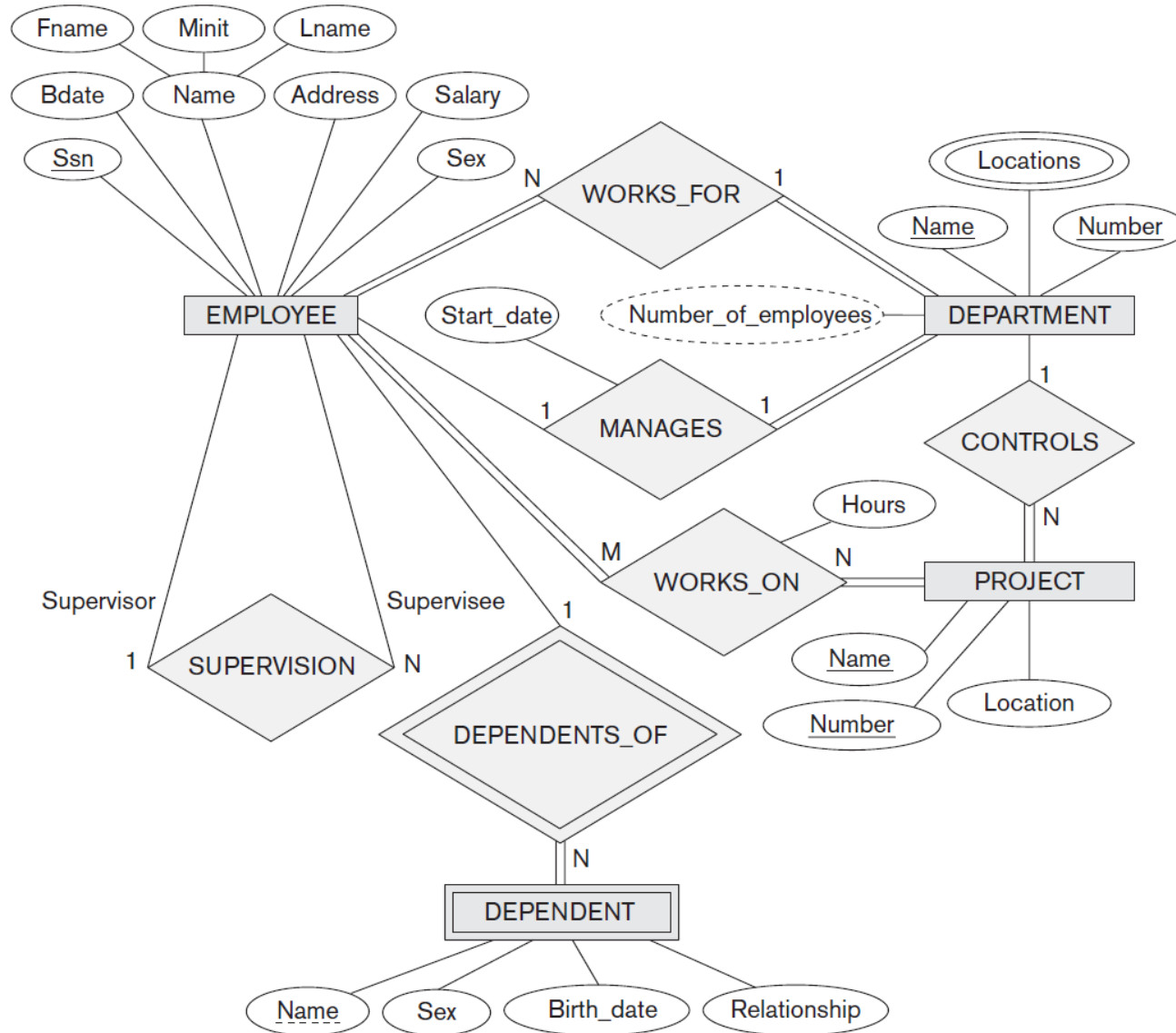
◆ Requirements (cont'd)

