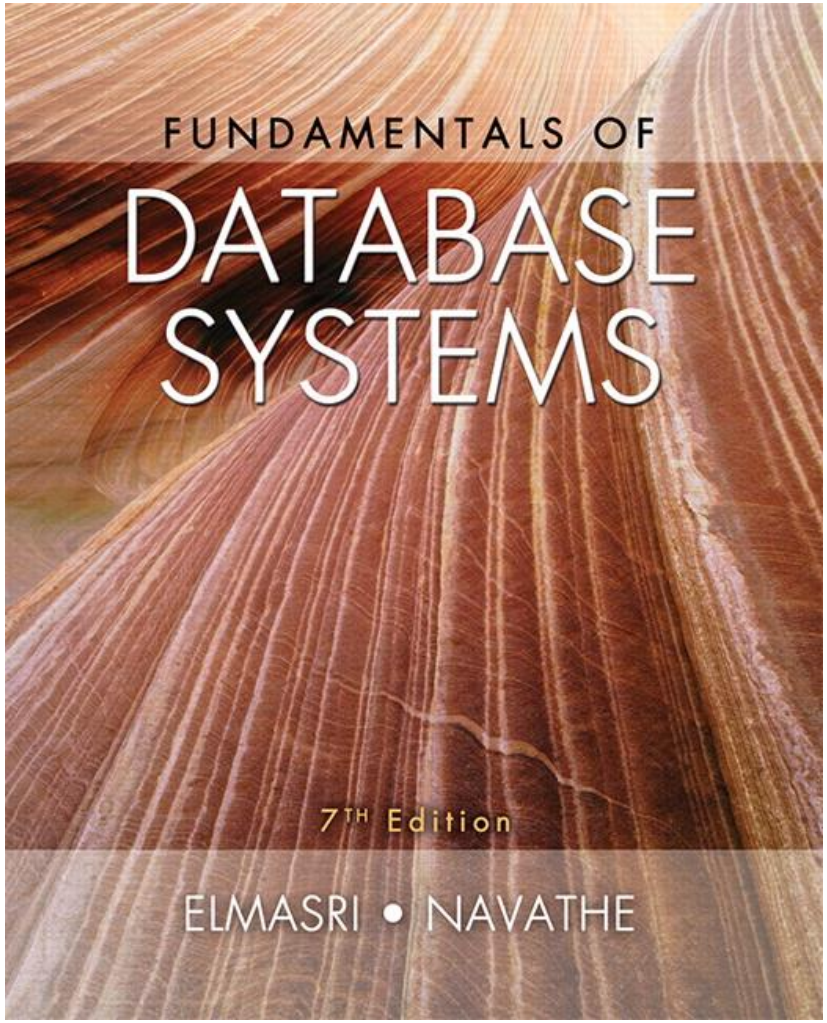


# CHAPTER 4

## Enhanced Entity-Relationship (EER) Modeling

## and

## Mapping EER Schemas to Relational Schemas (Chapter 9: 9.2.1 and 9.2.2)

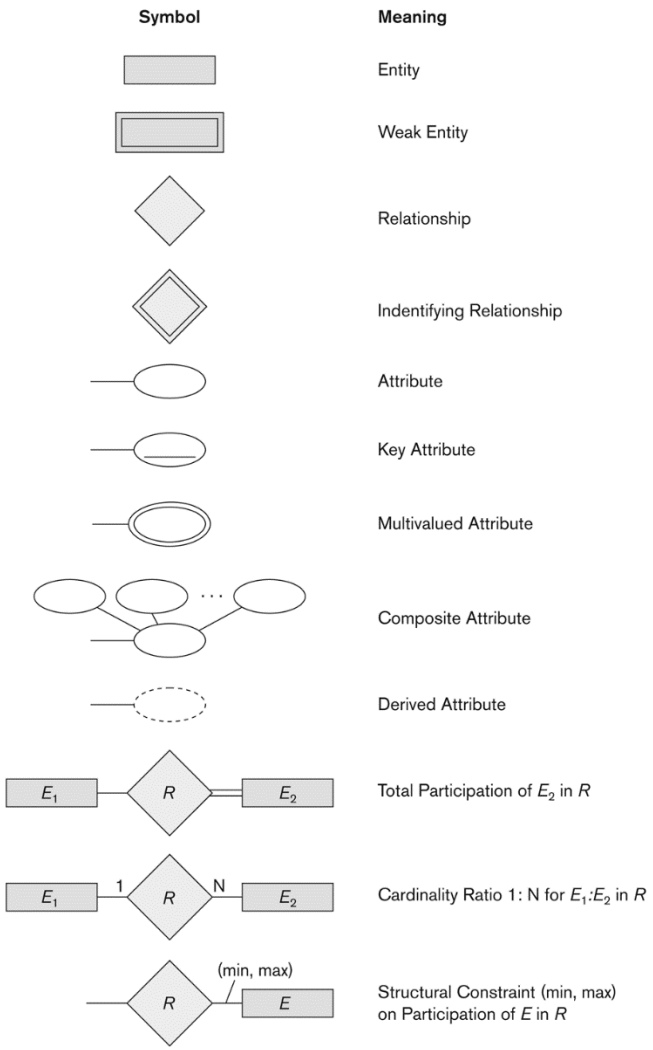
# Chapter 4 Outline

- EER stands for Enhanced ER or Extended ER
- EER Model Concepts
  - EER Includes all modeling concepts of basic ER
  - Additional concepts:
    - subclasses/superclasses
    - specialization/generalization
    - attribute and relationship inheritance
  - Constraints on Specialization/Generalization

- IGNORE (LEAVE OUT): Sections 4.4, 4.5, 4.6, 4.7
  - 4.4 – Union Types
  - 4.5 – Design Choices
  - 4.6 – Other notations including UML diagrams
  - 4.7 – Data Abstraction, Knowledge Representation and Ontologies

