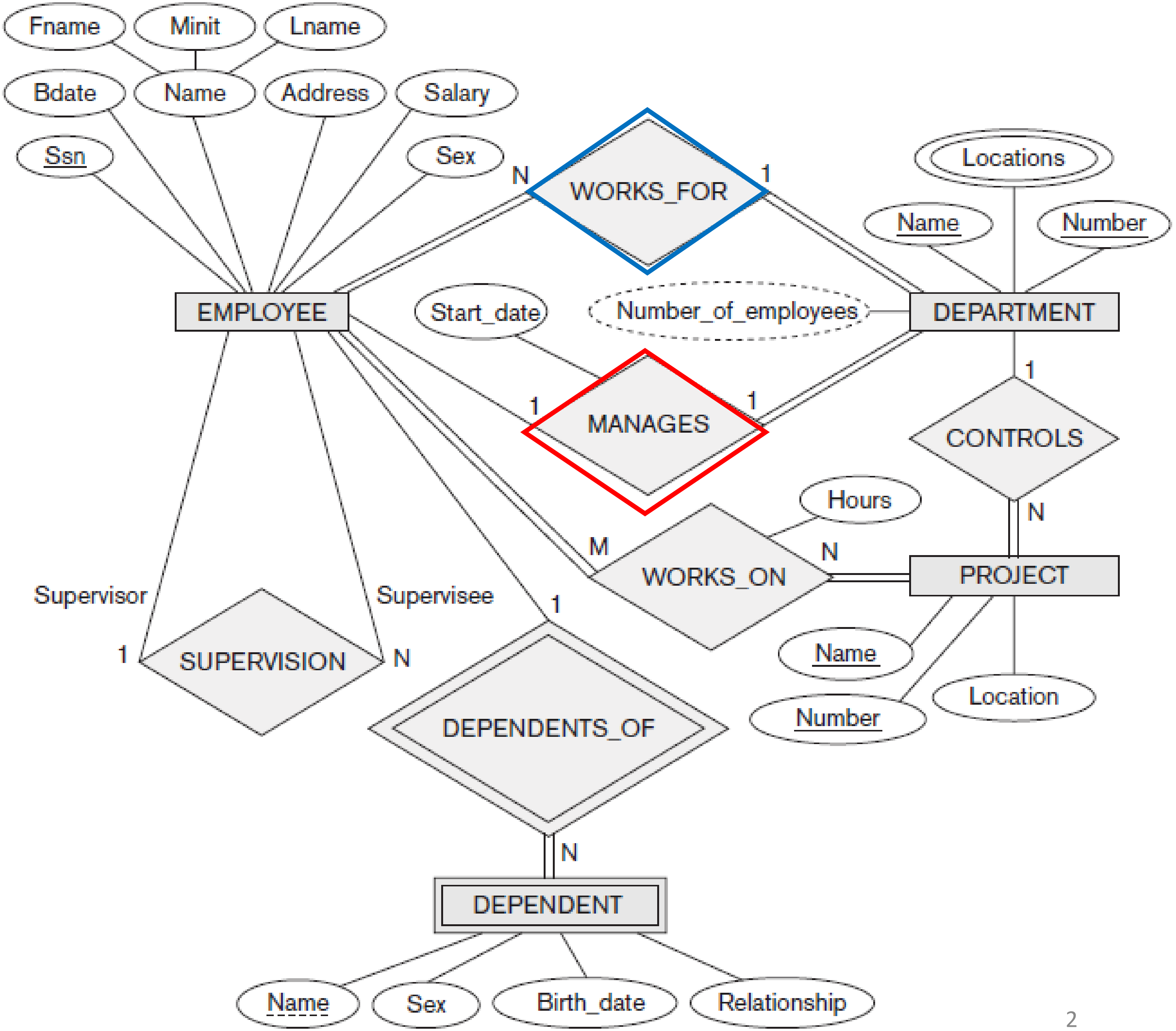
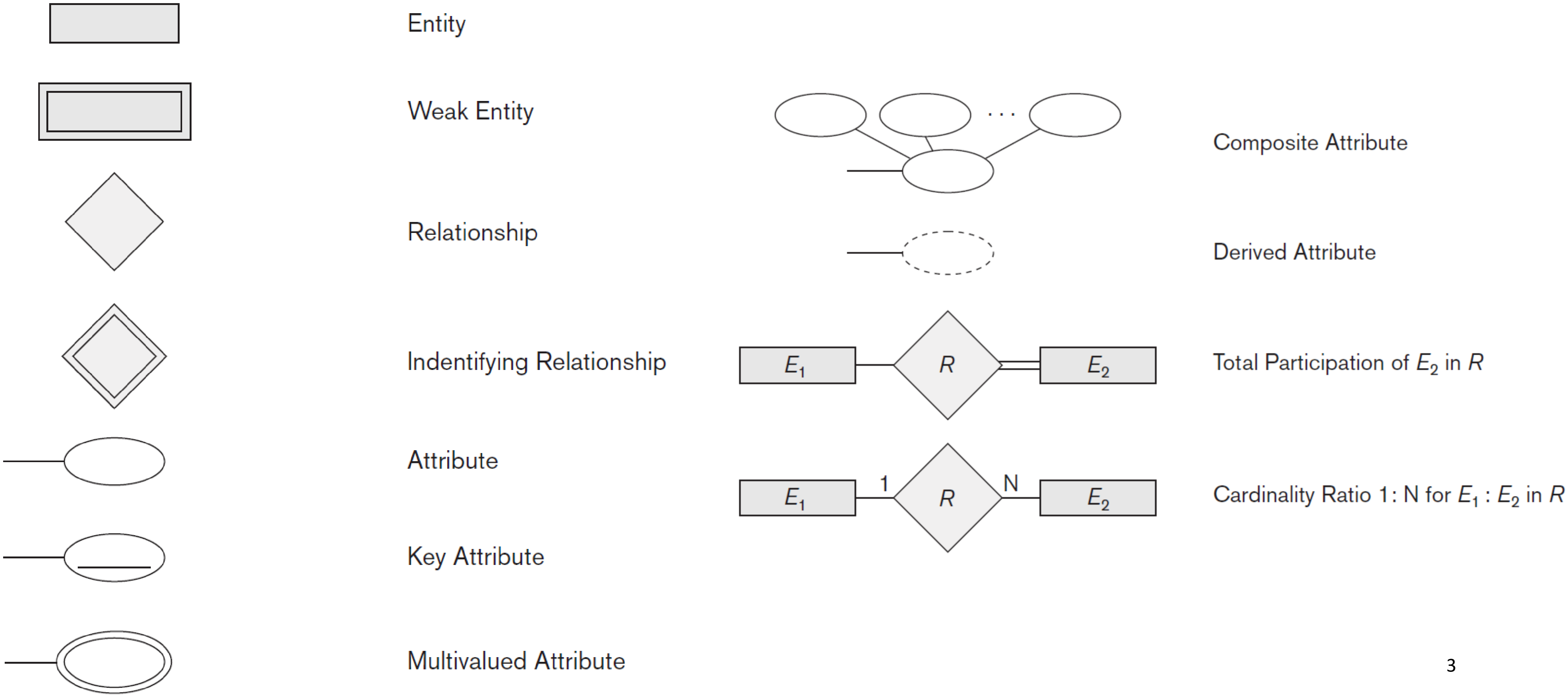
Review for ER Model in Chapter 3