- Employee: store each employee's name, Social Security number, address, salary, sex (gender), and birth date
- Employee is assigned to one department
- Employee may work on several projects
- Keep track of number of hours per week that employee work on each project
- Keep track of the direct supervisor

◆ Requirements (cont'd)

- Keep track of the dependents of each employee
- Dependent: dependent's first name, sex, birth date, and relationship to the employee

COMPANY Database



Entities and Attributes

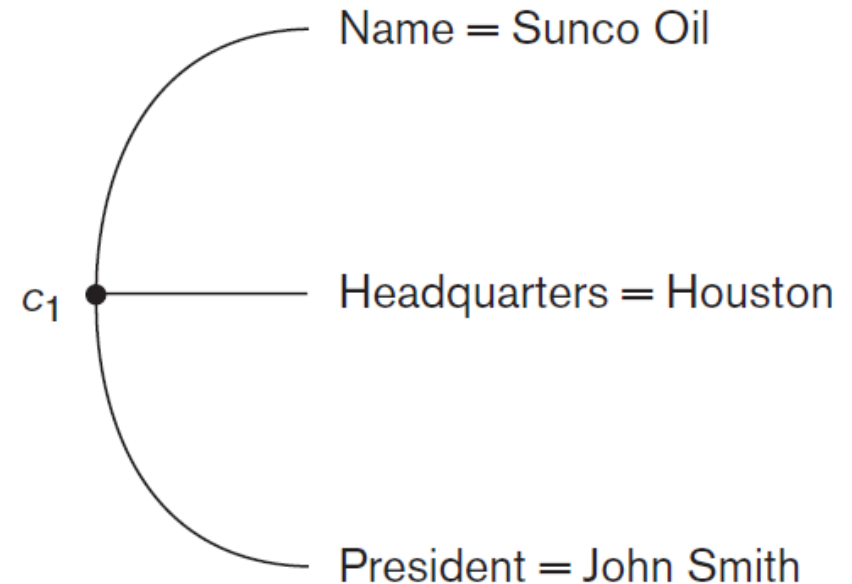
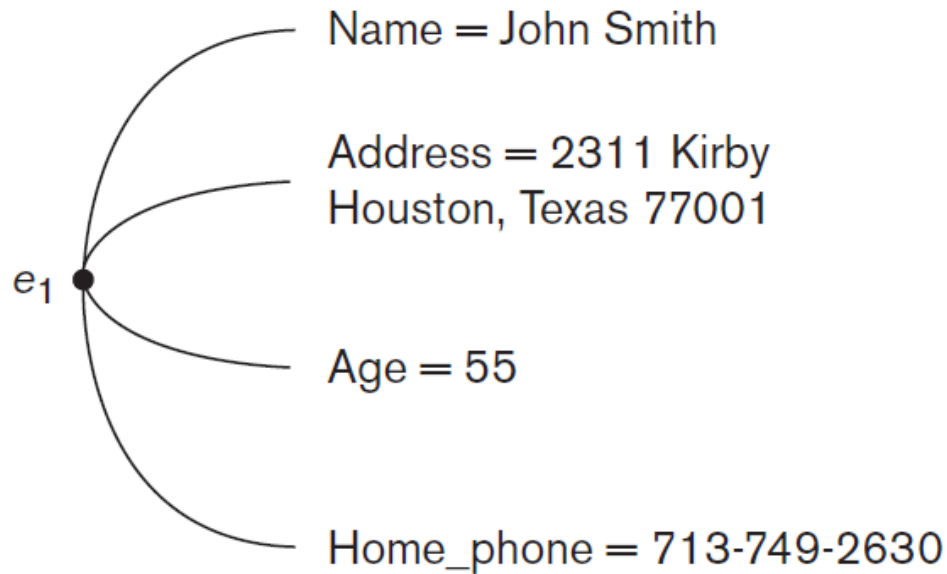
◆ Entity

- *Thing* in real world with independent existence
- Examples: person, car, house, student, employee, ...

◆ Attributes

- Particular *properties* that describe the entity
- Examples: name, age, address, job, phone number, ...