# NOTATION for ER diagrams

**Figure 3.14**
Summary of the notation for ER diagrams.



# ER DIAGRAM – for Company Database

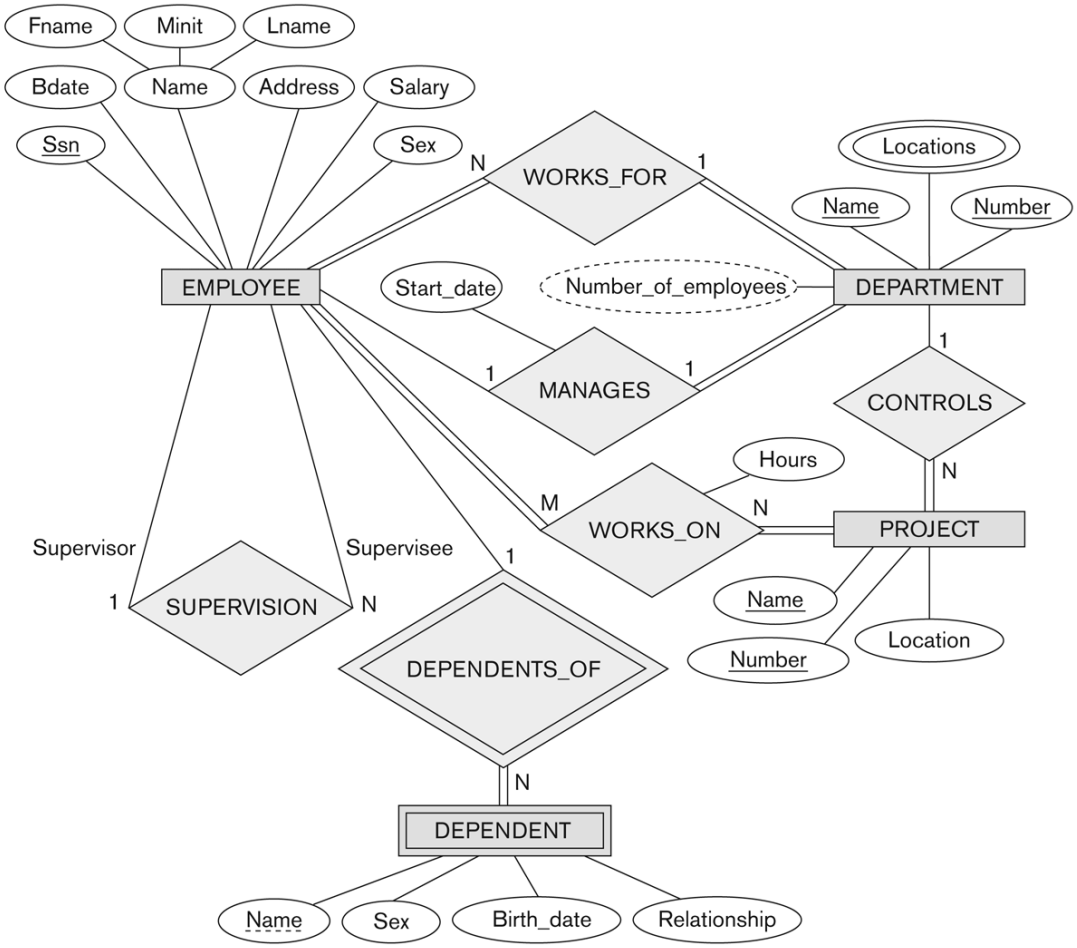


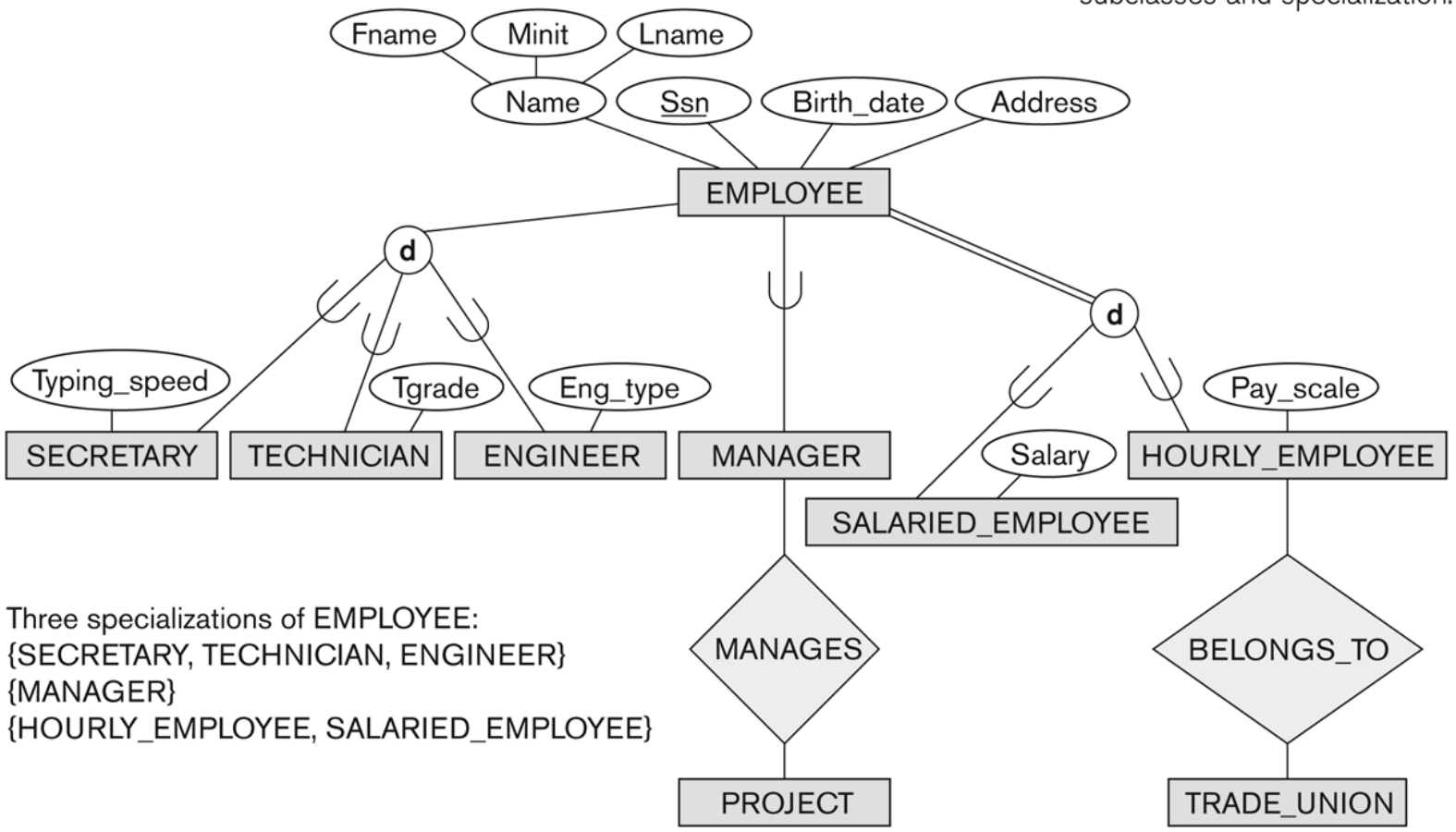

**Figure 3.2**
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

# Subclasses and Superclasses

- An entity type may have additional meaningful subgroupings of its entities
  - Example: EMPLOYEE may be further grouped into:
    - SECRETARY, ENGINEER, TECHNICIAN, …
      - Based on the EMPLOYEE's Job
    - MANAGER
      - EMPLOYEEs who are managers (the role they play)
    - SALARIED_EMPLOYEE, HOURLY_EMPLOYEE
      - Based on the EMPLOYEE's method of pay
- EER diagrams extend ER diagrams to represent these additional subgroupings, called *subclasses* or *subtypes*

# Subclasses and Superclasses

**Figure 4.1**
EER diagram notation to represent subclasses and specialization.



# Subclasses and Superclasses

- Each of these subgroupings is a subset of EMPLOYEE entities
- Each is called a subclass of EMPLOYEE
- EMPLOYEE is the superclass for each of these subclasses
- These are called superclass/subclass relationships:
  - EMPLOYEE/SECRETARY
  - EMPLOYEE/TECHNICIAN
  - EMPLOYEE/MANAGER
  - ….could be additional subclasses

