Enhanced Entity-Relationship (EER) Modeling

Introduction

Enhanced Entity Relationship (EER) model is an extension of the original ER model

Why do we need EER (Enhanced Entity Relationship) Model ?

Which concepts and relationships cannot be captured by ER Model ?

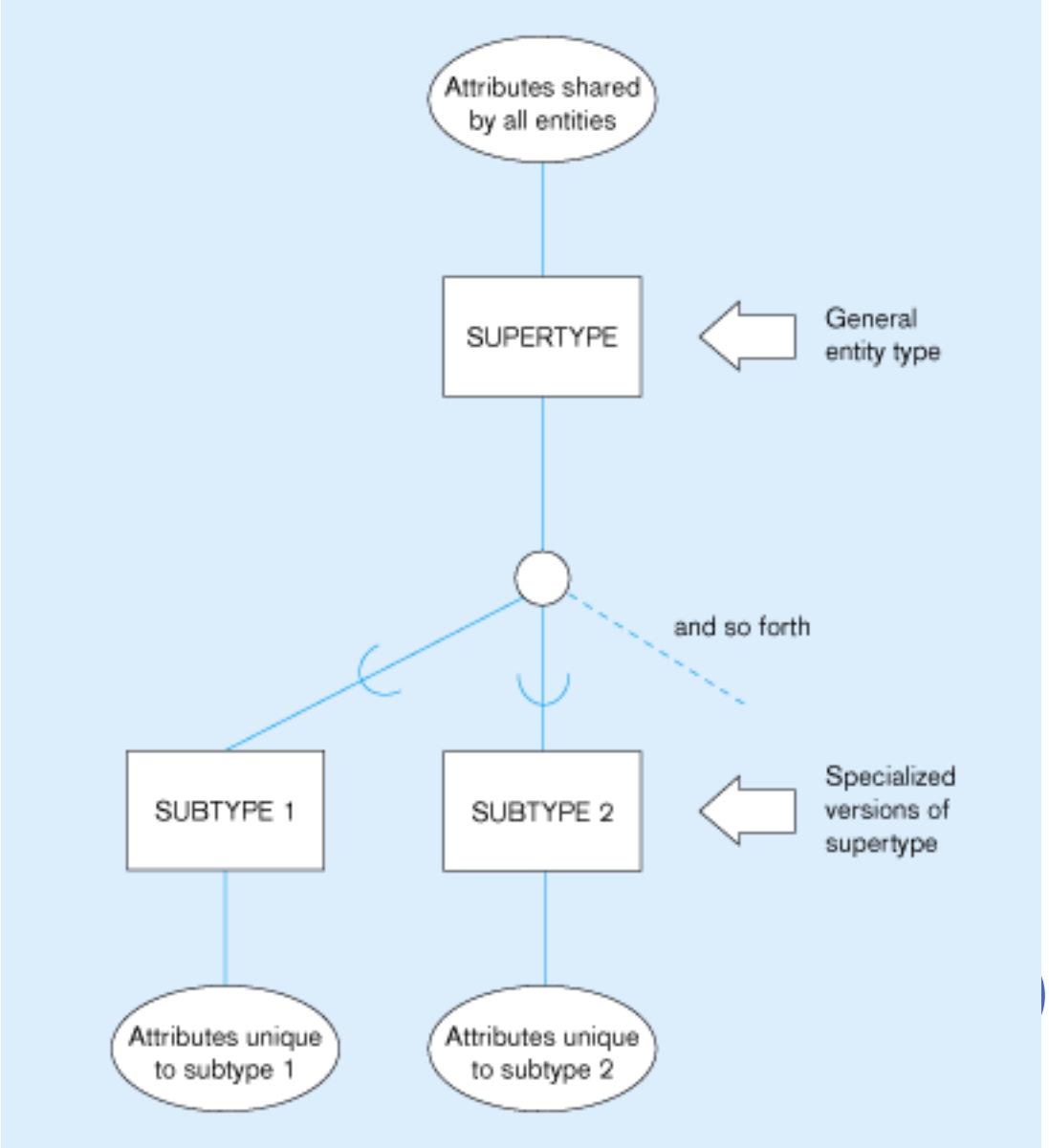
Basic notation for supertype/subtype relationships

Model a general entity type (**supertype**)

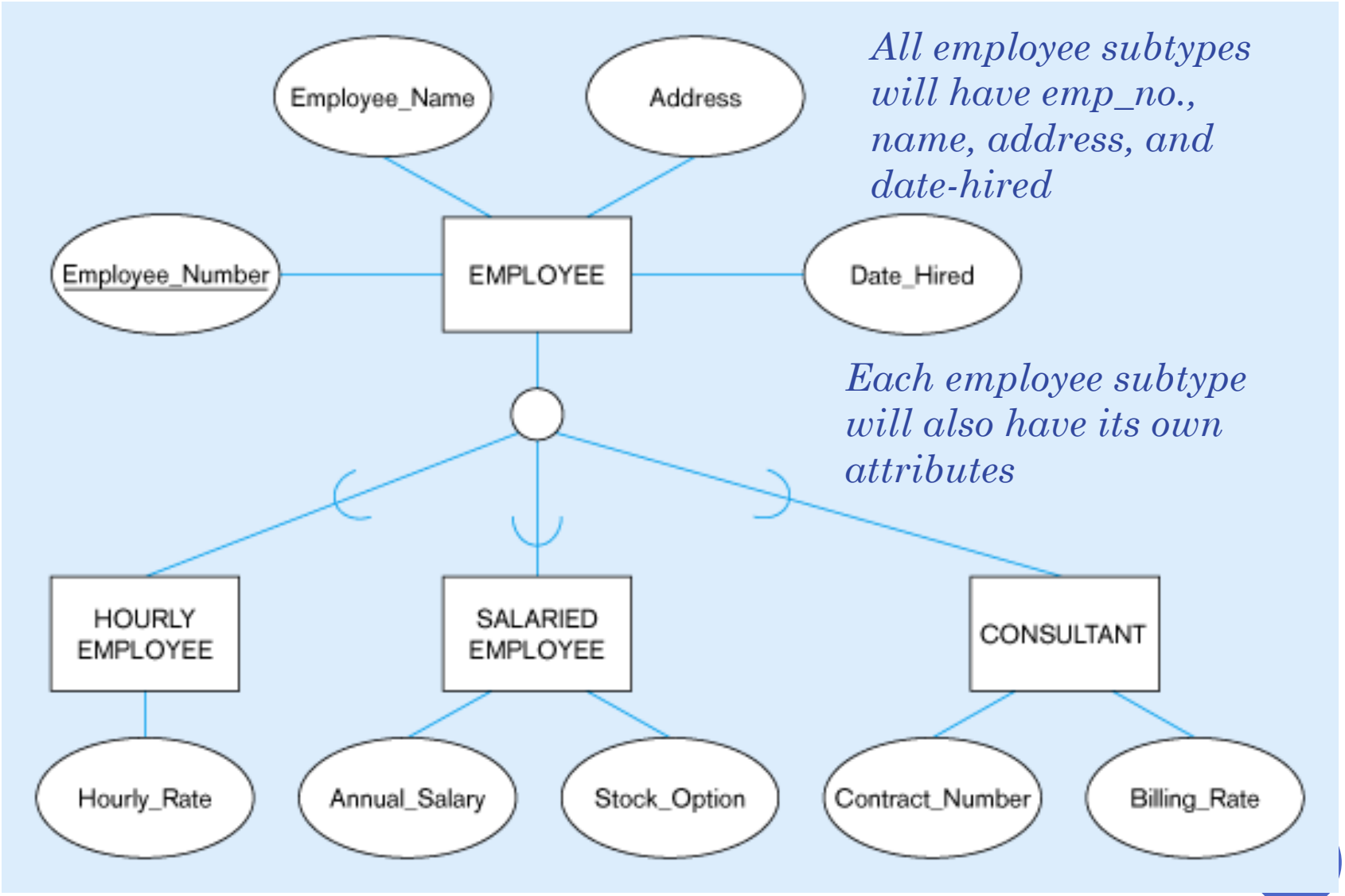
and

Then subdivide it into several specialised entity types (**subtypes**)

Each subtype **inherits** from its supertype and may have ***special attributes of its own***