Entities and Attributes



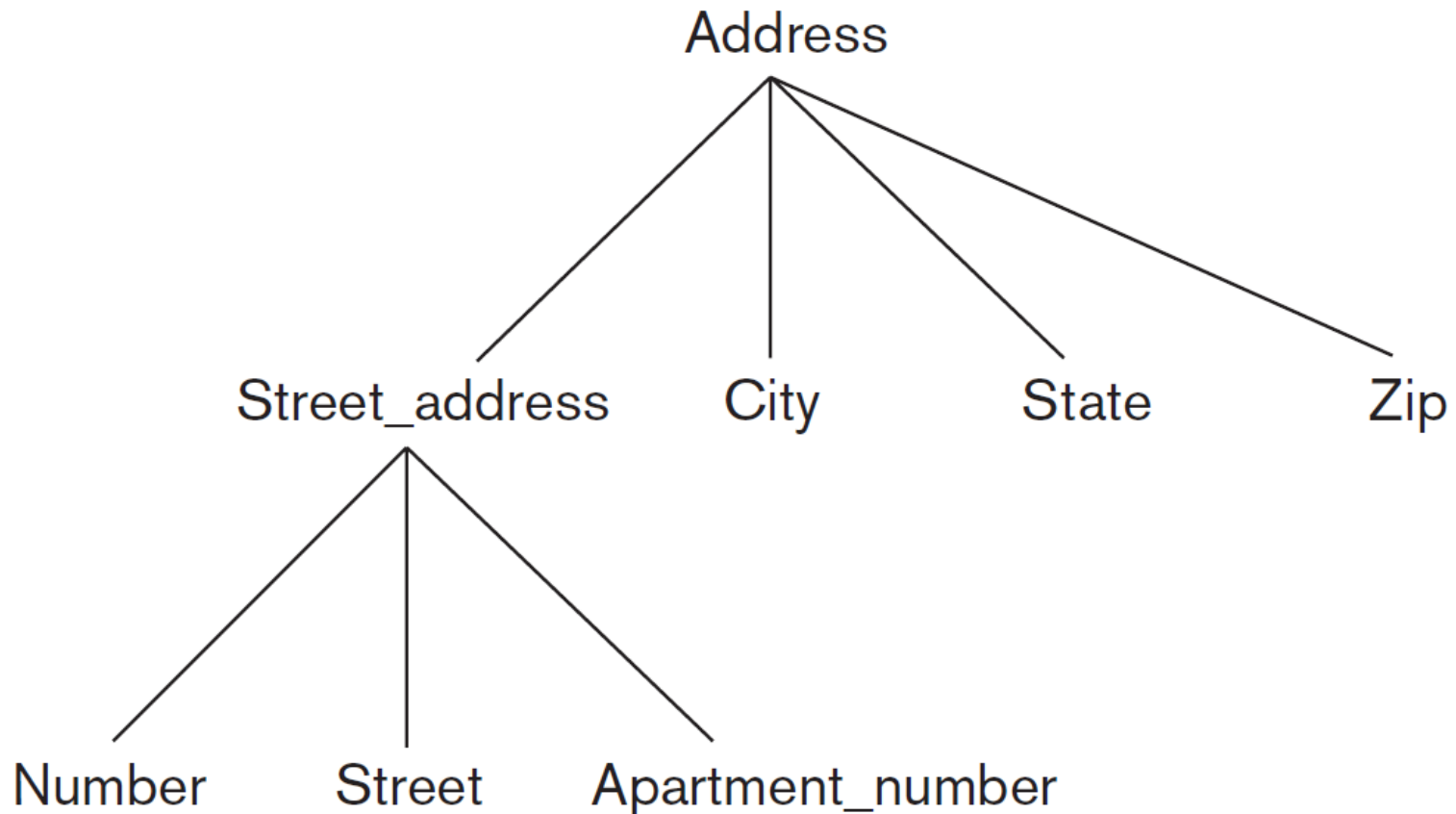
Types of Attributes

◆ Category 1

- Simple (atomic) attributes
 - Attributes that are *not divisible*
- Composite attributes
 - Can be *divided into smaller subparts*
 - Can form a *hierarchy*