# Subclasses and Superclasses –contd.

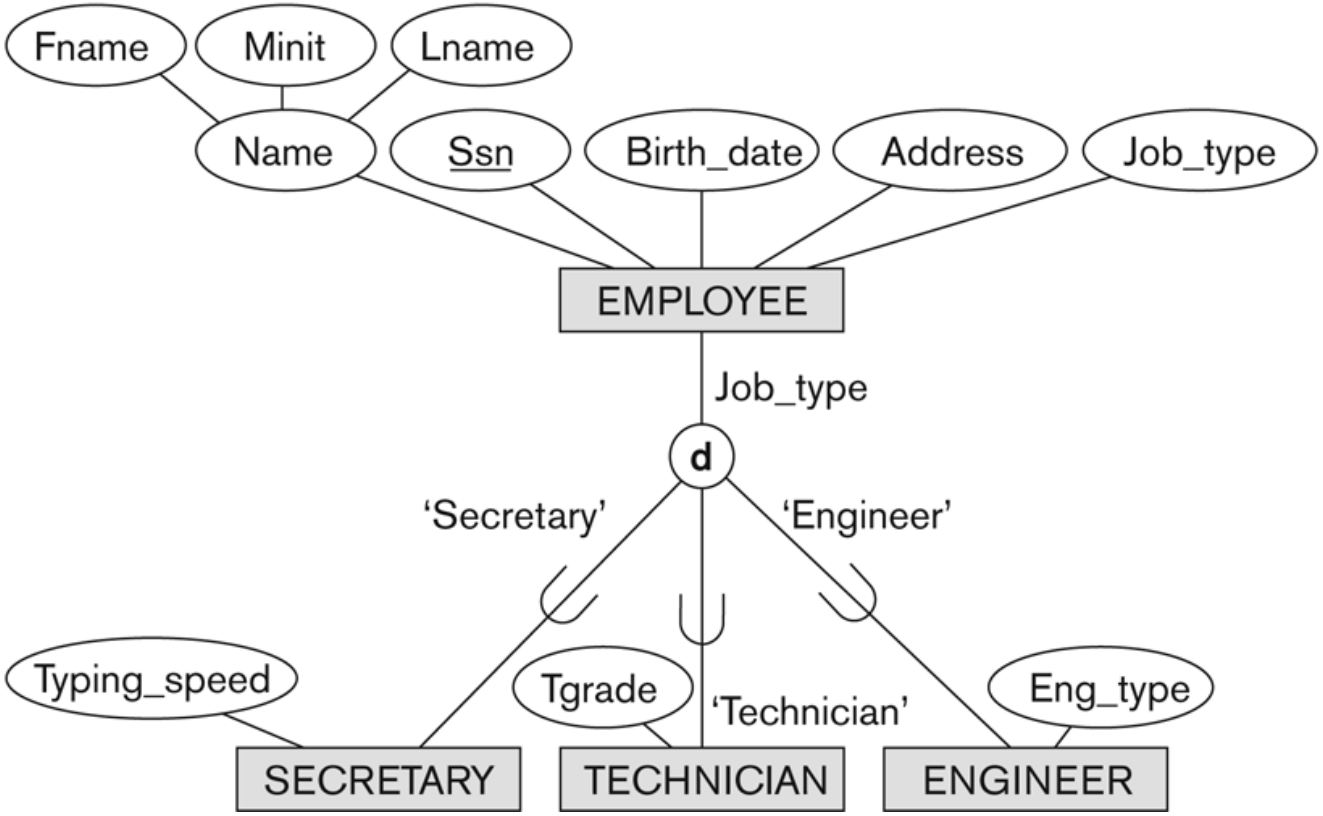
- These are also called IS-A relationships
  - SECRETARY IS-A EMPLOYEE, TECHNICIAN IS-A EMPLOYEE, ….
- Note: An entity that is member of a subclass represents the same real-world entity as some member of the superclass:
  - The subclass member is the same entity in a *distinct specific role*
  - An entity cannot exist in the database merely by being a member of a subclass; it must also be a member of the superclass
  - A member of the superclass can be optionally included as a member of any number of its subclasses

# Subclasses and Superclasses –contd.

- Examples:
  - A salaried employee who is also an engineer belongs to the two subclasses:
    - ENGINEER, and
    - SALARIED_EMPLOYEE
  - A salaried employee who is also an engineering manager belongs to the three subclasses:
    - MANAGER,
    - ENGINEER, and
    - SALARIED_EMPLOYEE
- It is not necessary that every entity in a superclass be a member of some subclass

# Representing **Specialization** in EER Diagrams

**Figure 4.4**
EER diagram notation for an attribute-defined specialization on Job_type.



# Attribute and Relationship Inheritance in Superclass / Subclass Relationships

- An entity that is member of a subclass *inherits*
  - All attributes of the entity as a member of the superclass
  - All relationships of the entity as a member of the superclass
- Example:
  - In the previous slide, SECRETARY (as well as TECHNICIAN and ENGINEER) inherit the attributes Name, SSN, …, from EMPLOYEE
  - Every SECRETARY entity will have values for the inherited attributes