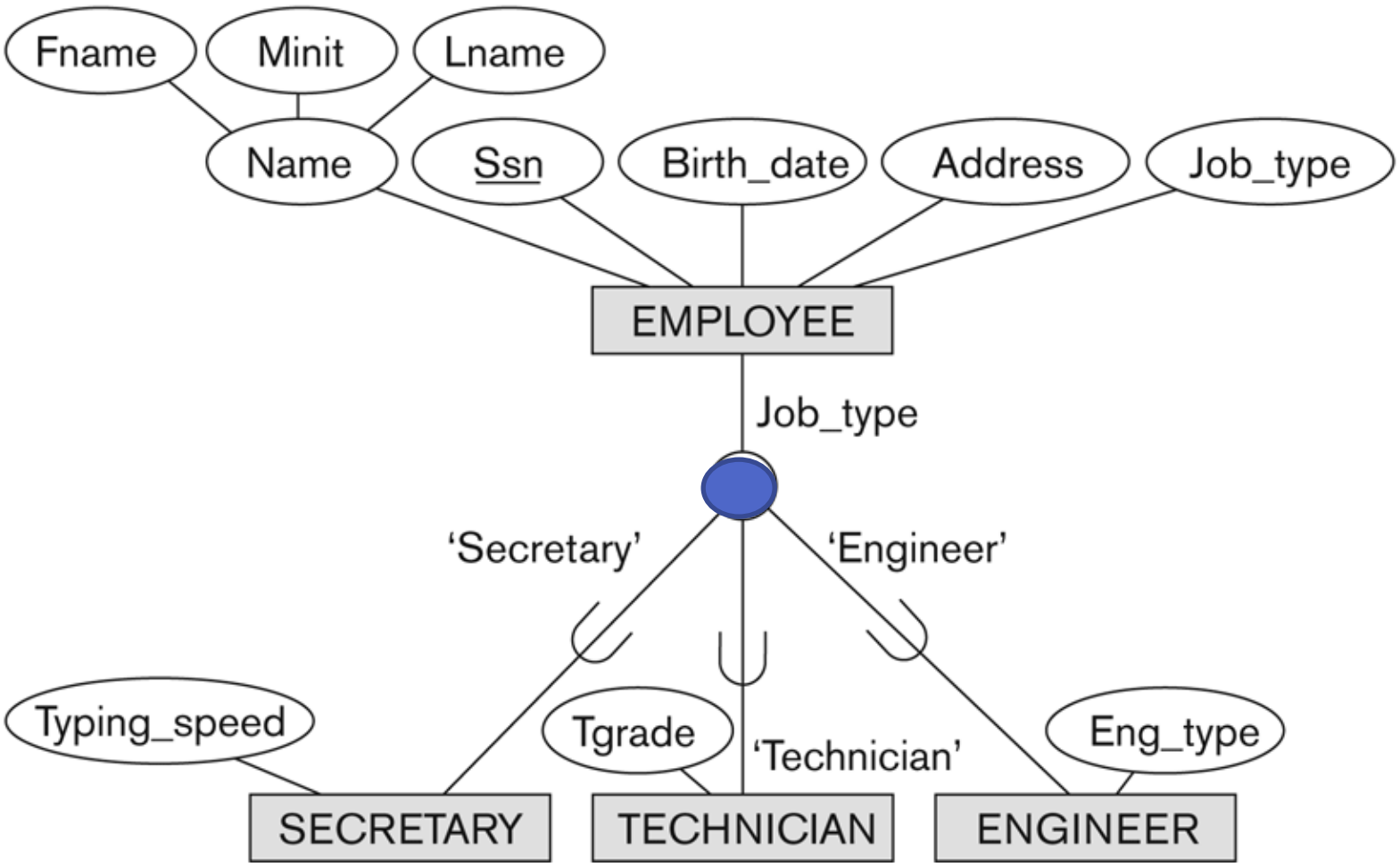


Employee supertype with three subtypes



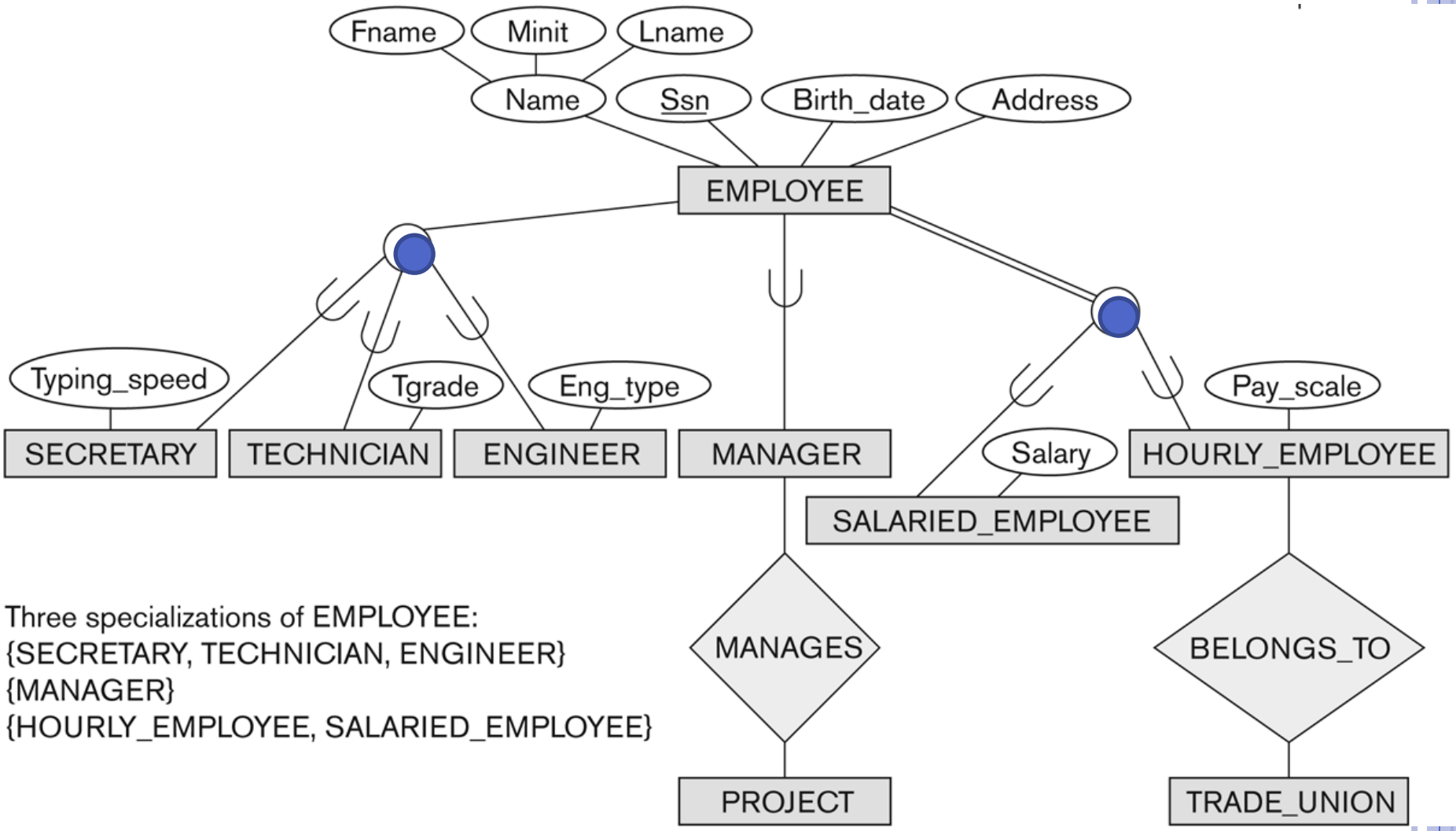
Subclasses and Superclasses

An entity type may have different sub-groupings



Subclasses and Superclass

An entity type may have different sub-groupings

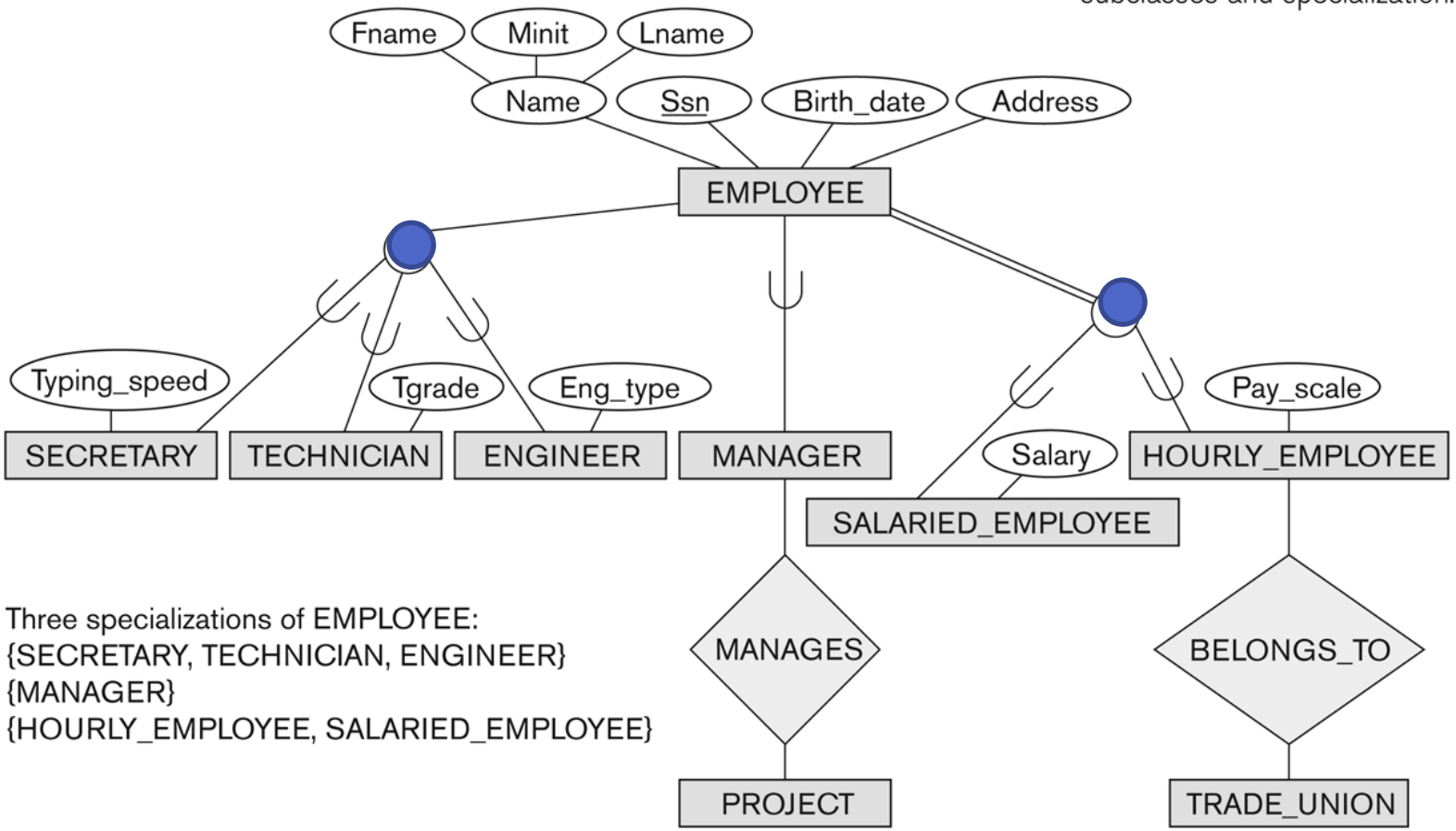


Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

Subclasses and Superclasses

A salaried employee who is also an engineer belongs to the two subclasses: **ENGINEER, and SALARIED_EMPLOYEE**