Types of Attributes

- ◆ A hierarchy of composite attributes



Types of Attributes

◆ Category 2

● Single-valued attributes

- Have a ***single value*** for a particular entity
- Example: age

● Multivalued attributes (set attributes)

- Have a ***set of values*** for the same entity
- Example: phone numbers

Types of Attributes

◆ Category 3

● Stored attributes

- Actually *stored* in the database
- Example: birth date

● Derived attributes

- Can be *derived* from other attributes
- Not stored in the database
- Example: age (can be derived from birth date)

Entity Type

- ◆ Defines a collection of entities that have a *same set of attributes*
- ◆ Indicates the *schema* or *intention* for a set of entities
- ◆ Described by its *name* and *attributes*

Entity Type



Entity Type Name:

EMPLOYEE

COMPANY

Name, Age, Salary

Name, Headquarters, President

schema

Entity Set:
(Extension)

instance

e_1 ●

(John Smith, 55, 80k)

e_2 ●

(Fred Brown, 40, 30K)

e_3 ●

(Judy Clark, 25, 20K)

⋮

c_1 ●

(Sunco Oil, Houston, John Smith)

c_2 ●

(Fast Computer, Dallas, Bob King)

⋮

◆ Domains

- Set of values that may be assigned to the attribute for each individual entity
- Example
 - Domain for AGE attribute of EMPLOYEE entity: 20 ~ 70

Key Attribute

- ◆ Attributes whose values are *distinct* for each individual entity
- ◆ Values of key attributes can be used to *identify* each entity *uniquely*
- ◆ Example: student number of STUDENT entity
(ex.)

Key Attribute

- ◆ Multiple attributes can form a key
- ◆ A entity type can have *more than two keys*
- ◆ Preceding uniqueness property of a key must hold for *every entity set* of the entity type
 - Not for a particular entity set

Key Attribute



CAR

Registration (Number, State), Vehicle_id, Make, Model, Year, {Color}

CAR₁

((ABC 123, TEXAS), TK629, Ford Mustang, convertible, 2004 {red, black})

CAR₂

((ABC 123, NEW YORK), WP9872, Nissan Maxima, 4-door, 2005, {blue})

CAR₃

((VSY 720, TEXAS), TD729, Chrysler LeBaron, 4-door, 2002, {white, blue})

•
•
•

Initial Conceptual Design of the COMPANY Database



DEPARTMENT

Name, Number, {Locations}, Manager, ManagerStartDate

PROJECT

Name, Number, Location, ControllingDepartment

EMPLOYEE

Name(FName, MInit, LName), SSN, Sex, Address, Salary, BirthDate, Department, Supervisor, {WorksOn (Project, Hours)}