# Specialization (1)

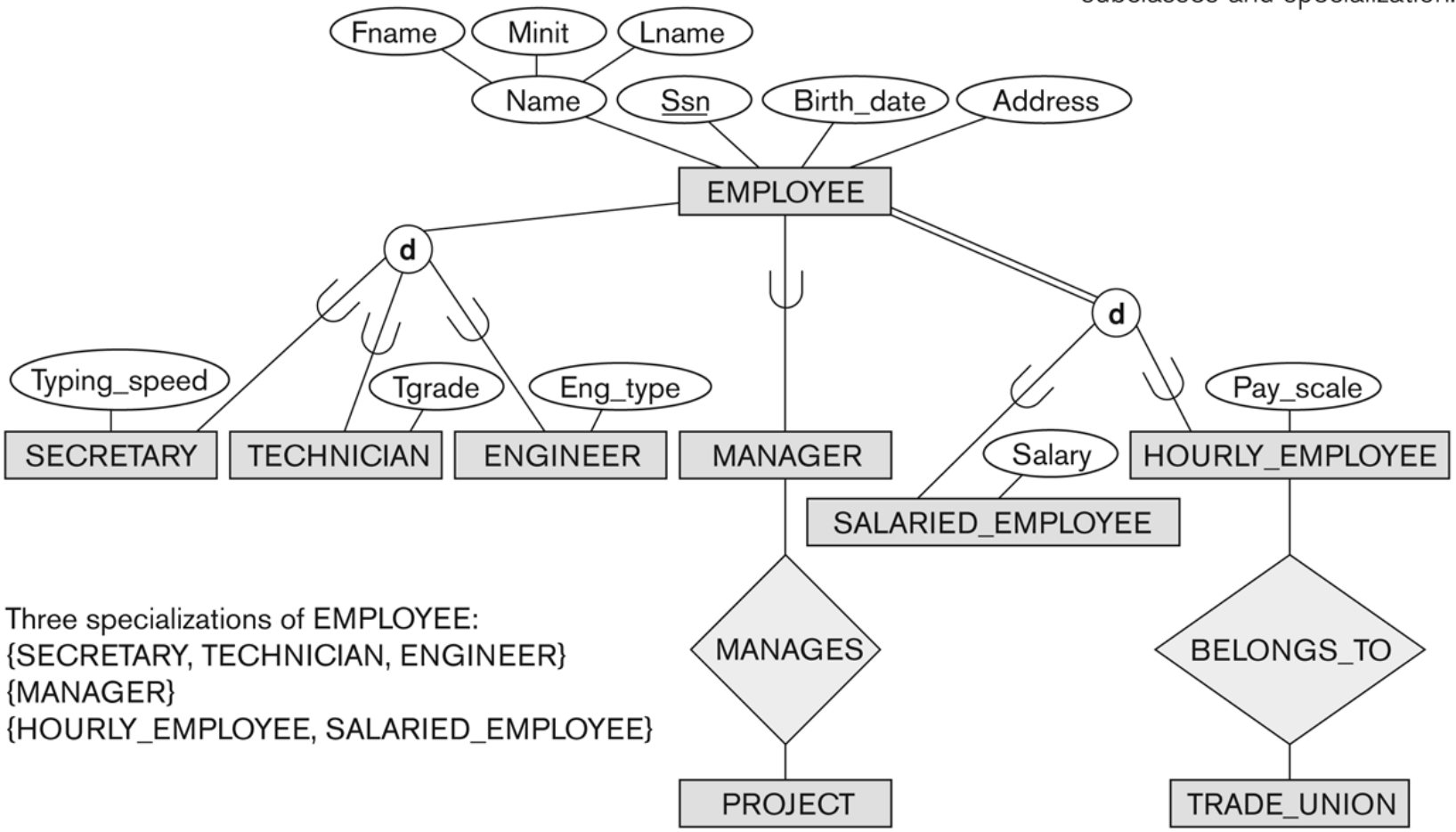
- Specialization is the process of defining a set of subclasses of a superclass
- The set of subclasses is based upon some distinguishing characteristics of the entities in the superclass
  - Example: {SECRETARY, ENGINEER, TECHNICIAN} is a specialization of EMPLOYEE based upon *job type,* a discriminating attribute.
  - Example: MANAGER *is a specialization of EMPLOYEE* based on *the role the employee plays*
    - May have several specializations of the same superclass

# Specialization (2)

- Example: Another specialization of EMPLOYEE based on *method of pay* is {SALARIED_EMPLOYEE, HOURLY_EMPLOYEE}.
  - Superclass/subclass relationships and specialization can be diagrammatically represented in EER diagrams as a hierarchy
  - Attributes of a subclass are called *specific* or *local* attributes.
    - For example, the attribute TypingSpeed of SECRETARY
  - The subclass can also participate in specific relationship types.
    - For example, a relationship BELONGS_TO of HOURLY_EMPLOYEE

# Specialization (3)

**Figure 4.1**
EER diagram notation to represent subclasses and specialization.




Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

# Generalization

- Generalization is the reverse of the specialization process
- Several classes with common features are generalized into a superclass;
  - original classes become its subclasses
- Example: CAR, TRUCK generalized into VEHICLE;
  - both CAR, TRUCK become subclasses of the superclass VEHICLE.
  - We can view {CAR, TRUCK} as a specialization of VEHICLE
  - Alternatively, we can view VEHICLE as a generalization of CAR and TRUCK