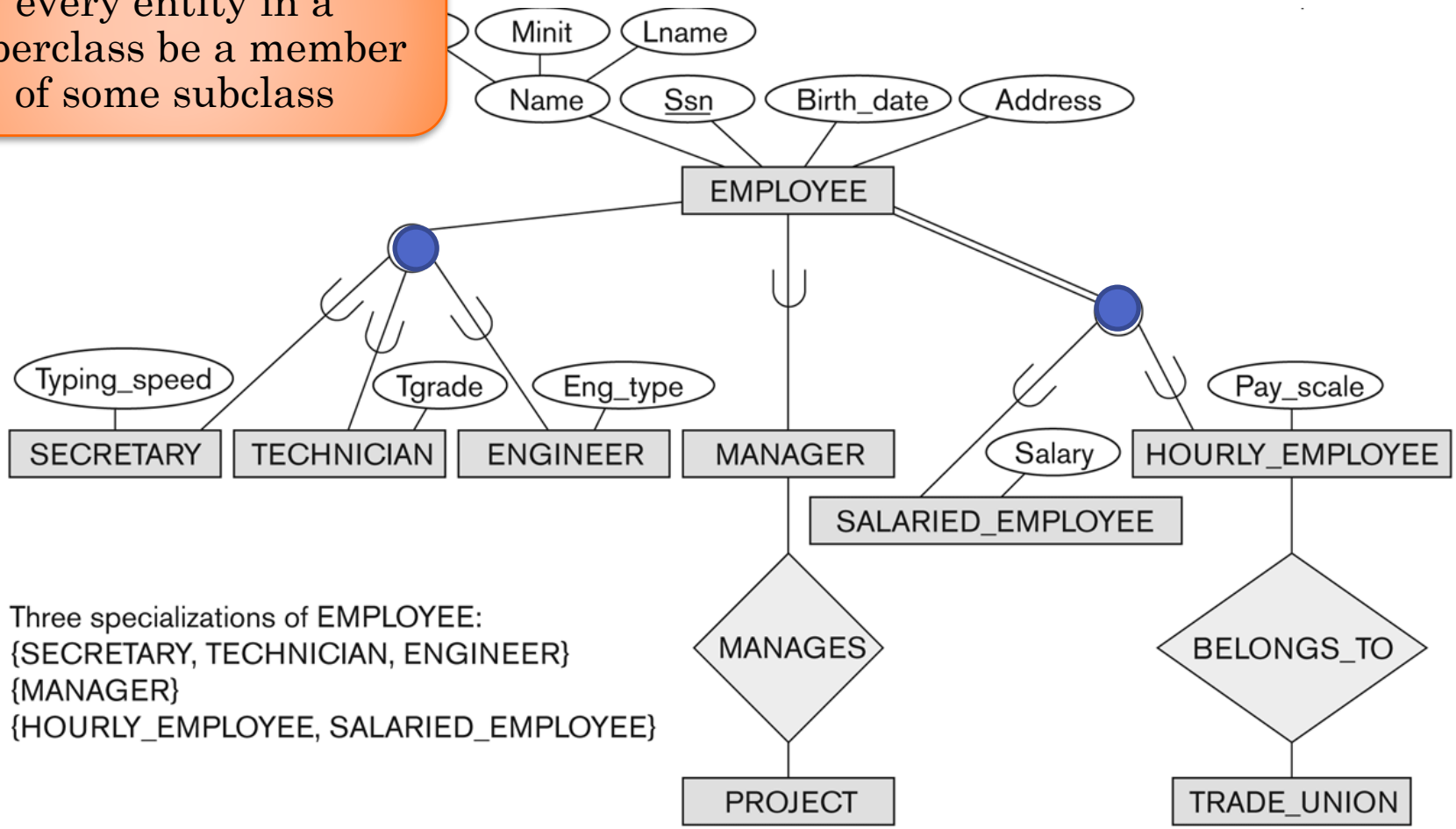
Figure 4.1
EER diagram notation to represent subclasses and specialization.



Subclasses and Superclasses

A salaried employee who is also an engineering manager belongs to the three subclasses

It is not necessary that every entity in a superclass be a member of some subclass



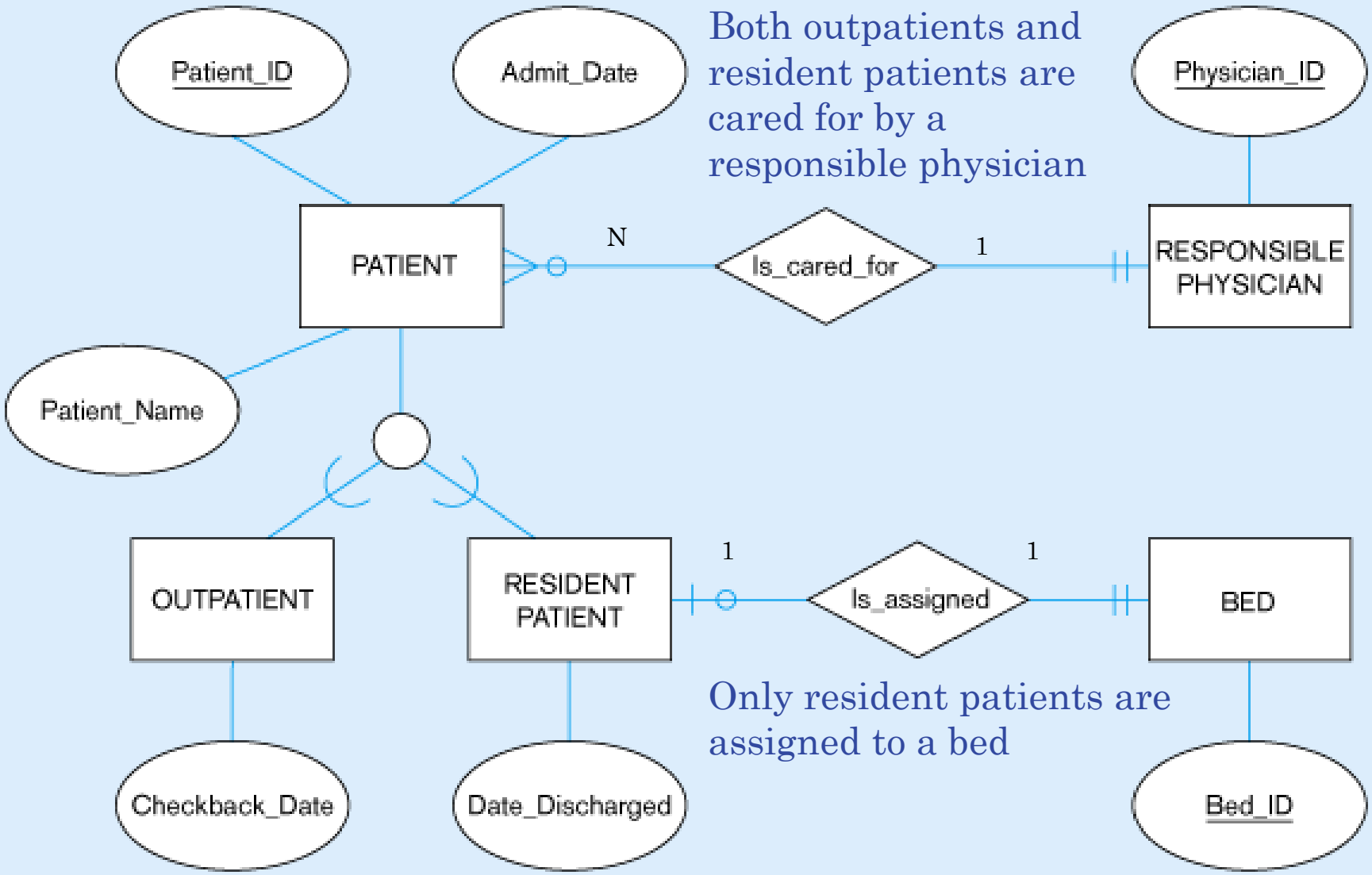
Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

When to use supertype/subtype relations

We use subtypes when

- There are attributes that apply to some (but not all) of the instances of an entity type
- The instances of a ***subtype*** participate in a relationship unique to that subtype

Patients ER (Crow Foot's Notation)