DEPENDENT

Employee, DependentName, Sex, BirthDate, Relationship

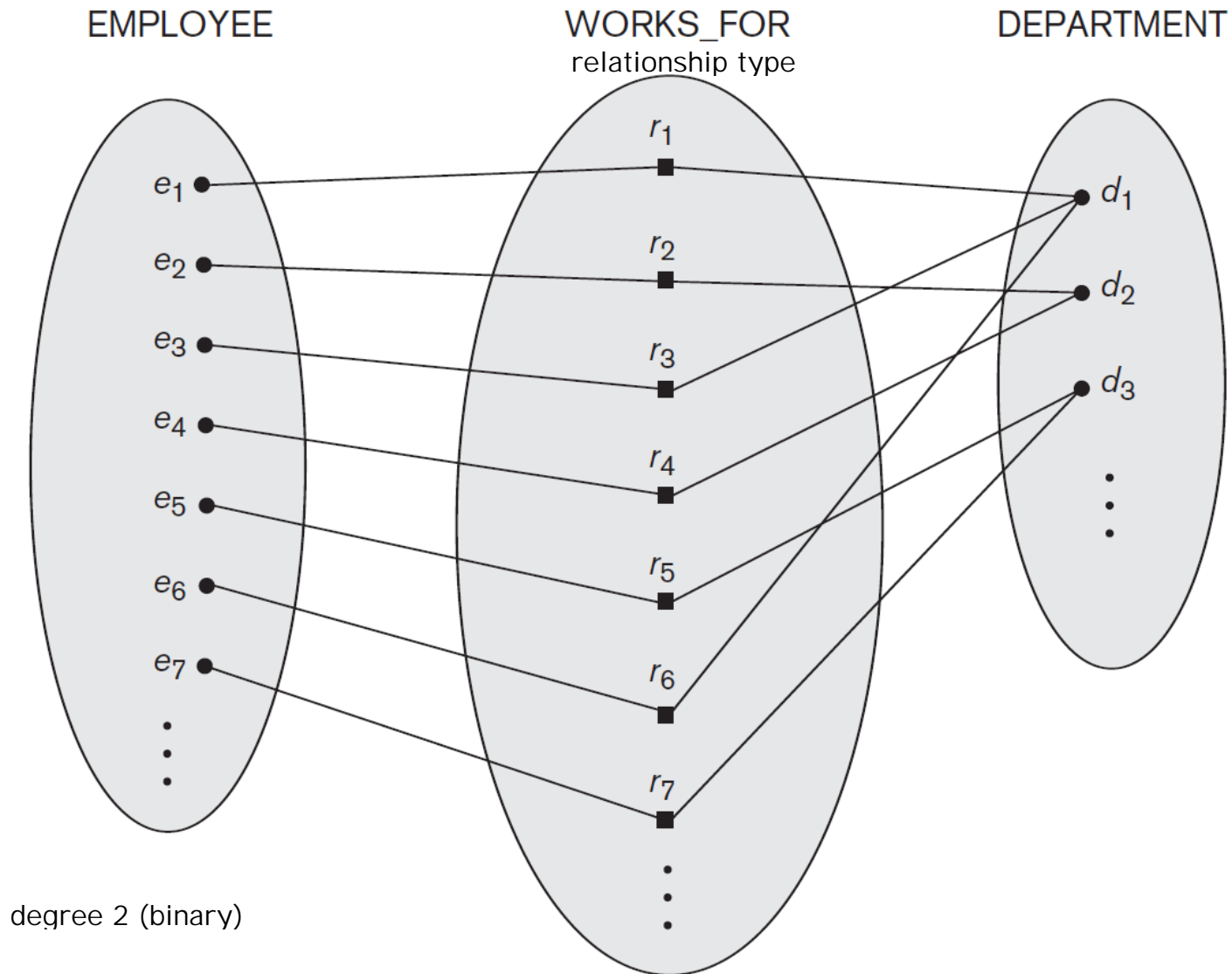
{ }: multivalued attribute, (): composite attribute

Concepts of Relationship

- ◆ Relationship instances
 - *Association* among entities from entity set

- ◆ Relationship type
 - Indicates *set of relationship instances* from entity types
 - Represented with *participating* entity types