# Generalization (2)

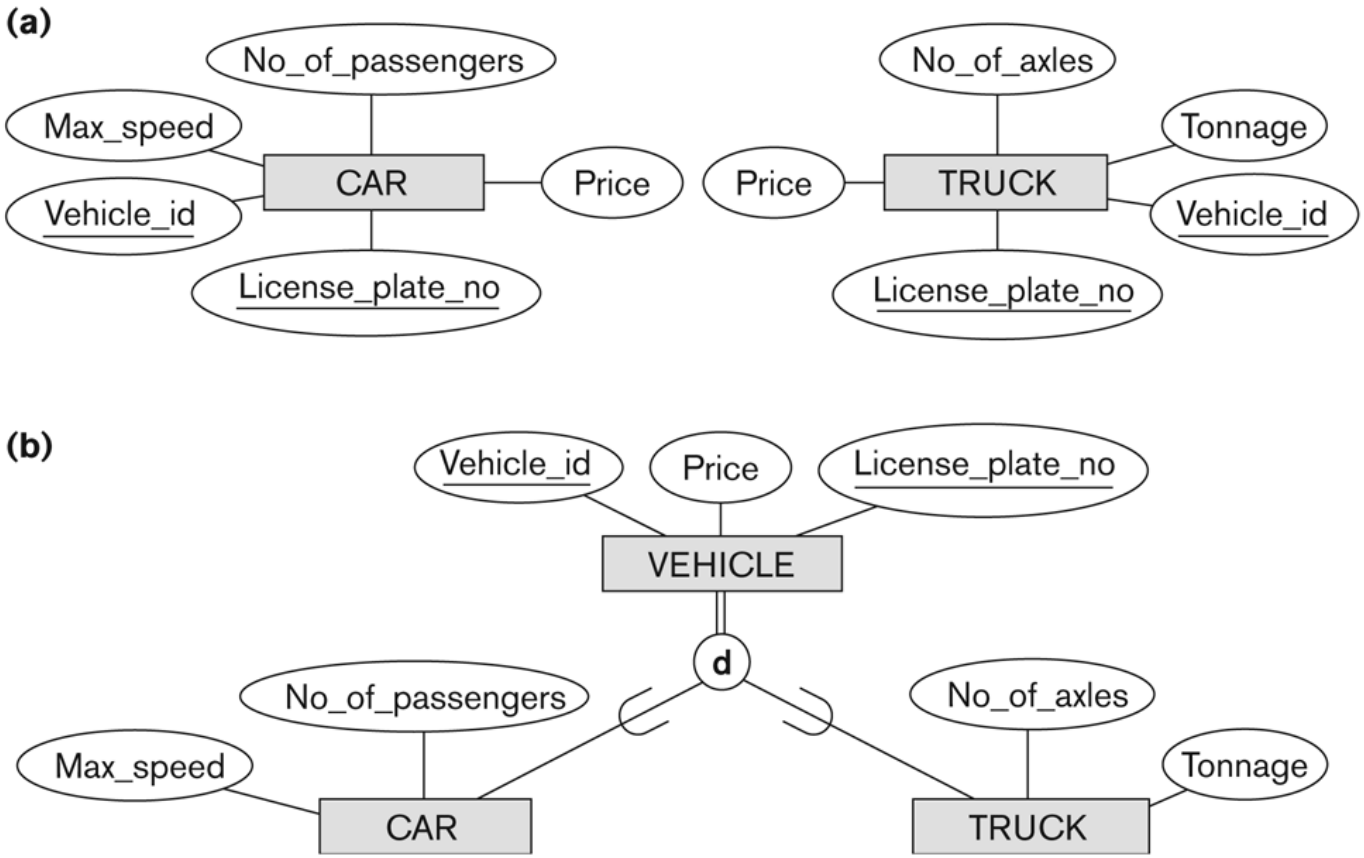


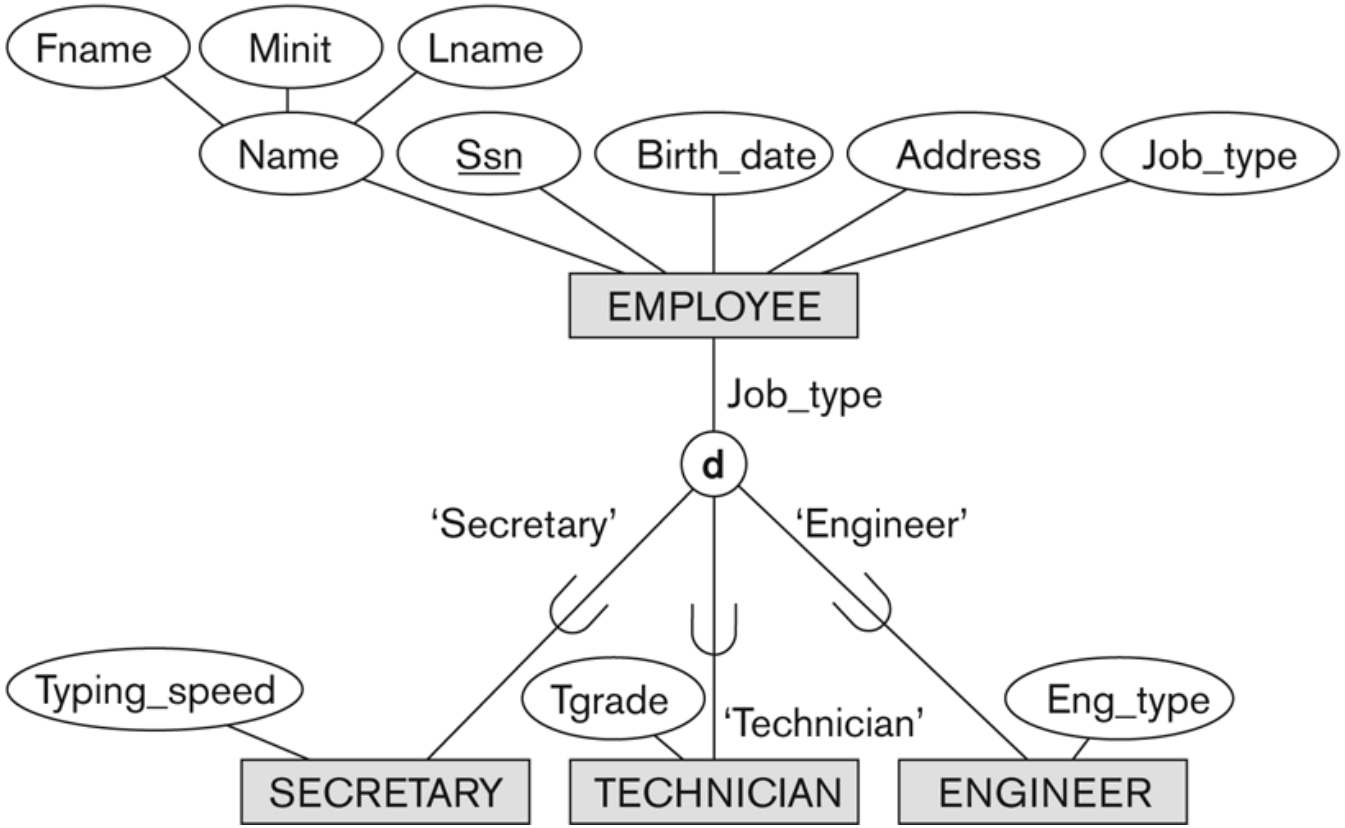

**Figure 4.3**
Generalization. (a) Two entity types, CAR and TRUCK.
(b) Generalizing CAR and TRUCK into the superclass VEHICLE.

# Types of Specialization

- Predicate-defined ( or condition-defined) : based on some predicate. E.g., based on value of an attribute, say, Age.
- Attribute-defined: shows the name of the attribute next to the line drawn from the superclass toward the subclasses (see Fig. 4.1)
- User-defined: membership is defined by the user on an entity by entity basis

# Displaying an attribute-defined specialization in EER diagrams

**Figure 4.4**
EER diagram notation for an attribute-defined specialization on Job_type.



# Constraints on Specialization and Generalization (3)

- Two basic constraints can apply to a specialization/generalization:
  - Disjointness Constraint:
  - Completeness Constraint:

# Constraints on Specialization and Generalization (4)

- Disjointness Constraint:
  - Specifies that the subclasses of the specialization must be *disjoint*:
    - an entity can be a member of at most one of the subclasses of the specialization
  - Specified by ***d*** in EER diagram
  - If not disjoint, specialization is *overlapping*:
    - that is the same entity may be a member of more than one subclass of the specialization
  - Specified by ***o*** in EER diagram