Generalization and specialization

Generalization

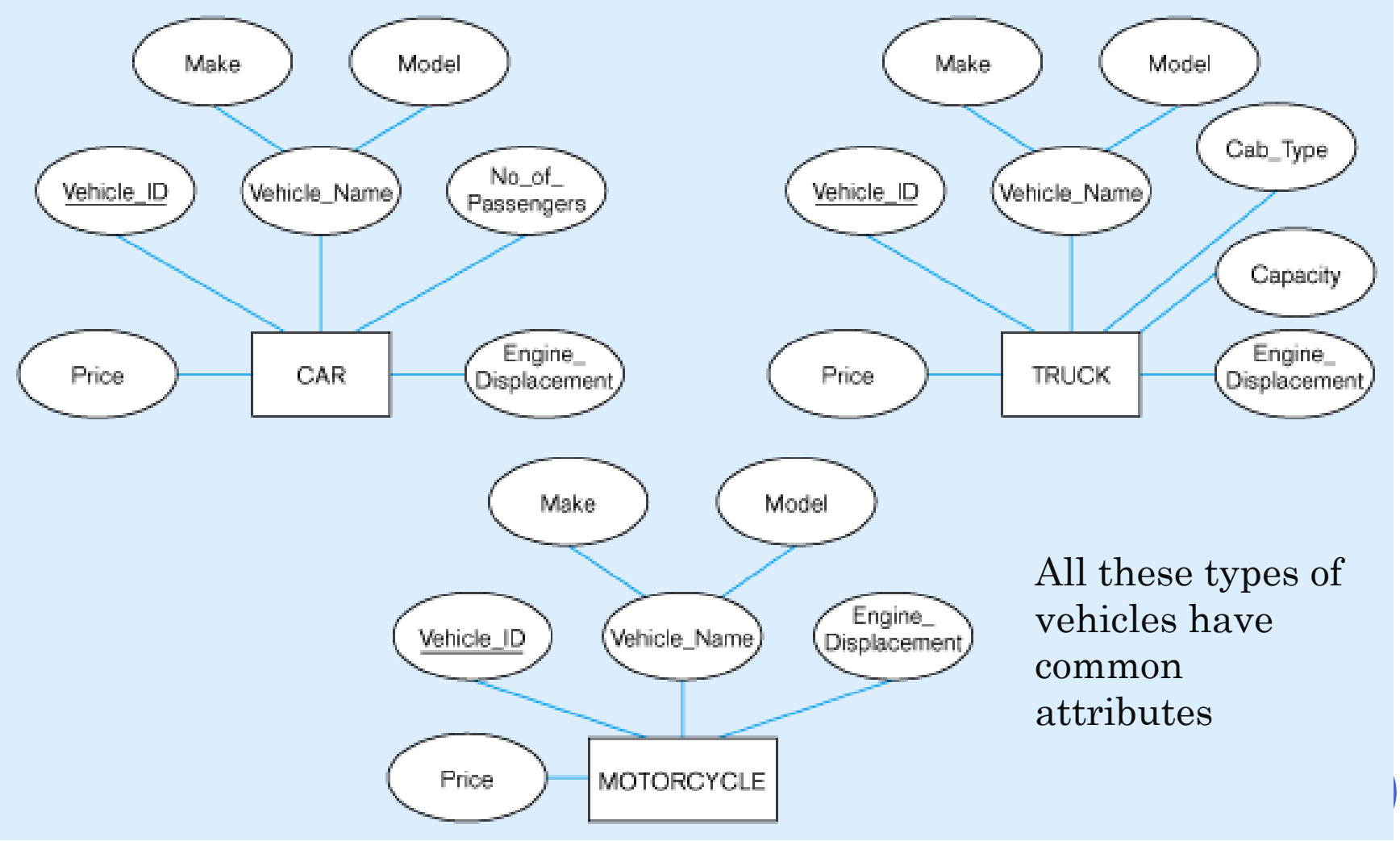
- The process of defining a more general entity type from a set of more specialized entity types.
- BOTTOM-UP