- MANAGES
  - E : D = 1 : 1
  - EMPLOYEE participation is partial
  - Department participation is total
  - Start_date

- WORKS_FOR
  - D : E = 1 : N
  - Employee participation is total
  - Department participation is total



Summary of the ER Diagram Notation



Entity

Weak Entity

Relationship

Indentifying Relationship

Attribute

Key Attribute

Multivalued Attribute

Composite Attribute

Derived Attribute

Total Participation of E_2 in R

Cardinality Ratio 1: N for E_1 : E_2 in R

Chapter 4: The Enhanced Entity-Relationship (EER) Model

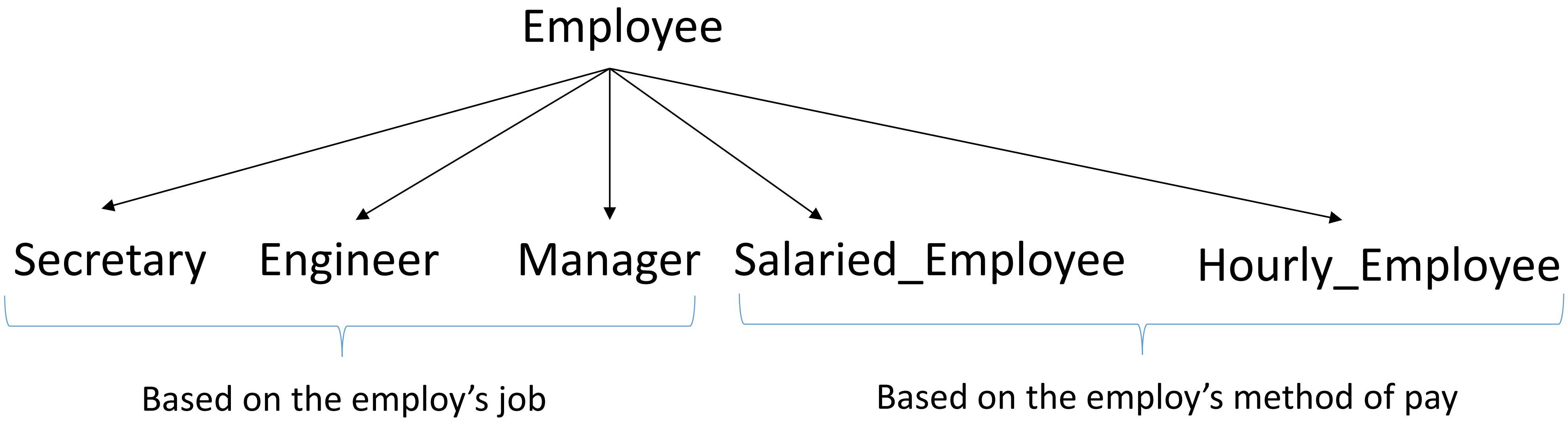
Why EER? -- Enhanced Entity Relationship

- There are limitations to the ER model
  - Sufficient for *traditional* database applications
  - Newer applications: need to reflected data properties and constraints precisely
  - More accurate database schemas are needed!

- ER model enhanced (EER) to include:
  - superclass/subclass relationships
  - type inheritance
  - specialization and generalization
  - constraints

Modeling Subclasses

- In many cases an entity set has numerous subgroupings or subsets of its entities
  - that are meaningful and need to be represented explicitly because of their significance to the database application



Employee

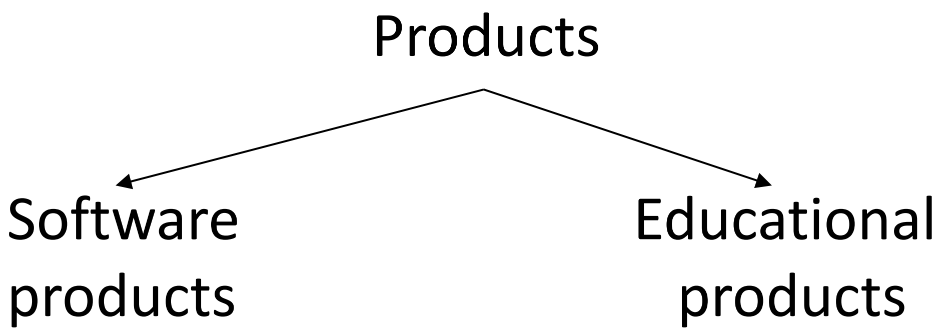
Secretary Engineer Manager Salaried_Employee Hourly_Employee

Based on the employ’s job

Based on the employ’s method of pay

Modeling Subclasses

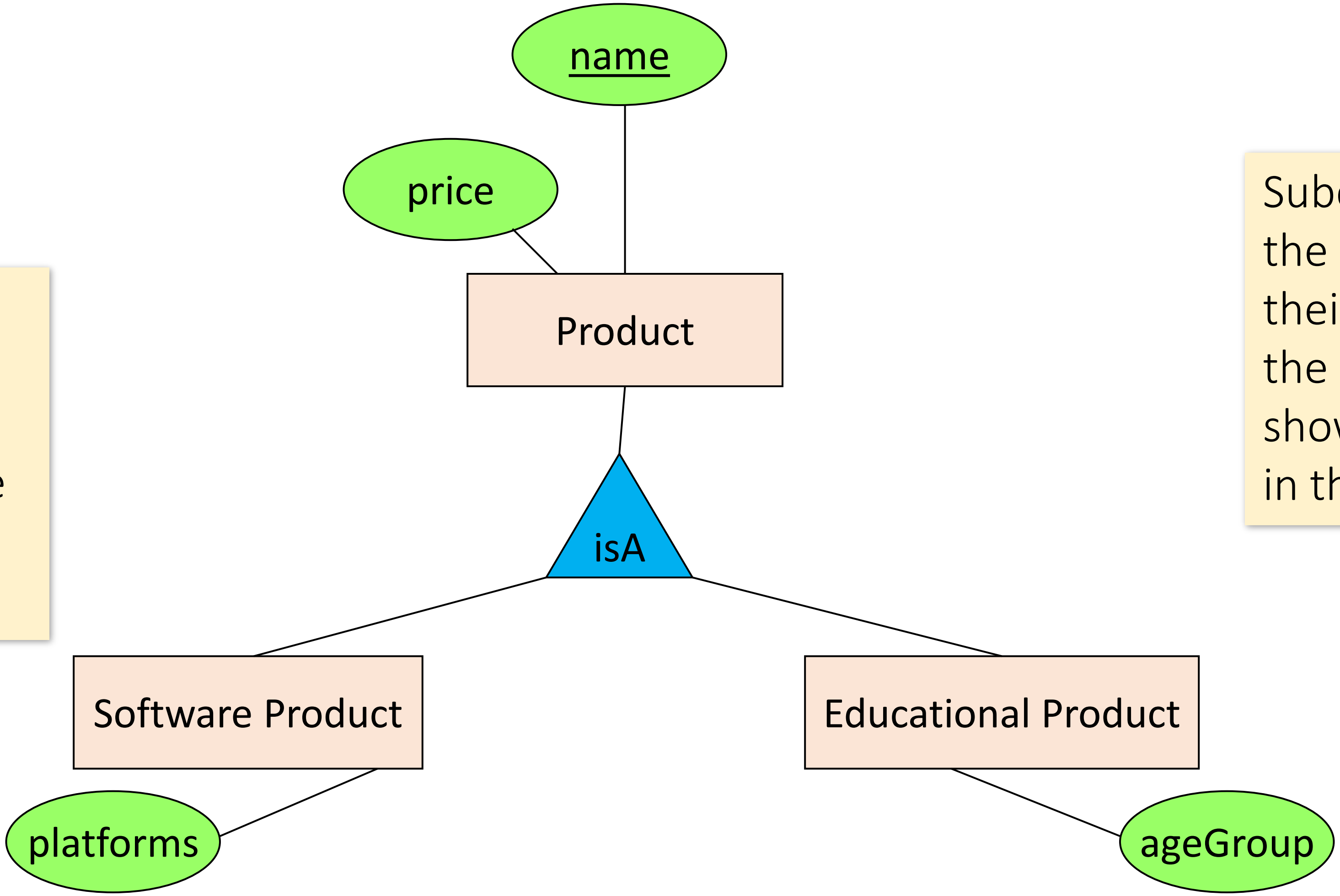
- Some objects in a class may be special, i.e. worthy of their own class
  - Define a new class?
    - *But what if we want to maintain connection to current class?*
  - Better: define a *subclass*
    - *Ex:*