WORKS_FOR Relationship



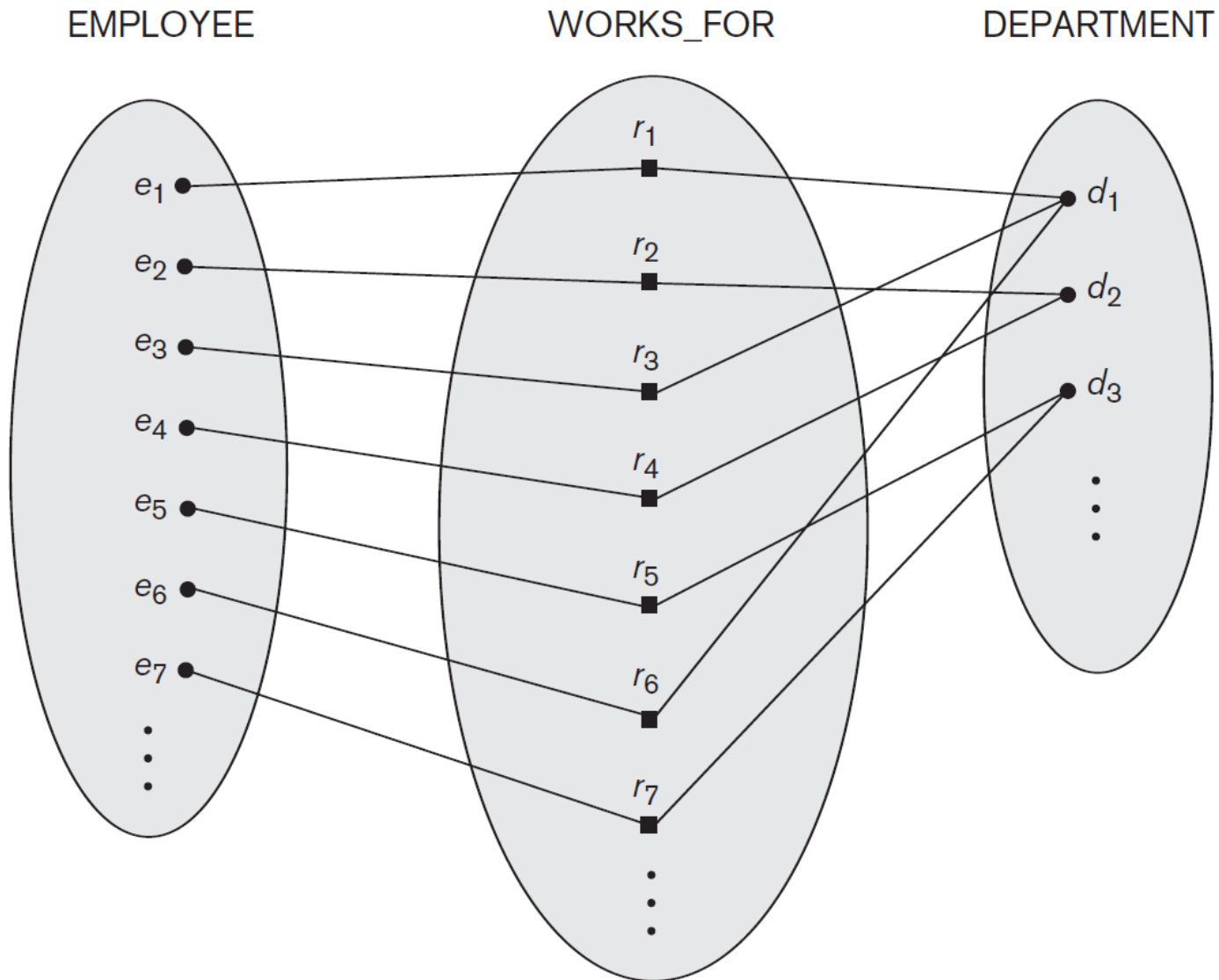
Relationship Degree

- ◆ *Degree* of a relationship type
 - Number of participating entity types
 - *Binary, ternary, ...*

Binary Relationship

- ◆ A relationship type of *degree two*
- ◆ Most commonly used
- ◆ Example
 - WORKS_FOR relation type has EMPLOYEE and DEPARTMENT as participating entity types