Specialization

- The process of defining one or more subtypes of the supertype
- TOP-DOWN

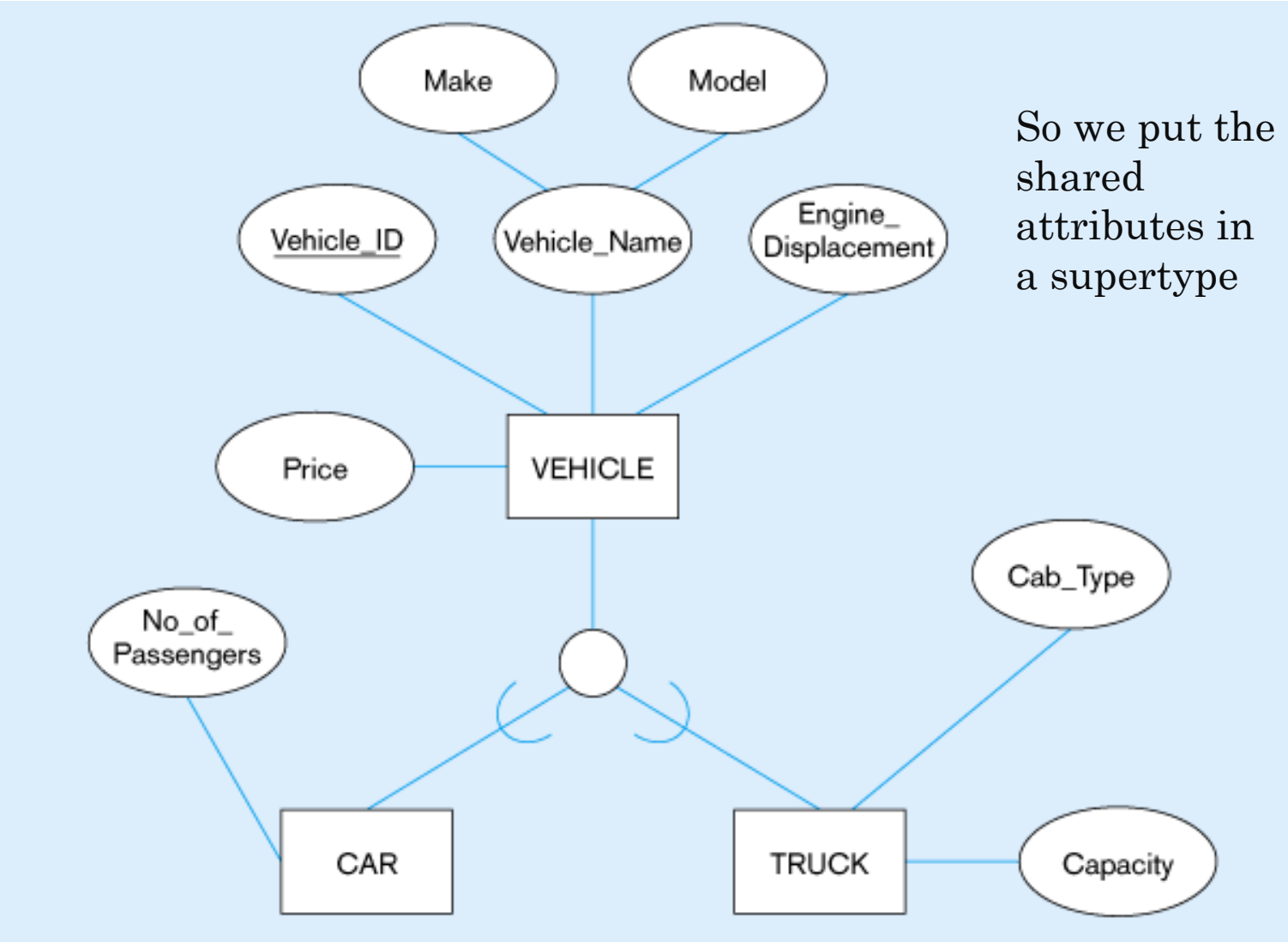
Example of Generalization

Three entity types: CAR, TRUCK, and MOTORCYCLE



All these types of vehicles have common attributes

Generalization to VEHICLE supertype



So we put the shared attributes in a supertype

Note: no subtype for motorcycle, since it has no unique attributes

Specialisation

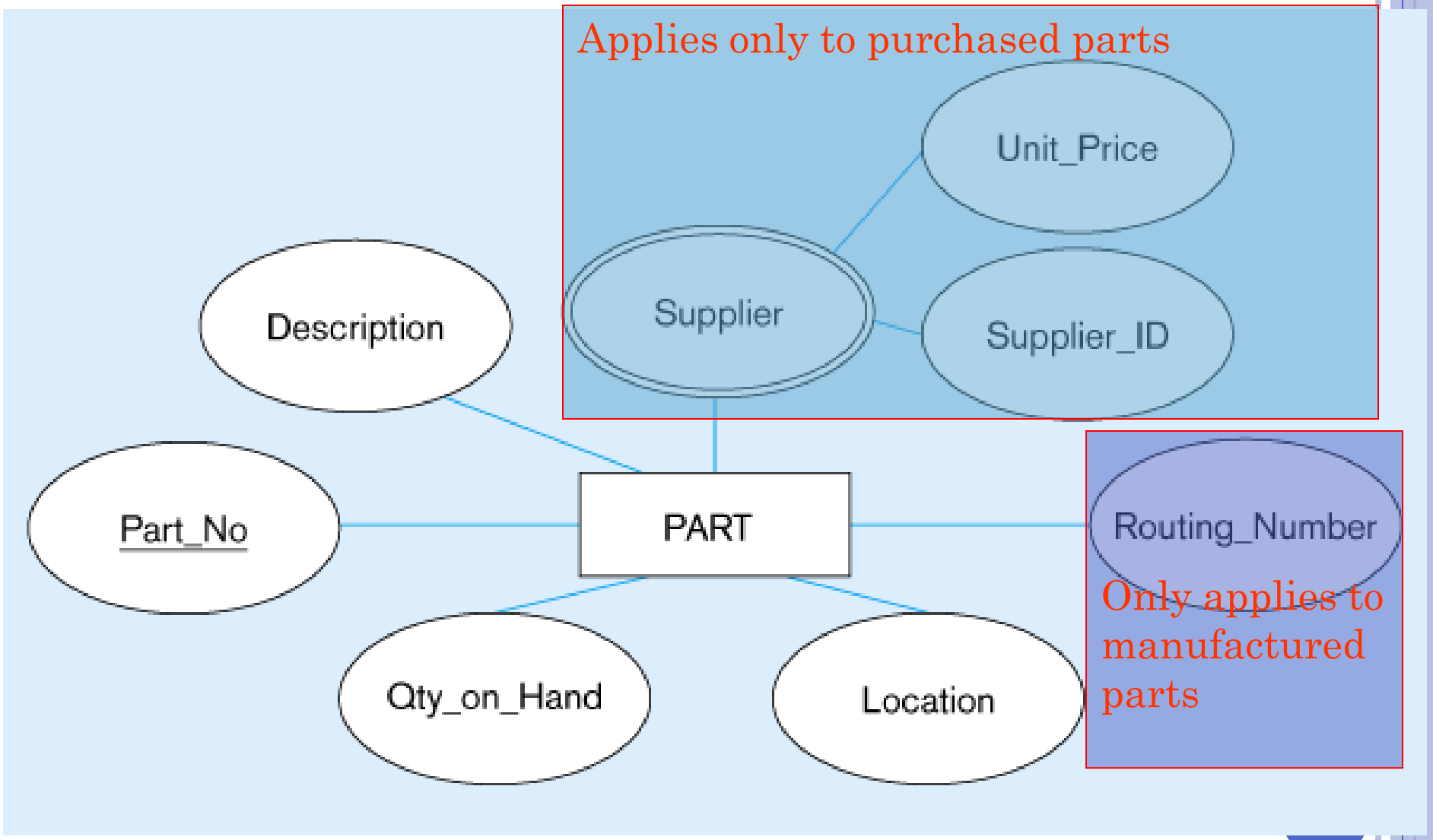
An entity type PART has attributes

- Part_No,
- Description,
- Unit_price,
- Location,
- Quantity_On_Hand,
- Routing_Number (apply only to manufactured parts)
- Supplier (There may be more than one supplier)

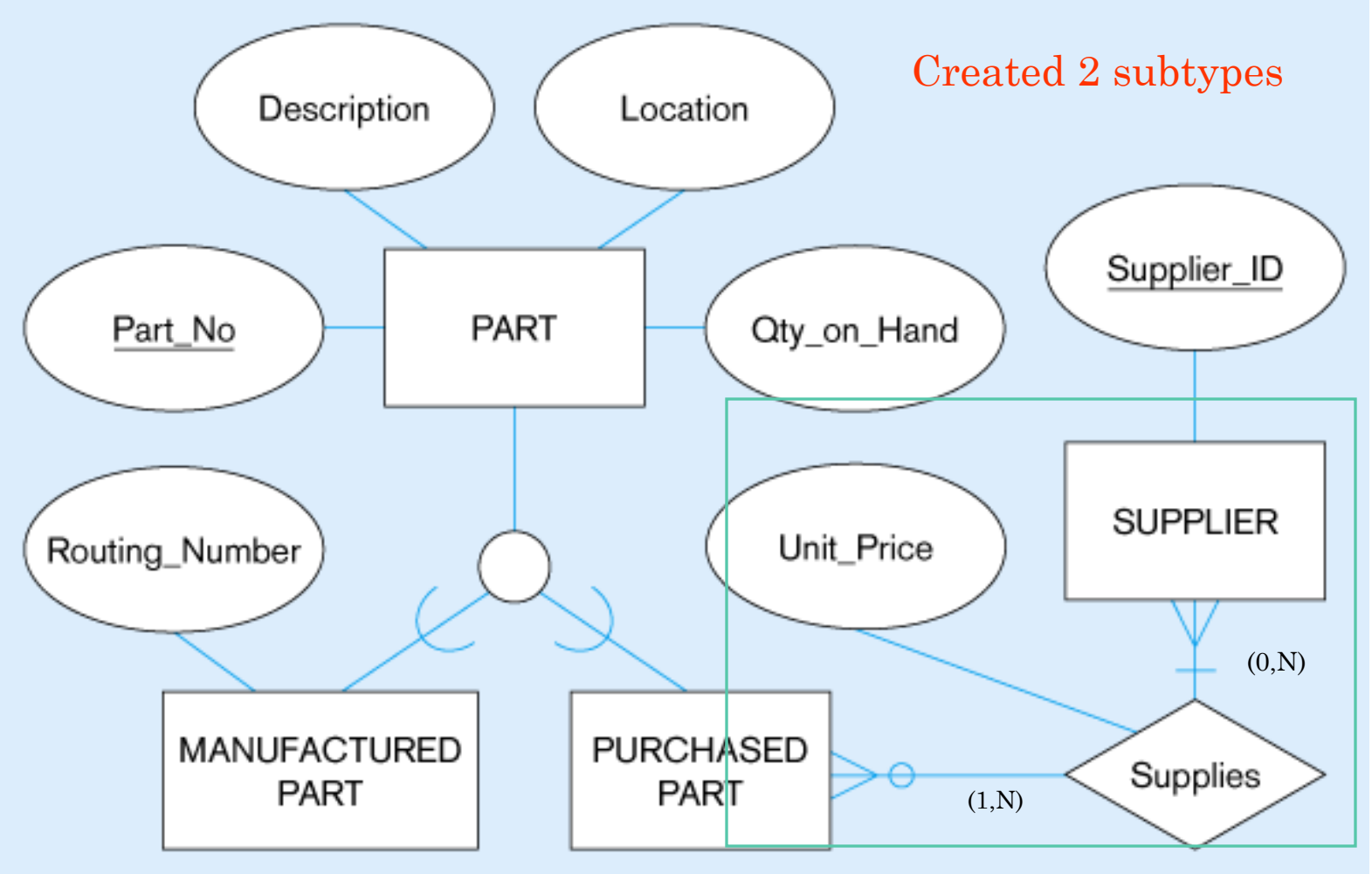
- Some parts are internally Manufactured Parts while others are externally Purchased Parts
- Some parts are obtained from both sources
- The choice depends on factors such as manufacturing capacity, unit price of the parts etc.