We can define **subclasses** in E/R!

Modeling Subclasses

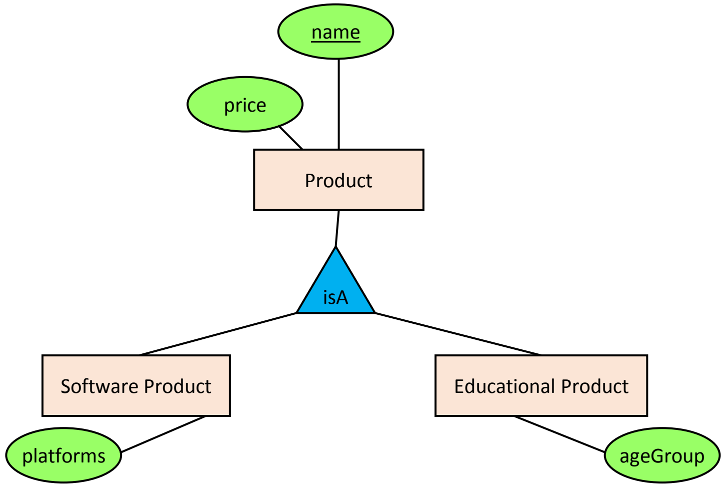
A member entity of the subclass represents the same real-world entity as some member of the superclass, but in a distinct specific role



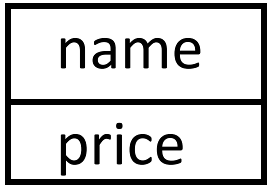
Subclasses contain all the attributes of *all* of their parent classes **plus** the new attributes shown attached to them in the E/R diagram

These are called superclass / subclass (IS-A) relationships

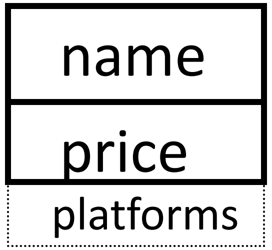
Understanding Subclasses



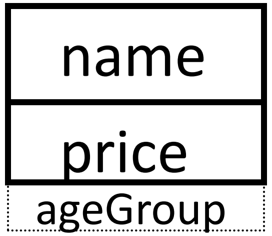
- Think in terms of records; ex:
  - Product



  - SoftwareProduct

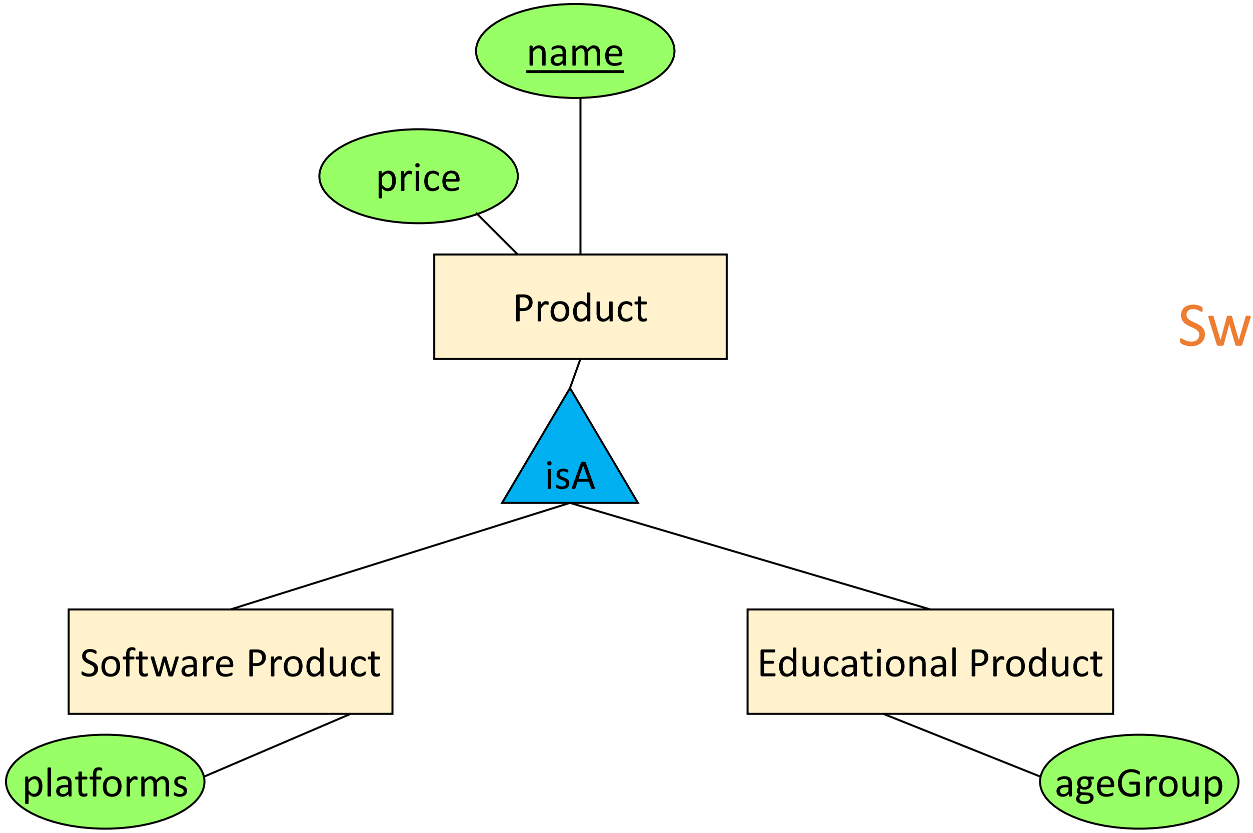


  - EducationalProduct



Child subclasses contain all the attributes of *all* of their parent classes **plus** the new attributes shown attached to them in the E/R diagram