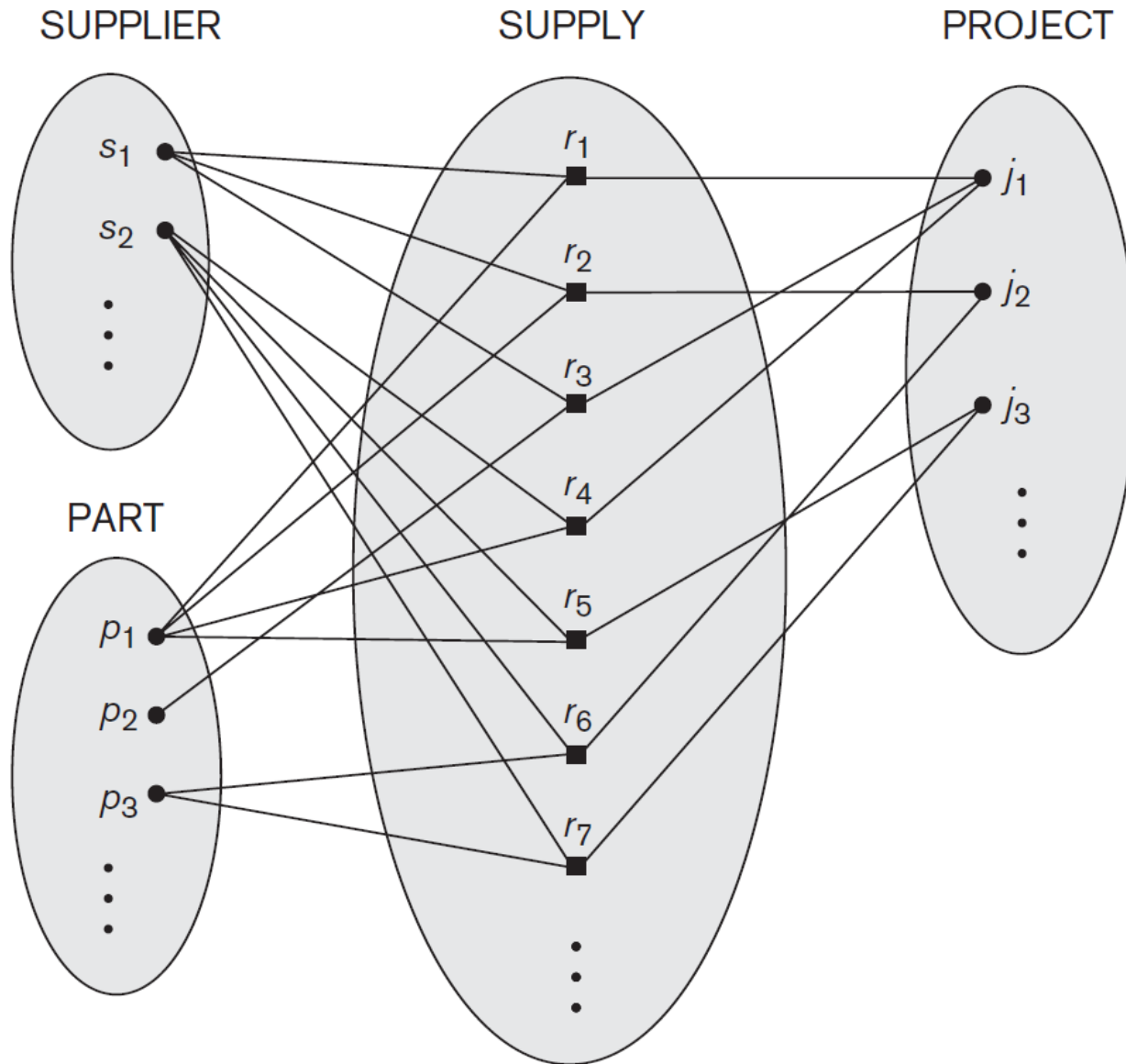
WORKS_FOR Relationship



Ternary Relationship

- ◆ A relationship type of *degree three*
- ◆ Example
 - SUPPLY relation type has SUPPLIER, PART, and PROJECT as participating entity types