Example of Specialization

Entity type PART



Specialization to MANUFACTURED PART and PURCHASED PART

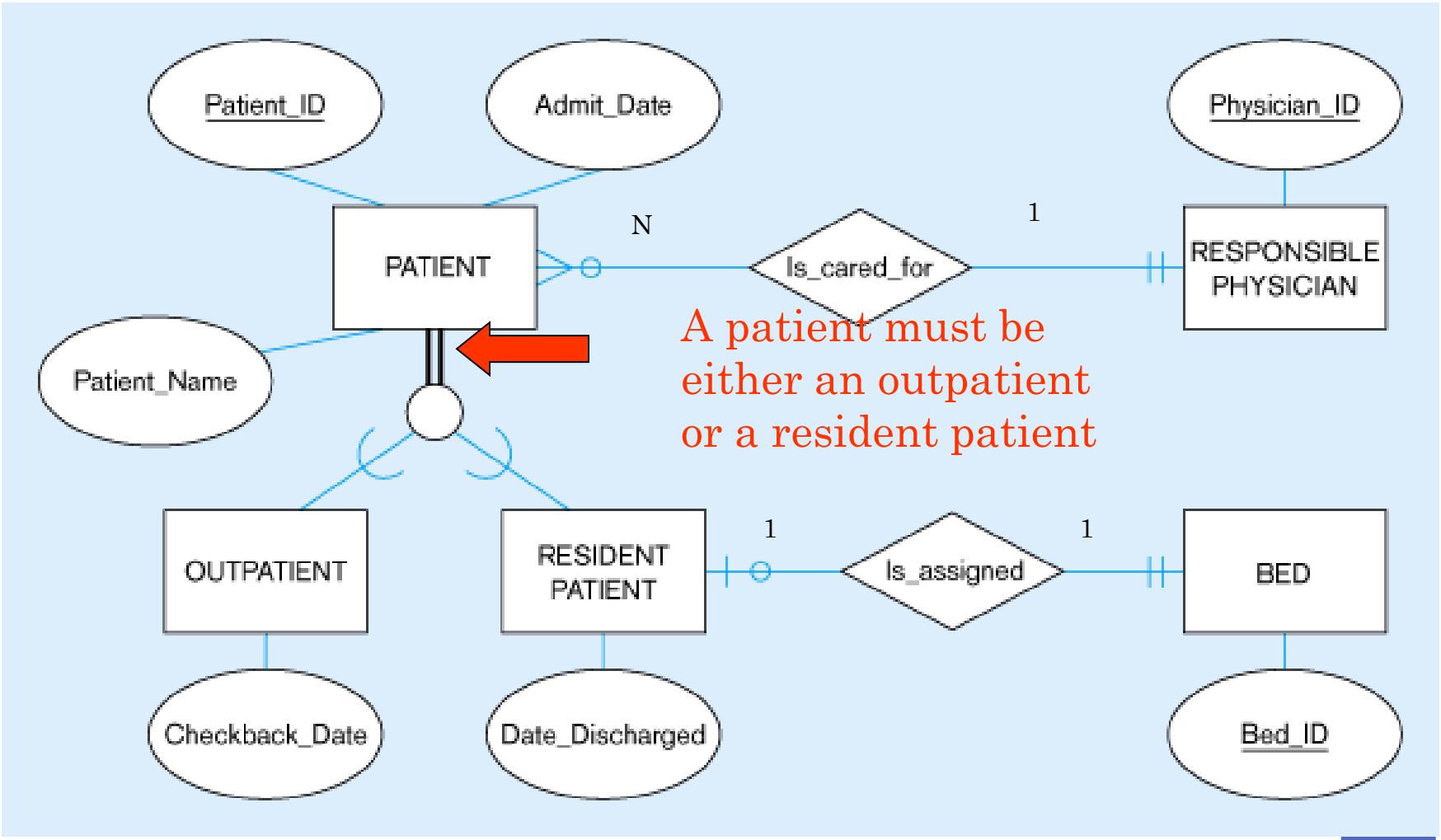


Note: multivalued attribute was replaced by a relationship to another entity

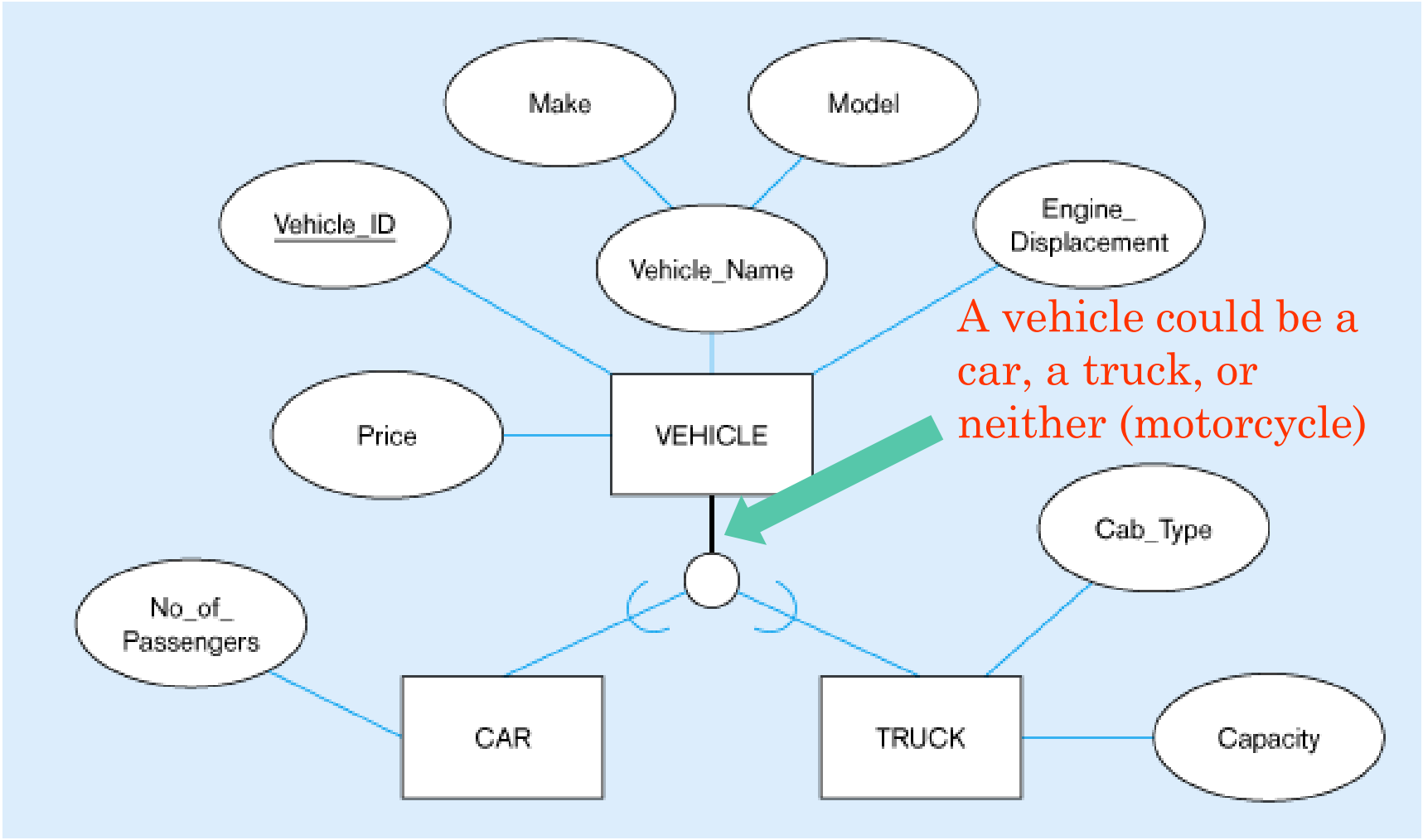
Completeness constraints

- Total Specialization: An entity instance of a supertype ***must*** also be a member of at least one subtype.
- Partial Specialization : An entity instance of the supertype is allowed not to belong to any subtype.