Think like tables…



Product

name	price
Gizmo	99
Camera	49
Toy	39

Sw.Product

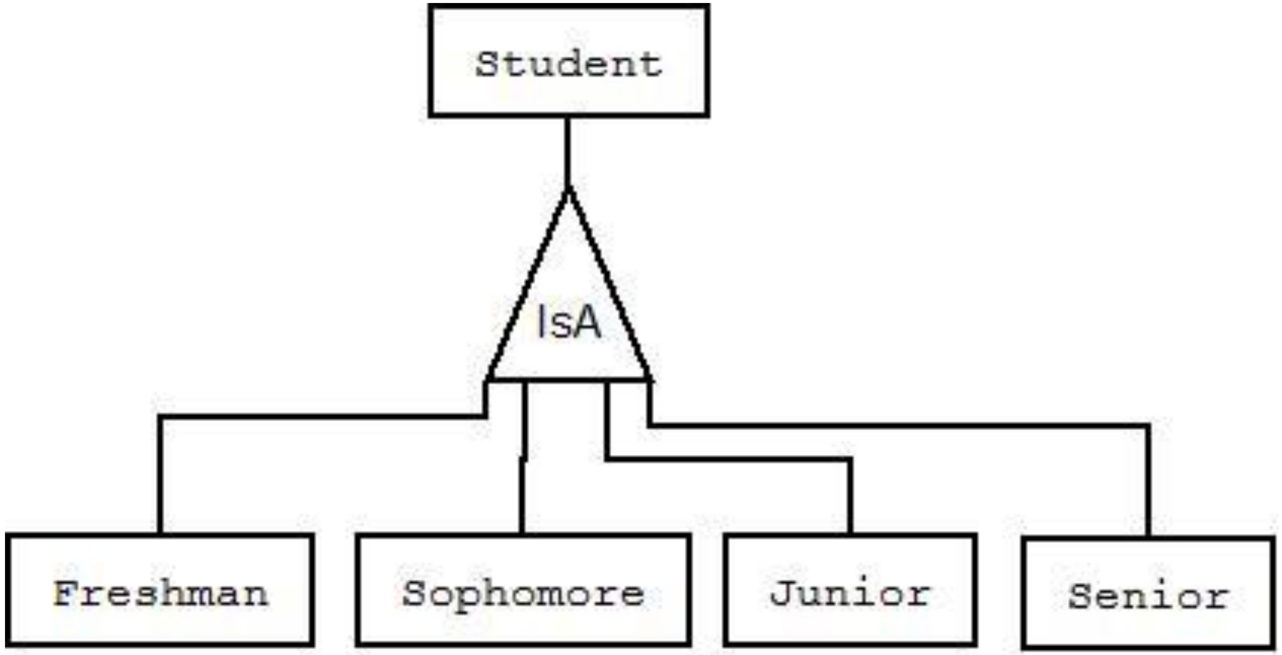
name	price	platforms
Gizmo	99	unix

Ed.Product

name	price	ageGroup
Camera	49	todler
Toy	39	retired

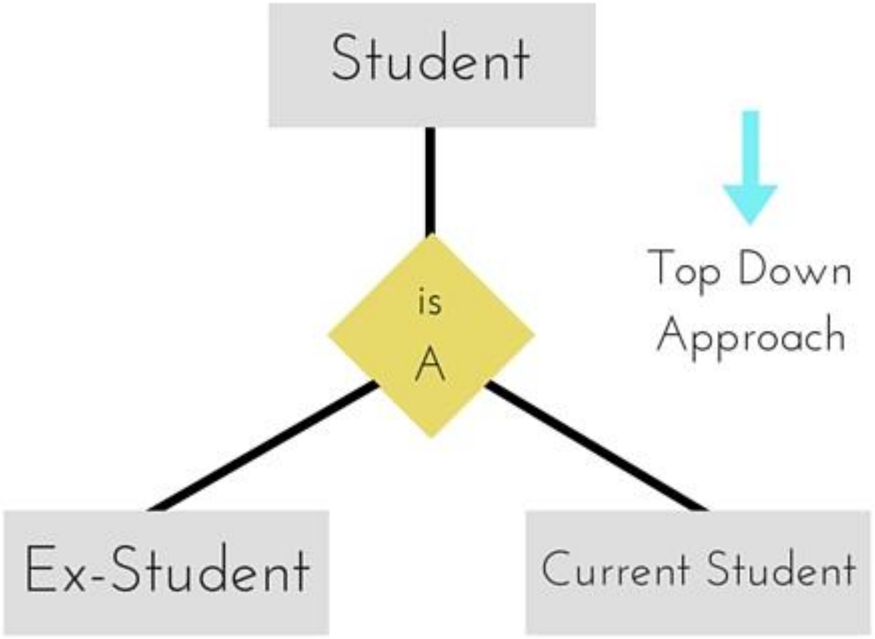
Inheritance in Superclass / Subclass Relationships

- An entity that is a member of a subclass inherits all attributes and relationships of the entity as a member of the superclass



Specialization

- Is the process of defining a set of subclasses of a superclass



May have several specializations of the same superclass

The set of subclasses is based upon some distinguishing characteristics of the entities in the superclass