# Constraints on Specialization and Generalization (5)

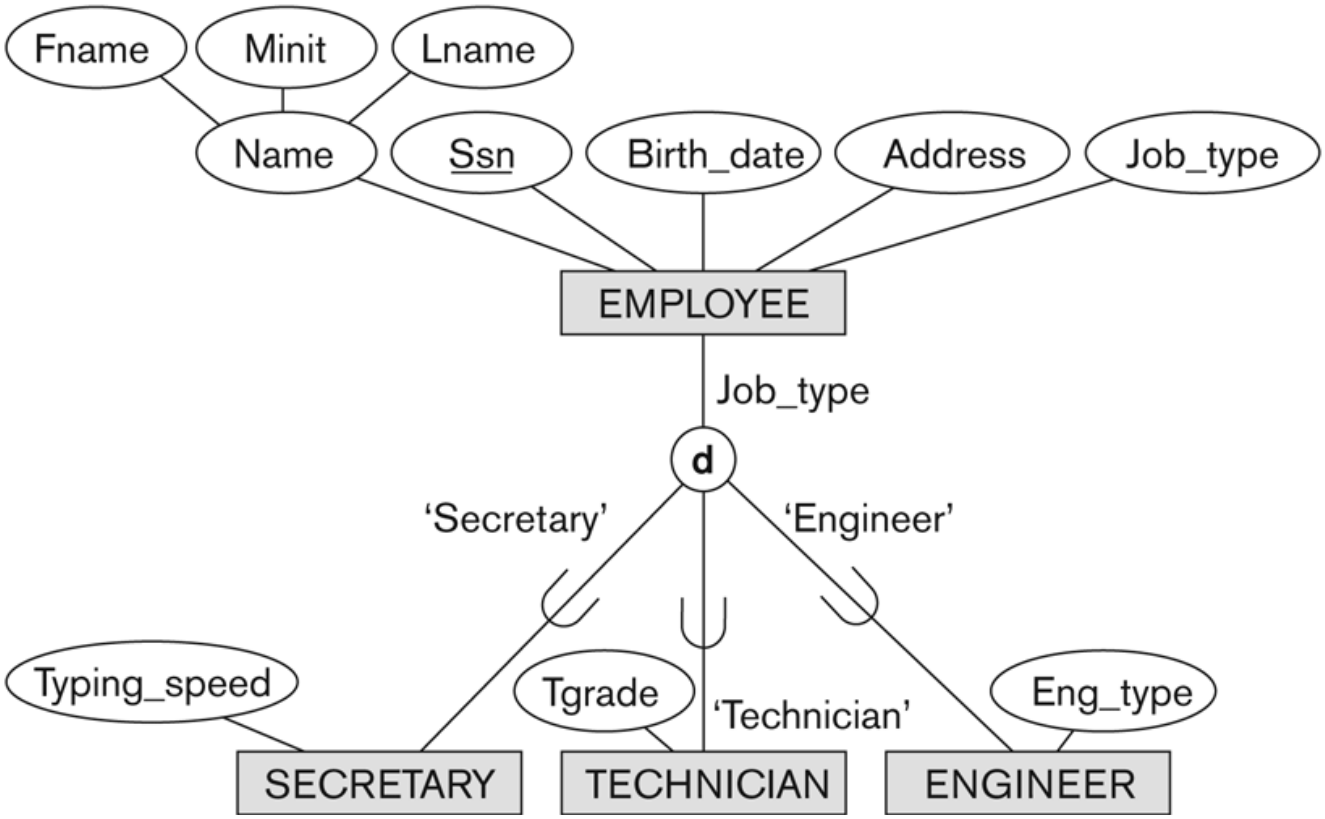
- Completeness (Exhaustiveness) Constraint:
  - *Total* specifies that every entity in the superclass must be a member of some subclass in the specialization/generalization
  - Shown in EER diagrams by a ***double line***
  - *Partial* allows an entity not to belong to any of the subclasses
  - Shown in EER diagrams by a **single line**

# Constraints on Specialization and Generalization (6)

- Hence, we have four types of specialization/generalization:
  - Disjoint, total
  - Disjoint, partial
  - Overlapping, total
  - Overlapping, partial
- Note: Generalization usually is total because the superclass is derived from the subclasses.

# Example of disjoint partial Specialization

**Figure 4.4**
EER diagram notation for an attribute-defined specialization on Job_type.



# Example of overlapping total Specialization

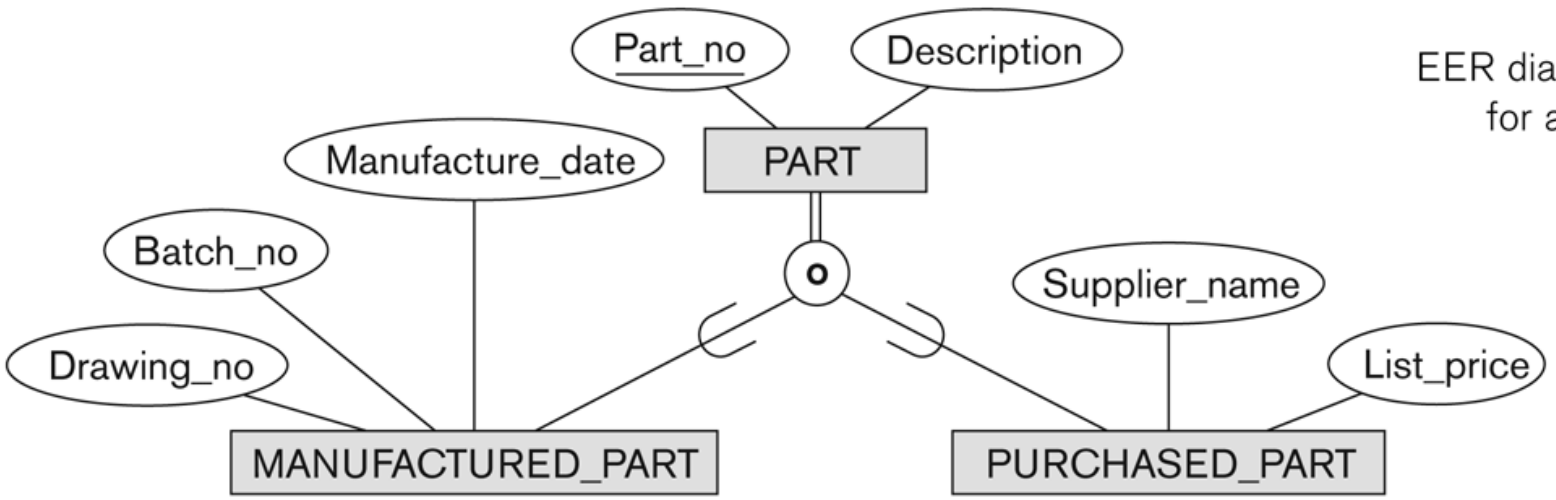



**Figure 4.5**
EER diagram notation for an overlapping (nondisjoint) specialization.

# Specialization/Generalization Hierarchies, Lattices & Shared Subclasses (1)

- A subclass may itself have further subclasses specified on it
  - forms a hierarchy or a lattice
- ***Hierarchy*** has a constraint that every subclass has only one superclass (called ***single inheritance***); this is basically a ***tree structure***
- In a ***lattice***, a subclass can be subclass of more than one superclass (called ***multiple inheritance***)