SUPPLY Relationship



Recursive Relationships



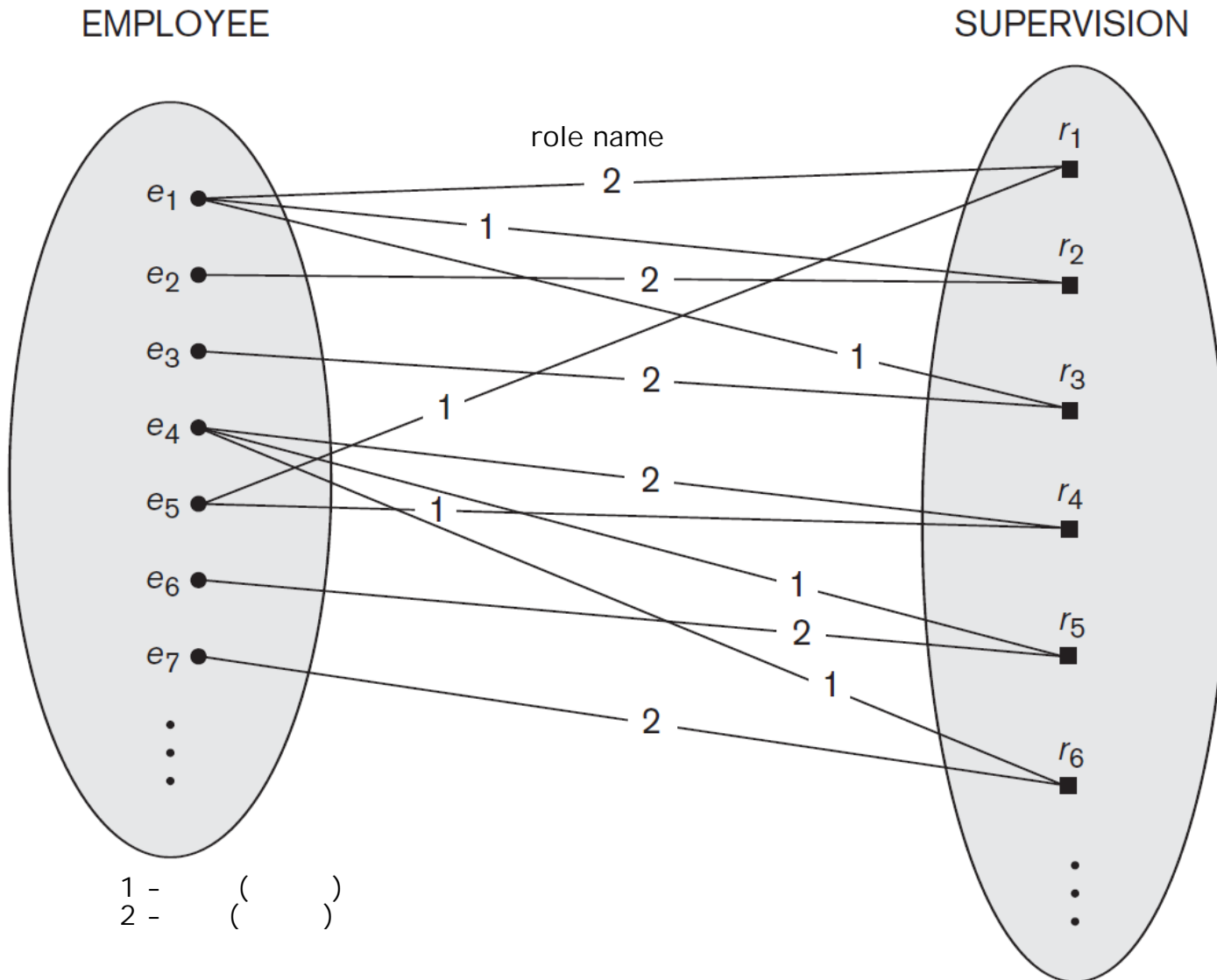
◆ Role name

- Signifies role that a participating entity plays in each relationship instance
- Example: EMPLOYEE and DEPARTMENT in WORKS_FOR relation type

Recursive Relationships

- ◆ Recursive relationship type
 - Same entity type participates more than once in a relationship type in different roles
 - Must specify role name

Recursive Relationships



- ◆ Database design process
- ◆ Basic ER model concepts of entities and their attributes
 - Different types of attributes
 - Structural constraints on relationships

References



1. Batini, Carlo., S. Ceri, S. B. Navathe, and Carol Batini, *Conceptual Database Design: an Entity/Relationship Approach*, Addison-Wesley, Reading MA, 1991.
2. Chen, P. P., "The entity-relationship model: toward a unified view of data," *ACM Trans. on Database Systems* **1**:1, pp. 9-36, 1976.
3. Thalheim, B., *Fundamentals of Entity-Relationship Modeling*, Springer-Verlag, Berlin, 2000.

Have a nice day!