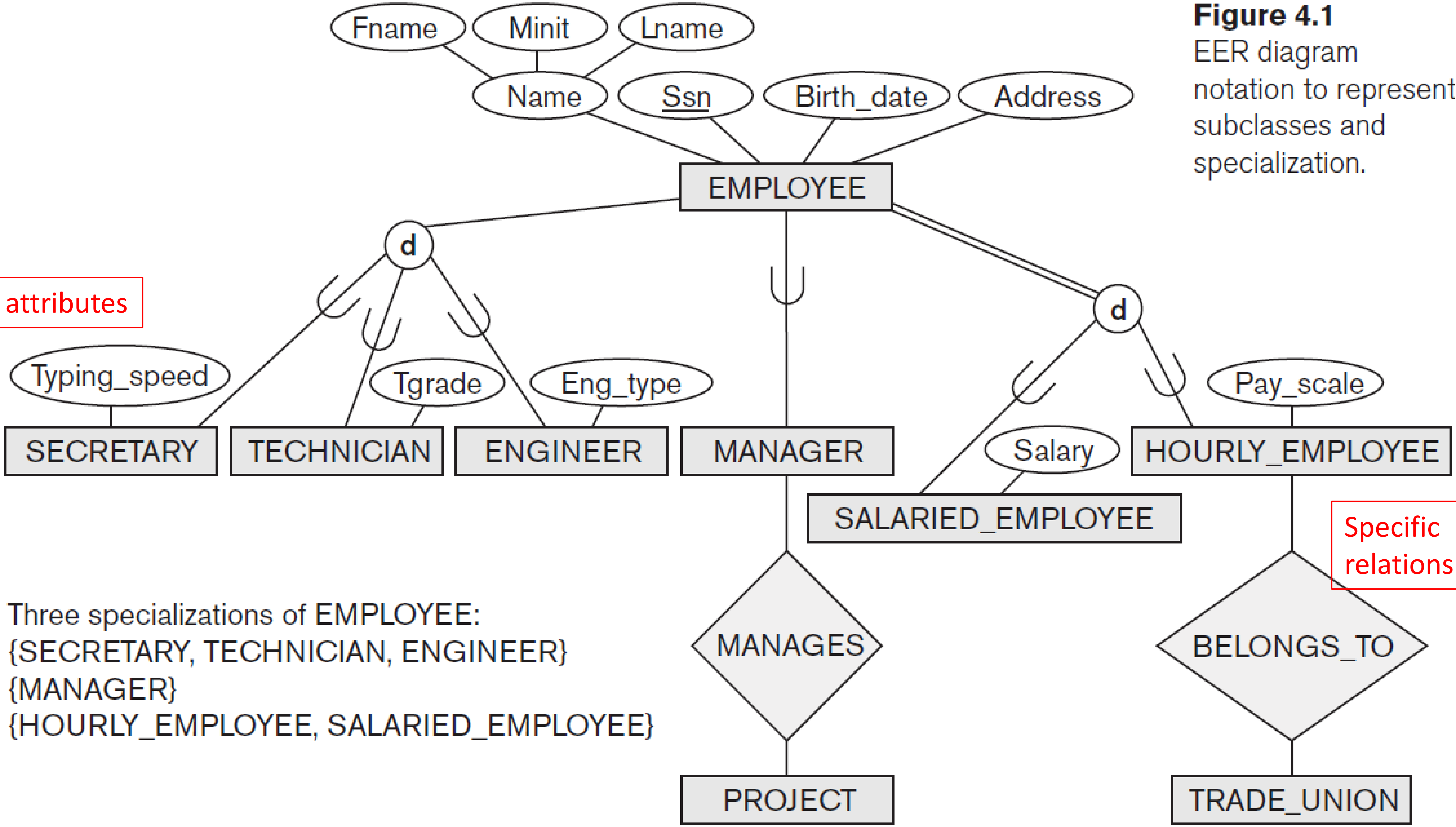


Figure 4.1
EER diagram notation to represent subclasses and specialization.

local attributes

Specific relationships

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

Reasons for Introducing Class/Subclass Relationships and Specialization

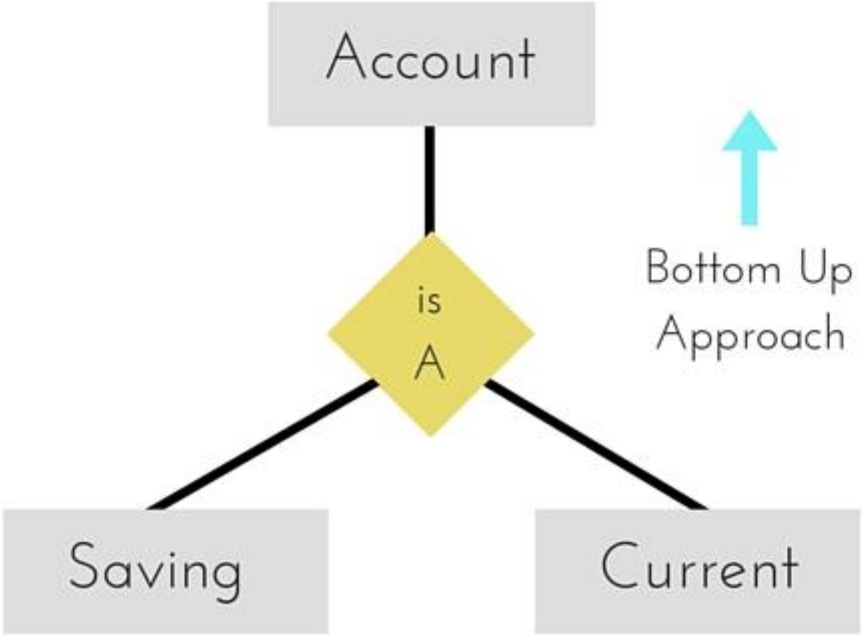
1) Certain attributes may apply to some but not all entities of the superclass entity type
   - A subclass is defined in order to group the entities to which these attributes apply.
   - See example in last slide

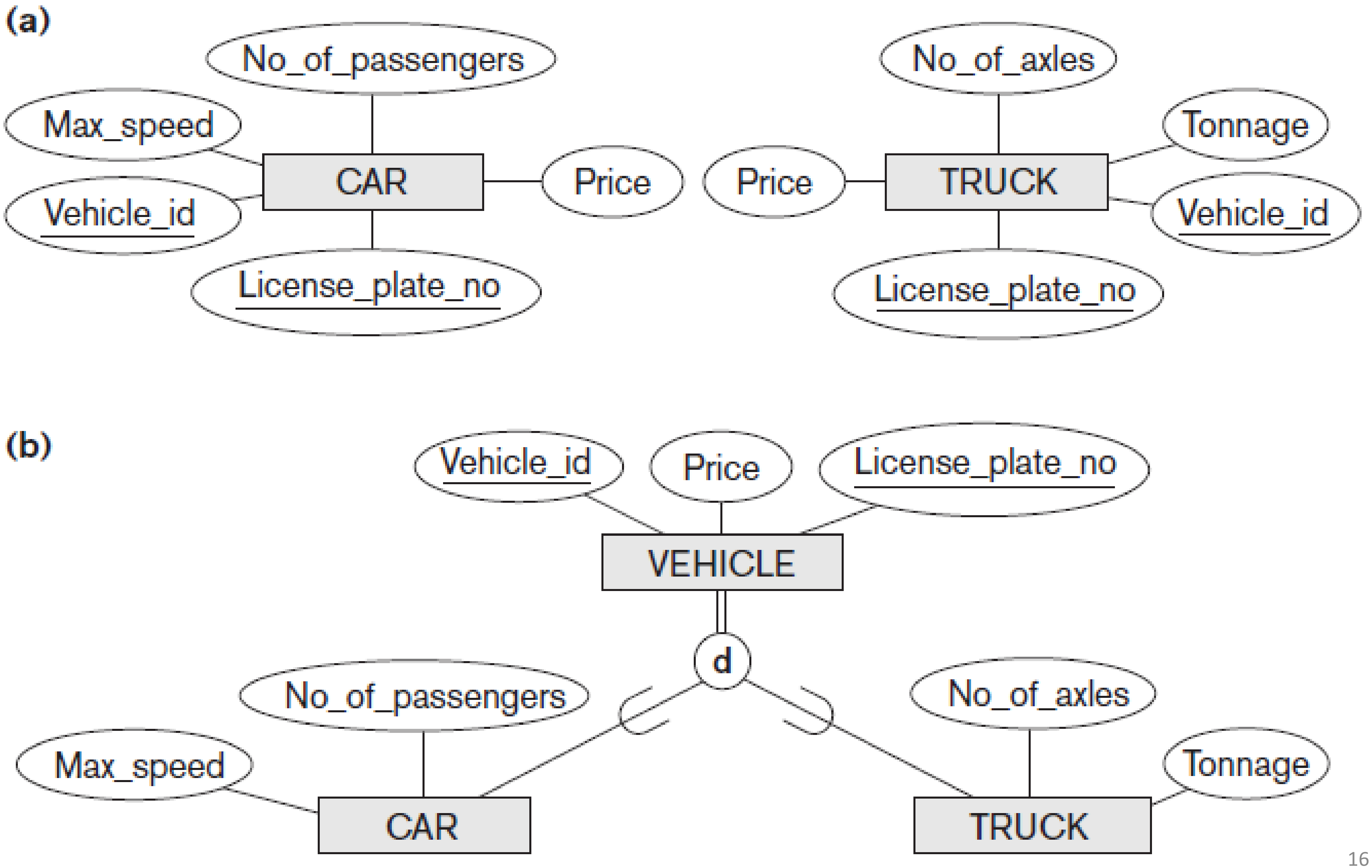
2) Some relationship types may be participated in only by entities that are members of the subclass
   - See example in last slide