Examples of completeness constraints
Total specialization rule



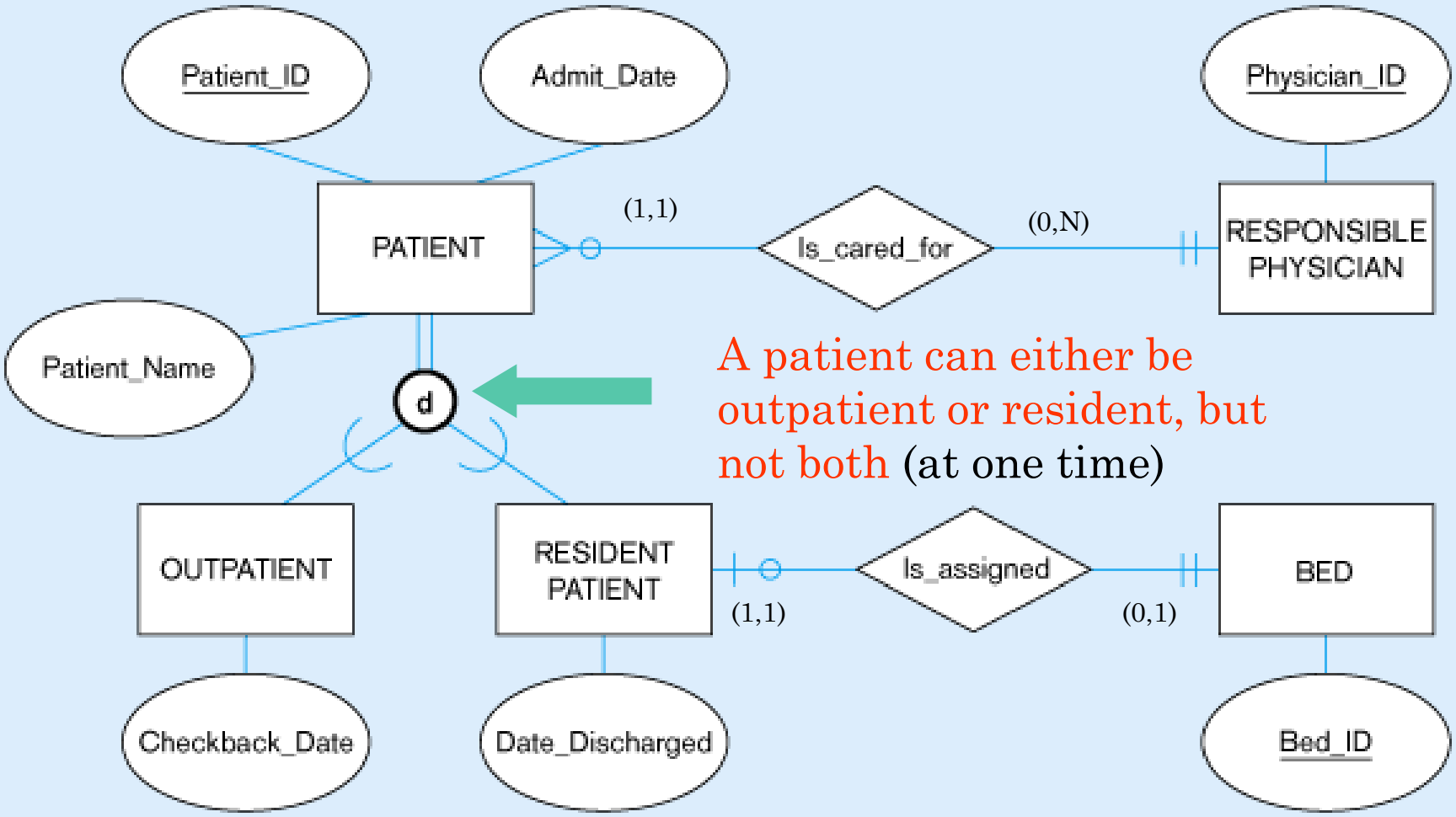
Partial specialization rule



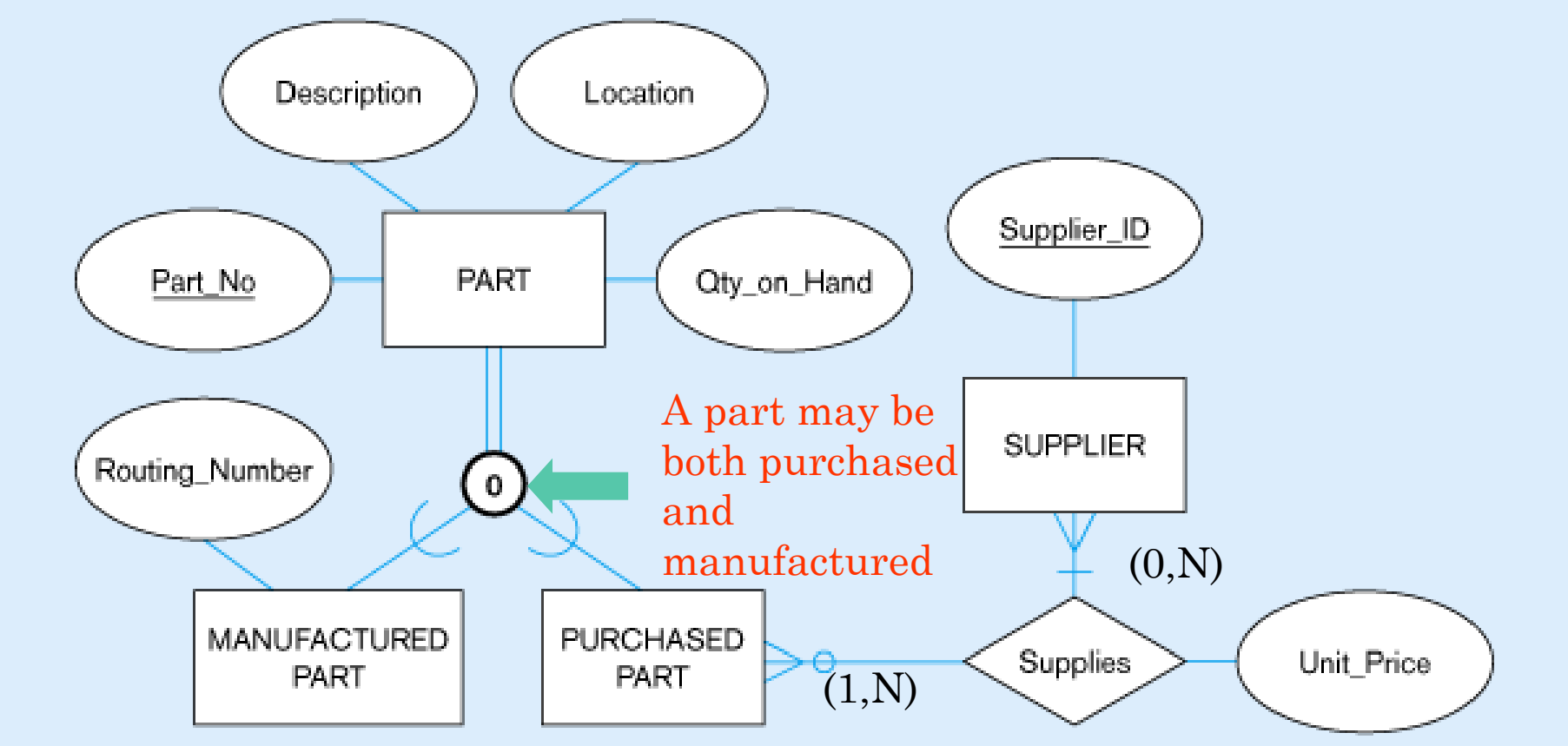
Disjointness constraint

- Can an instance of a supertype may *simultaneously* be a member of two (or more) subtypes?
  - Yes
- We have two possibilities:
  - **Disjoint or**
  - **Overlapping Subtypes**

Examples of disjoint constraints



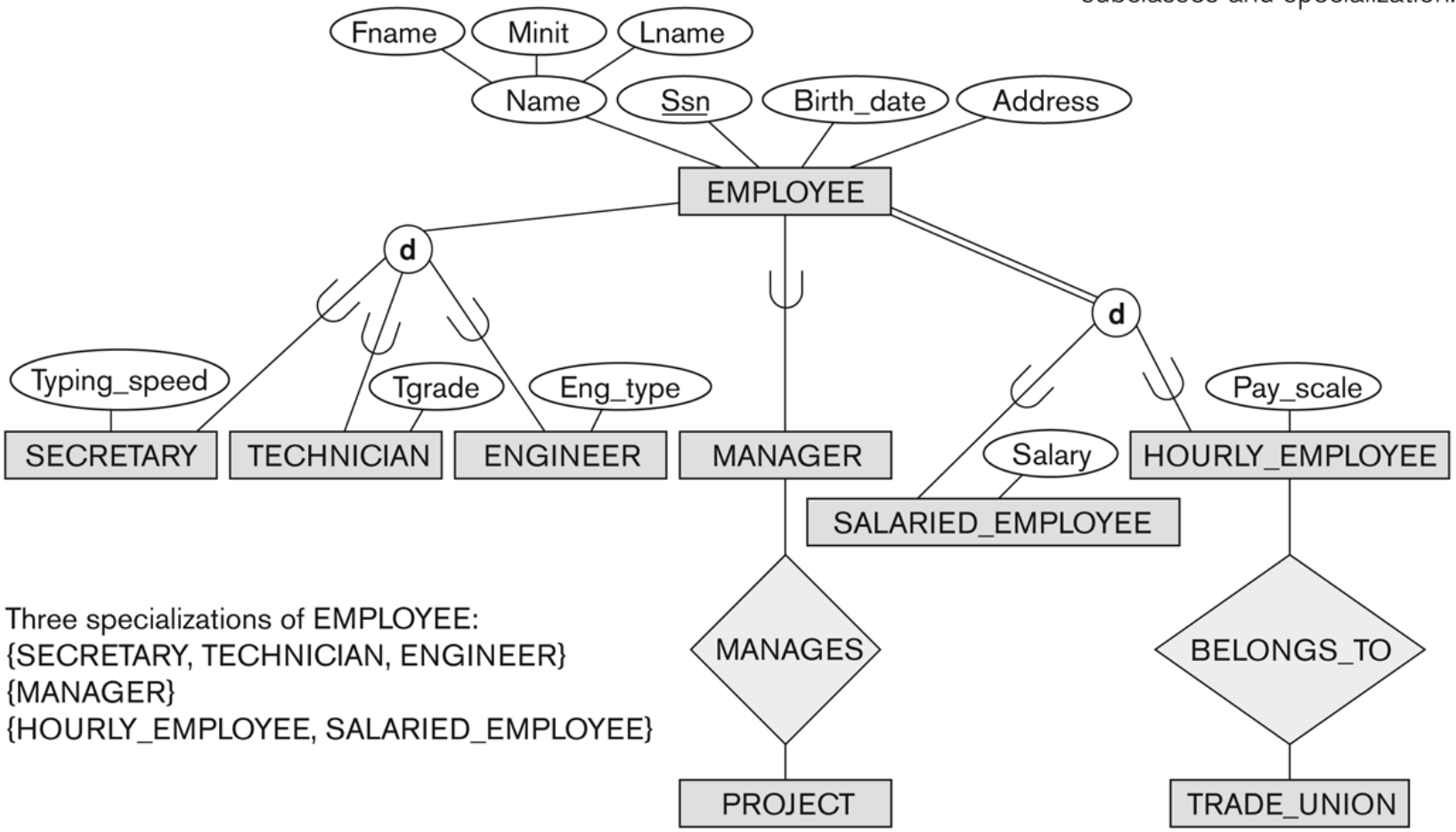
Overlap rule



Double line suggests any part must be either a purchased part or a manufactured part, or it may simultaneously be both of these

Specialization

Figure 4.1
EER diagram notation to represent subclasses and specialization.



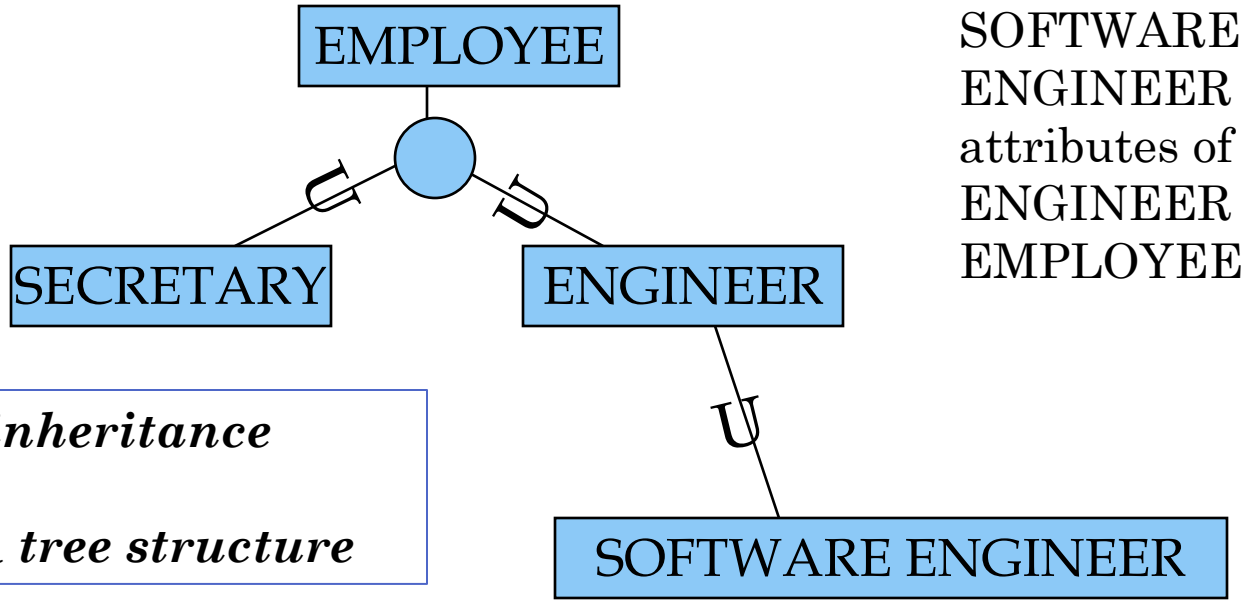
Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

Hierarchies & Lattices

- A subclass may itself have further subclasses specified on it that forms a hierarchy or a lattice
- A subclass inherits attributes of all its predecessor superclasses