# Shared Subclass “Engineering_Manager”resulting in a Lattice

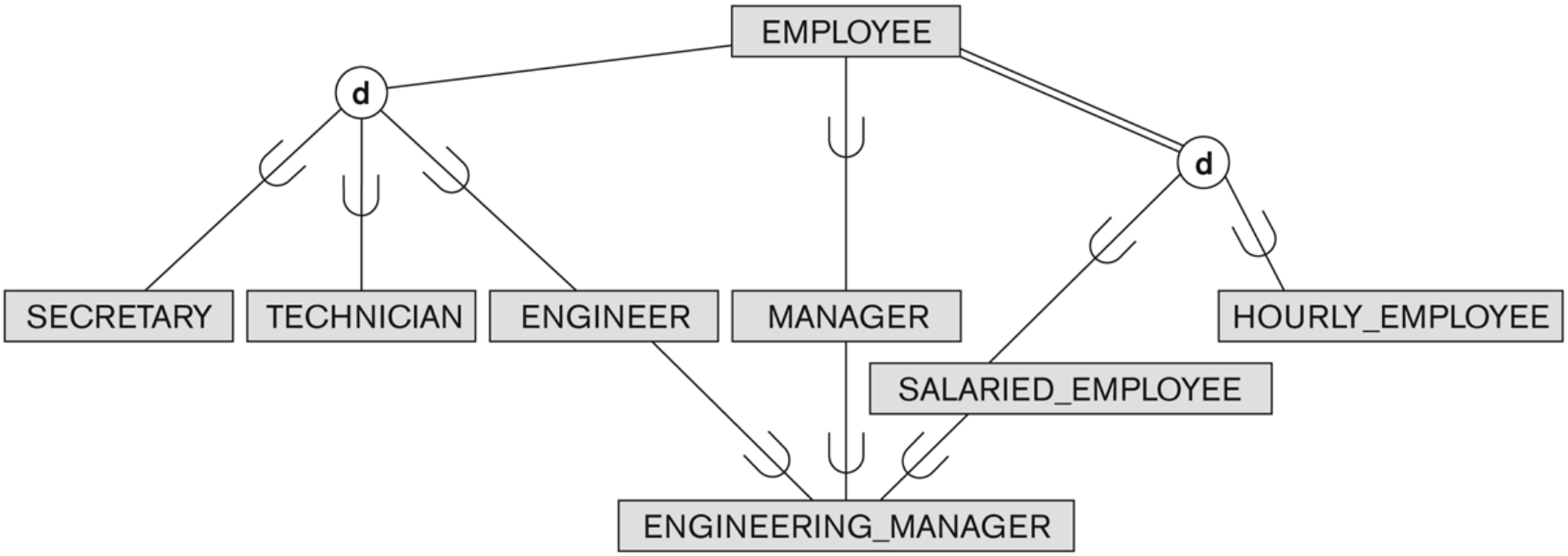


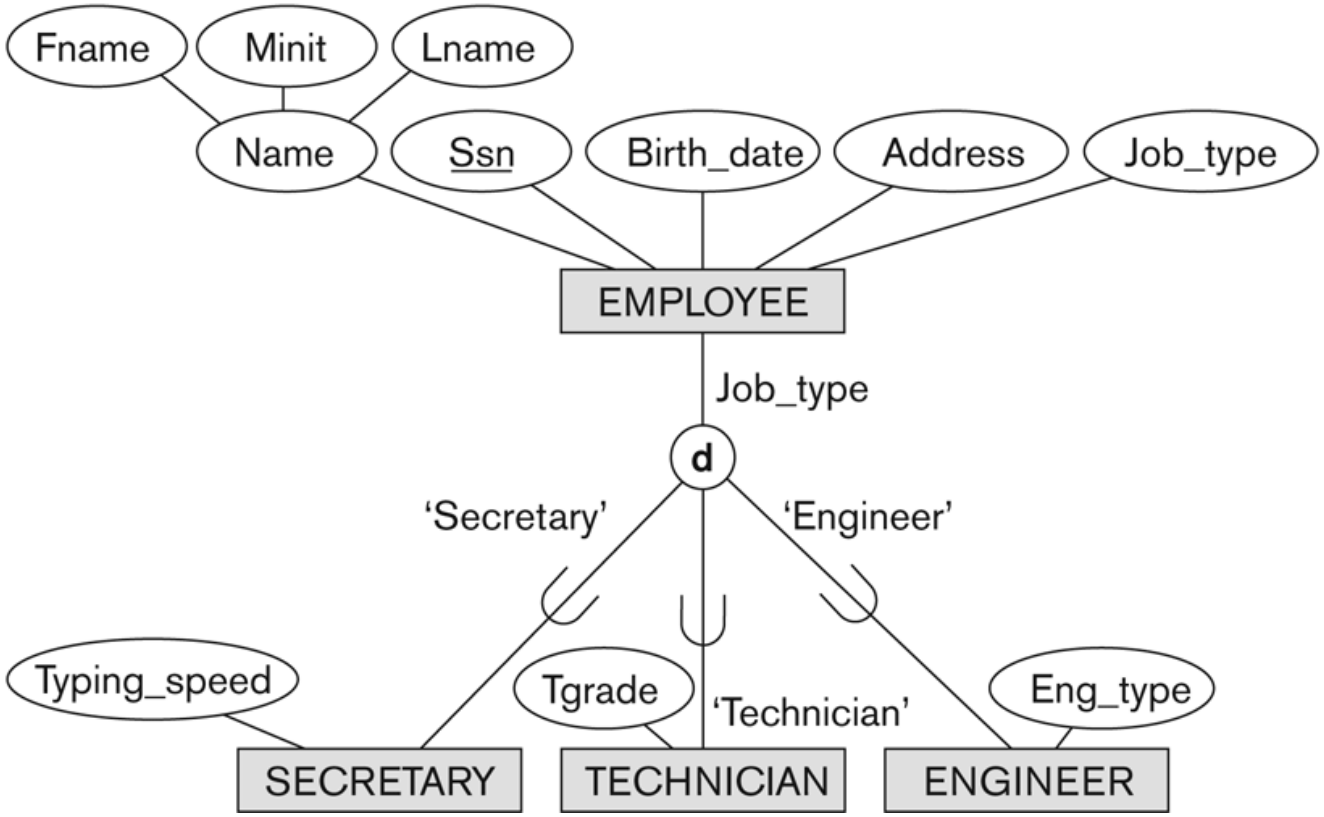

**Figure 4.6**
A specialization lattice with shared subclass ENGINEERING_MANAGER.

## CHAPTER 9 Mapping Algorithm:

- **Step8: Options for Mapping Specialization or Generalization.**
  - Convert each specialization with m subclasses {S1, S2,….,Sm} and generalized superclass C, where the attributes of C are {k,a1,…an} and k is the (primary) key, into relational schemas using one of the four following options:
    - Option 8A: Multiple relations-Superclass and subclasses
    - Option 8B: Multiple relations-Subclass relations only
    - Option 8C: Single relation with one type attribute
    - Option 8D: Single relation with multiple type attributes

# Example EER Schema to be mapped

**Figure 4.4**
EER diagram notation for an attribute-defined specialization on Job_type.



# Possible Mappings (1)

Possible to map this into anywhere from 1 to 4 relations

- One relation option:

Combine all attributes into a single relation and introduce one or more discriminating attributes to identify the subclass as needed:

Employee(Ssn, Fname, Lname, Minit, Birth_date,Address, Job_type, Typing_speed, Tech_grade, Eng_type)

Note: Some specialized class attributes will be null based on which subclass an employee belongs to.

- Four relations option: most straight-forward.

Employee(Ssn, Fname, Lname, Minit, Birth_date,Address, Job_type )

Secretary (Ssn, Typing_speed)

Technician (Ssn, Tech_grade)

Engineer (Ssn, Eng_type)

# Possible Mappings (2)

- Three relations option:

If the specialization was TOTAL, i.e., if each employee was either a secretary or a technician or an engineer, then we can inherit the attributes of Employee into each subclass and set up three relations:

Secretary (Ssn, Fname, Lname, Minit, Birth_date,Address, Typing_speed)

Technician (Ssn, Fname, Lname, Minit, Birth_date,Address, Tech_grade)

Engineer (Ssn, Fname, Lname, Minit, Birth_date,Address, Eng_type)

There is no need for storing the “Job-type” discriminating attribute any more. There is no need for the Employee relation any more.

- Two relations option:

In this schema, there is no logical and reasonable option of mapping to two relations. But the designer could lump all employees except engineers in one relation and only Engineers in a separate relation if there was lot more data about engineers.