Generalization

- Generalization is the reverse of the specialization process
- Several classes with common features are generalized into a superclass
  - Original classes become its subclasses

Suppress the differences among several entity sets, identify their common features, and generalize them into a single superclass




(a)
No_of_passengers
Max_speed
CAR
Price
Vehicle_id
License_plate_no
No_of_axles
Tonnage
Price
TRUCK
Vehicle_id
License_plate_no
(b)
Vehicle_id
Price
License_plate_no
VEHICLE
d
No_of_passengers
Max_speed
CAR
No_of_axles
Tonnage
TRUCK

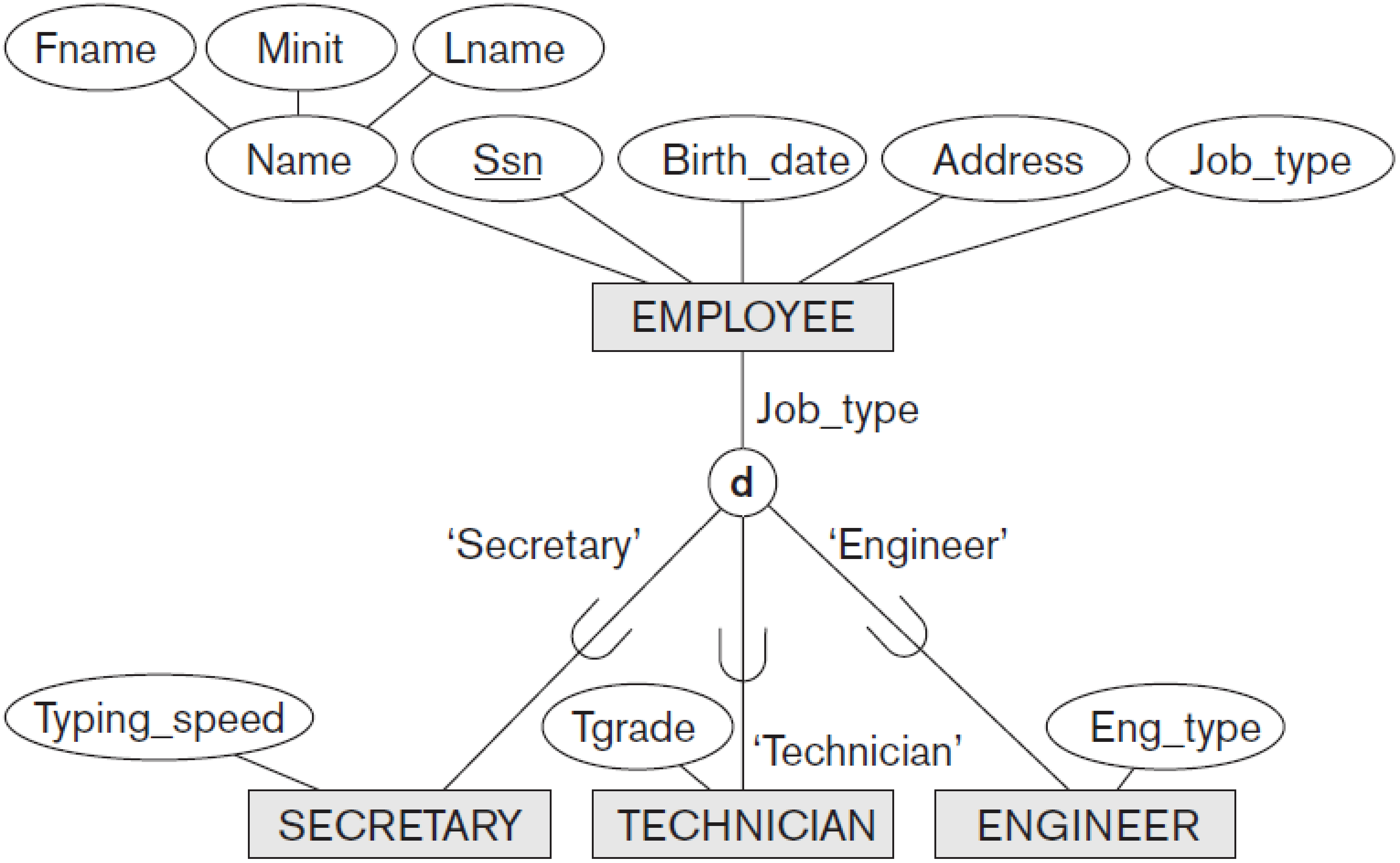
Constraints on Specialization and Generalization

1. **Membership** constraint

- If we place a **condition** on the value of some attributes of the superclass, then we can exactly determine the entities that will become members of each **subclass**
  - Such subclasses are called **condition-defined (attribute-defined)**

- Condition is used to determine Membership!

Constraints on Specialization and Generalization (cont.)



- We can specify the condition of **membership** in the SECRETARY subclass by the condition:

  (Job_type = 'Secretary')

Constraints on Specialization and Generalization (cont.)

2. **Disjointness** constraint

- Specifies that the subclasses of the specialization must be disjoint sets
  - An entity can be a member of at most one of the subclasses of the specialization
  - Can I conclude: A specialization that is attribute-defined implies disjointness constraint?
  - Specified by *d* in EER diagram

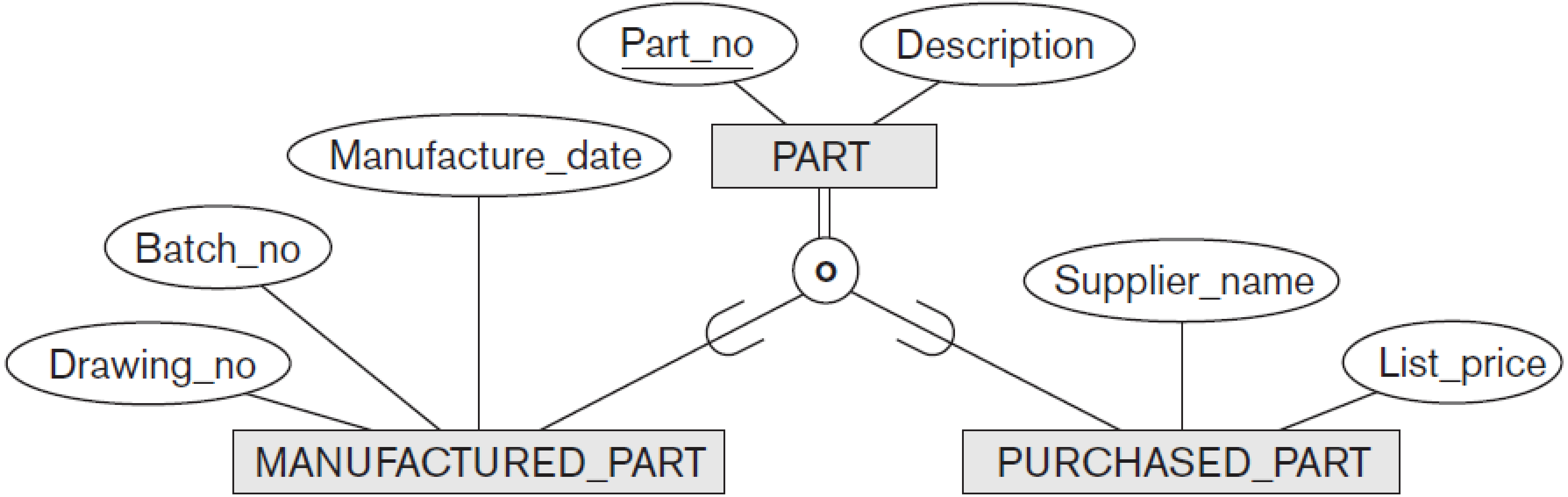
Constraints on Specialization and Generalization (cont.)

2. Disjointness constraint (cont.)

- If the subclasses are not constrained to be disjoint, their sets of entities may be **overlapping**
  - 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 (cont.)

2. **Disjointness** constraint (cont.)



Constraints on Specialization and Generalization (cont.)

3. **Completeness** constraint (cont.)

- A **total specialization** constraint specifies:
  - *Every* entity in the superclass must be a member of *at least one* subclass in the specialization
  - Shown in EER diagrams by a *double line*

- Likewise, a **partial specialization** allows an entity in the superclass not to belong to any of the subclasses
  - Shown in EER diagrams by a *single line*

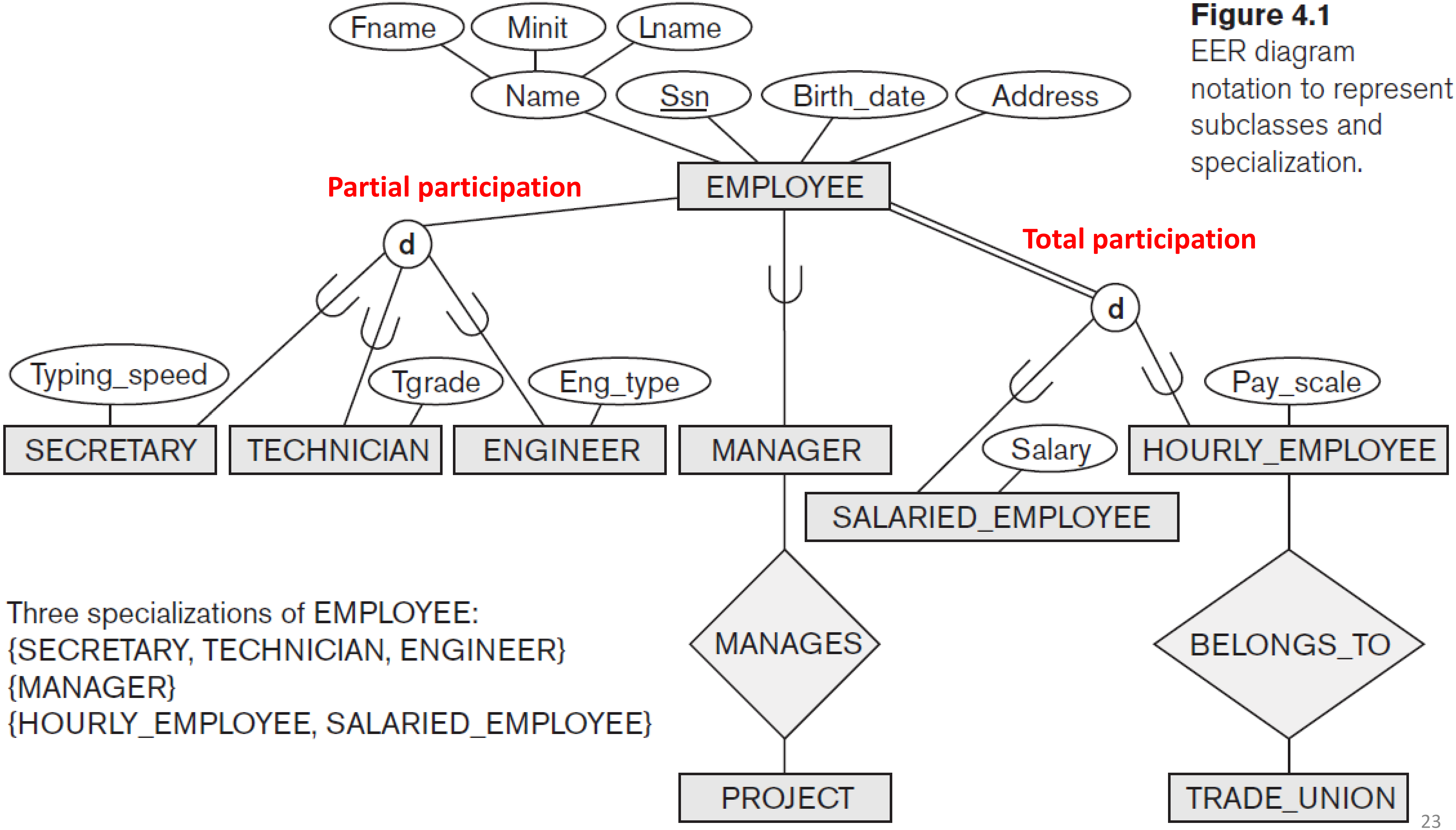


Figure 4.1
EER diagram notation to represent subclasses and specialization.

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

Constraints on Specialization and Generalization (cont.)

- Certain insertion and deletion rules apply to specialization (and generalization) as a consequence of the constraints
  - Deleting an entity from a superclass implies that it is automatically deleted from all the subclasses to which it belongs.
  - Inserting an entity in a superclass implies that the entity is mandatorily inserted in all attribute-defined subclasses for which the entity satisfies the defining predicate.
  - Inserting an entity in a superclass of a total specialization implies that the entity is mandatorily inserted in at least one of the subclasses of the specialization.