# EER to Relational Mapping- general guideline

BOTTOM LINE

- For Specialization hierarchies with one superclass and n subclasses, there are possibilities of mapping from to 1 relation to (n+1) relations.
- The mapping design is highly subjective, but must maintain appropriate attributes that determine what subclass an entity from superclass belongs to.

# Specialization / Generalization Lattice Example (UNIVERSITY)

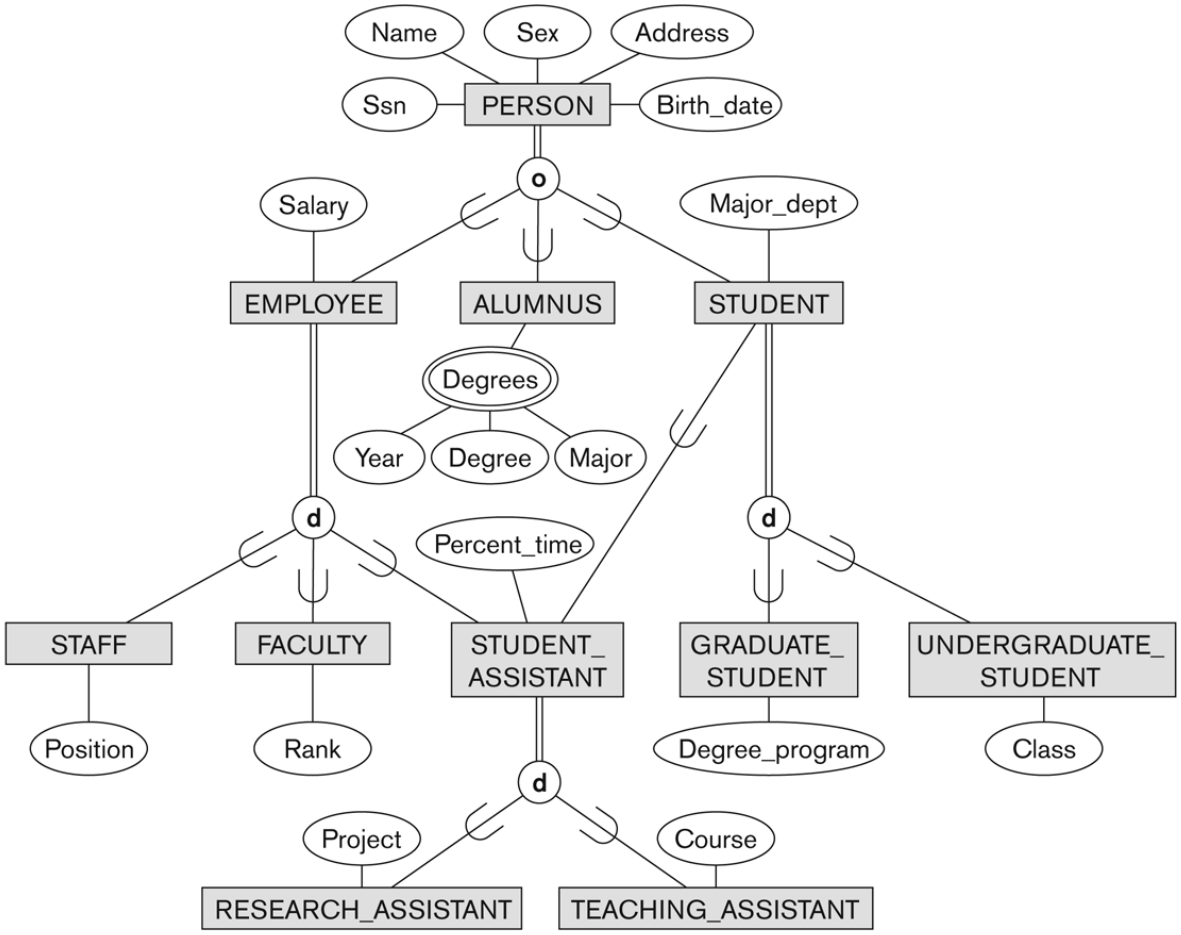



**Figure 4.7**
A specialization lattice with multiple inheritance for a UNIVERSITY database.

# Alternative Diagrammatic Notations (Appendix A in Elmasri/Navathe textbook)

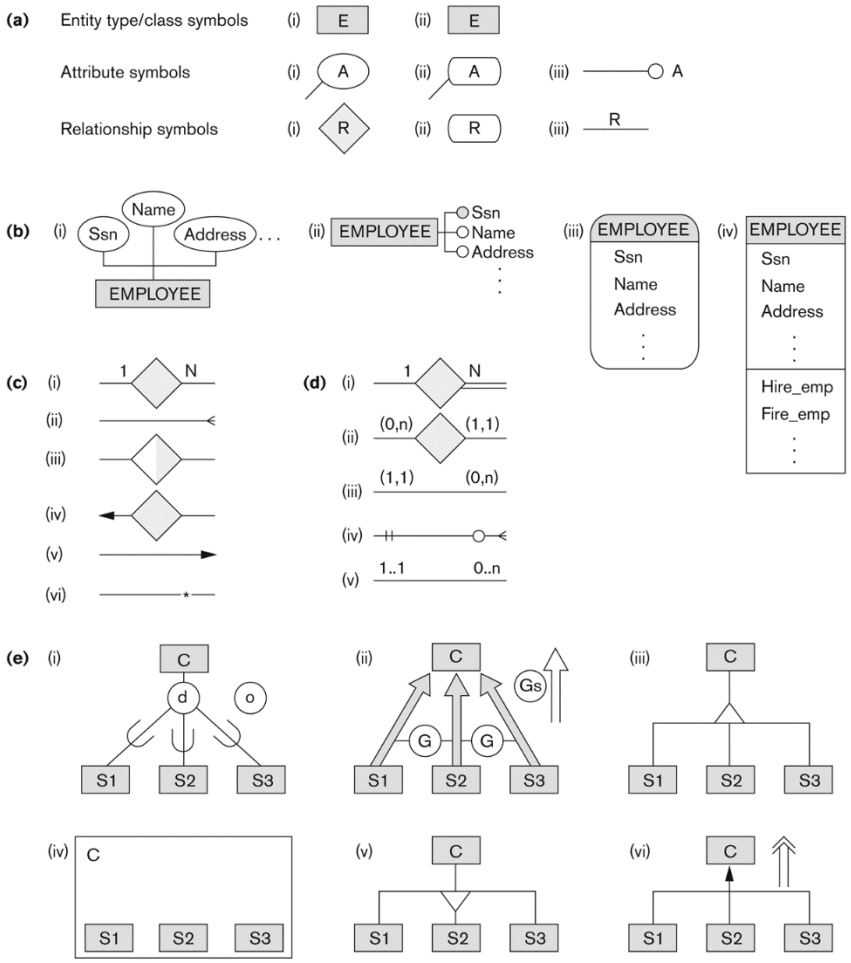



**Figure A.1**
Alternative notations. (a) Symbols for entity type/class, attribute, and relationship. (b) Displaying attributes. (c) Displaying cardinality ratios. (d) Various (min, max) notations. (e) Notations for displaying specialization/generalization.