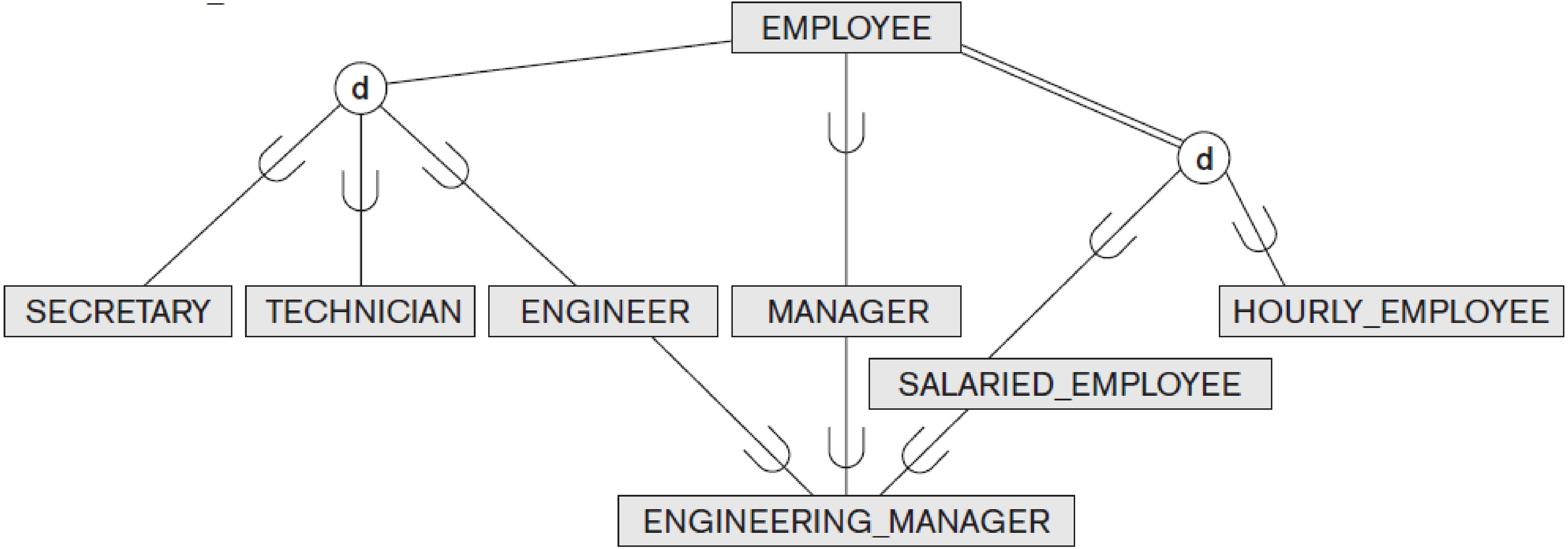
Specialization and Generalization Hierarchies and Lattices

- A subclass may have further subclasses specified on it
  - Forming a **hierarchy** or **lattice** of specializations

- **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)

Figure 4.6
A specialization lattice with shared subclass ENGINEERING_MANAGER.



Hierarchy or Lattice?

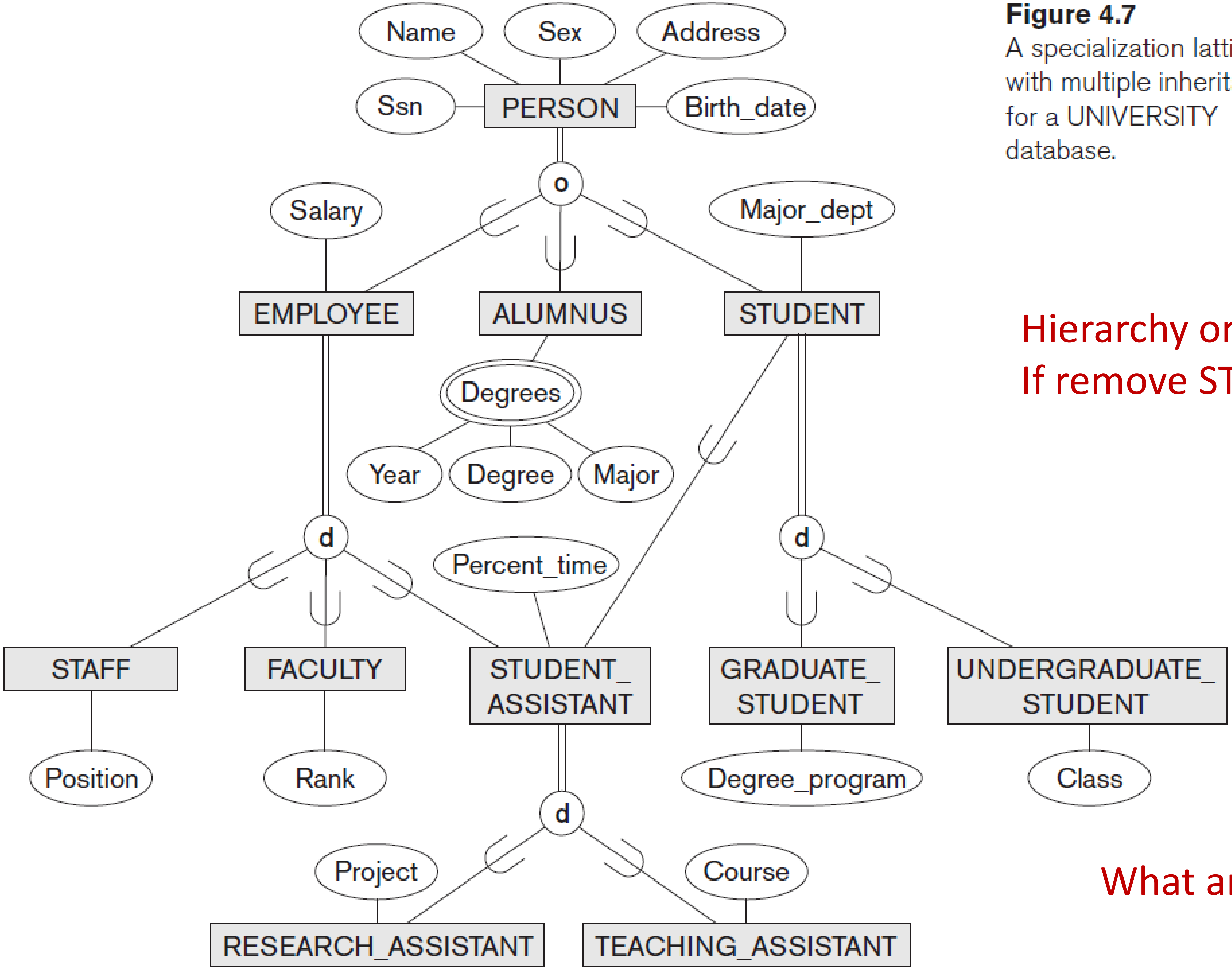


Figure 4.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

Hierarchy or Lattice?

If remove STUDENT_ASSISTANT?

What are the requirements?

Specialization and Generalization Hierarchies and Lattices (cont.)

- A subclass inherits the attributes not only of its direct superclass, but also of all its predecessor superclasses all the way to the root of the hierarchy or lattice if necessary
    - Forming a **hierarchy** or **lattice** of specializations