HIERARCHIES

- Hierarchy – subclass participates in one class/subclass relationship



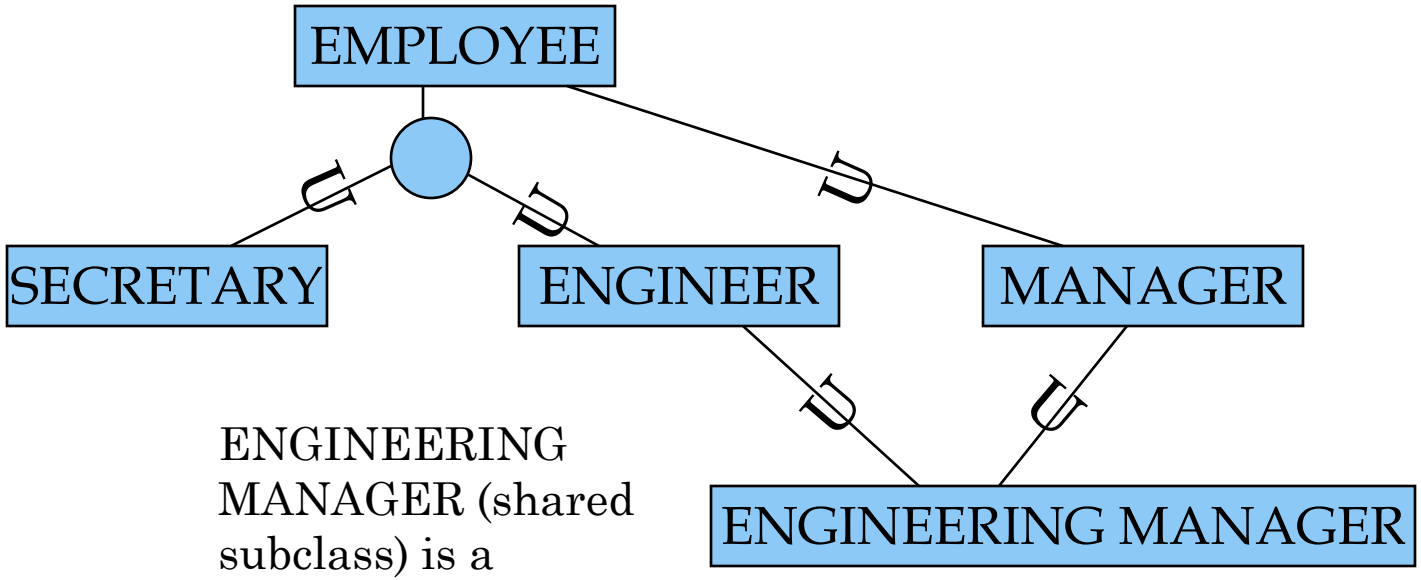
SOFTWARE ENGINEER has all the attributes of an ENGINEER and EMPLOYEE

single inheritance

Forms a ***tree structure***

LATTICES(SHARED SUBCLASS)

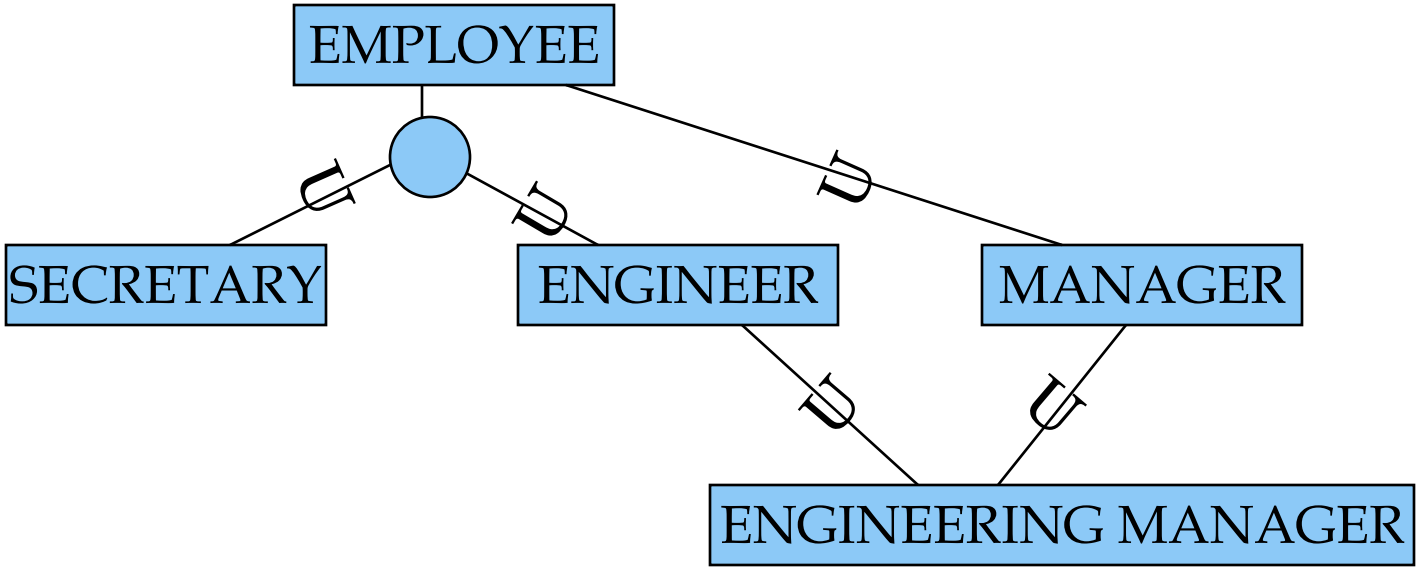
- Lattice – subclass participates in more than one class/subclass relationship (multiple inheritance)



ENGINEERING MANAGER (shared subclass) is a MANAGER and an ENGINEER

SHARED SUBCLASS

- A shared subclass is a subclass in:
  - *more than one* distinct superclass/subclass relationships
  - each relationships has a single superclass
  - shared subclass leads to multiple inheritance



MODEL HUMAN RESOURCES IN A UNIVERSITY

We have three types of resources: EMPLOYEE, STUDENT and ALUMNI. All three types have attributes such as SSN, Name, Address, Gender and BirthDate.

A person may belong to more than one subtype such as ALUMNUS and EMPLOYEE. Alumnus have degrees. And Employee gets salary.

The two major subtypes of Employee are: FACULTY and STAFF. There may be other types of employees. Each staff member have position and faculty member have rank. An employee cannot be both Faculty and Staff at the same time.

There are only two subtypes for student: GRADUATE and UNDERGRADUATE. For Graduate we record test-scores and for Undergrad we record class standing.

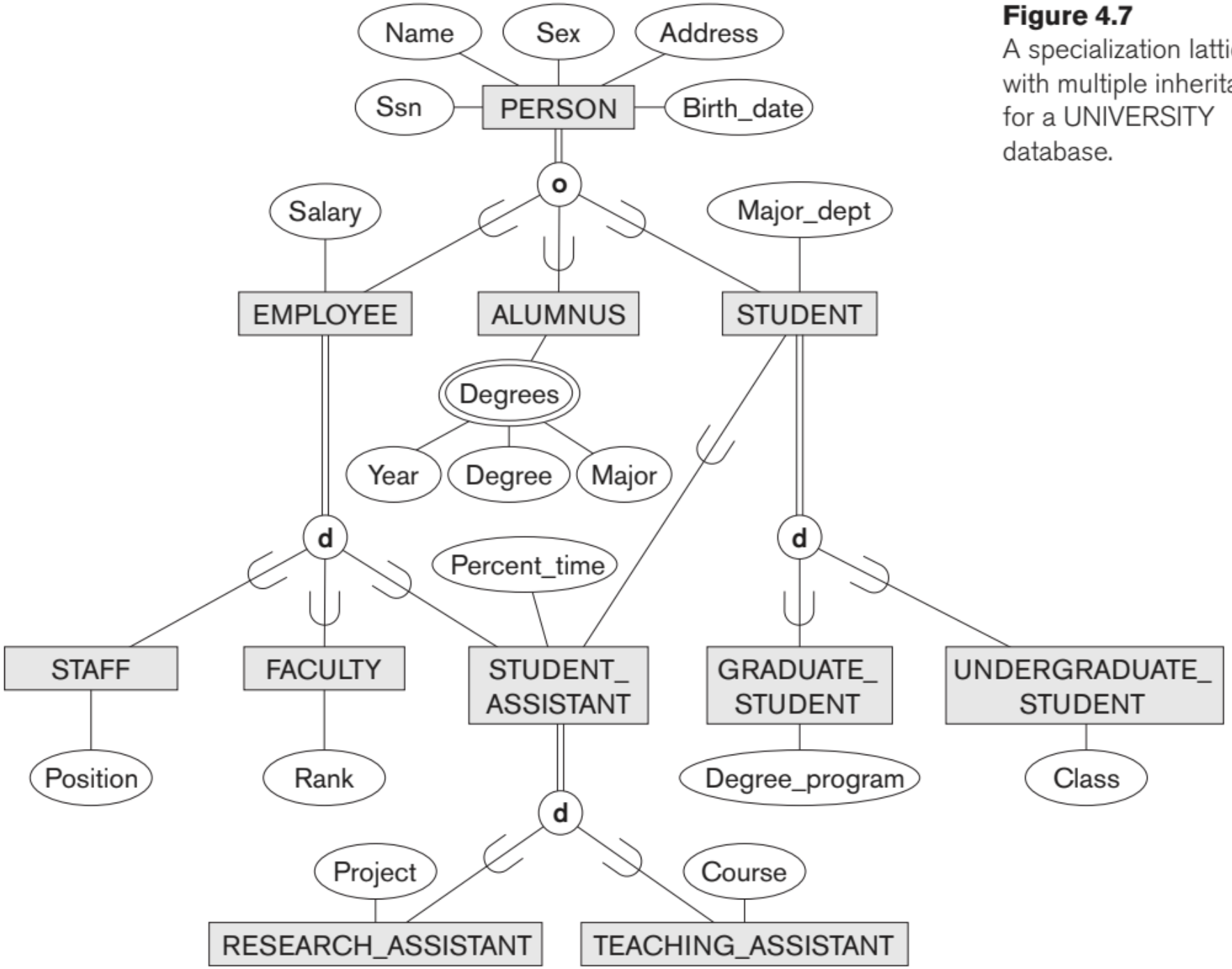
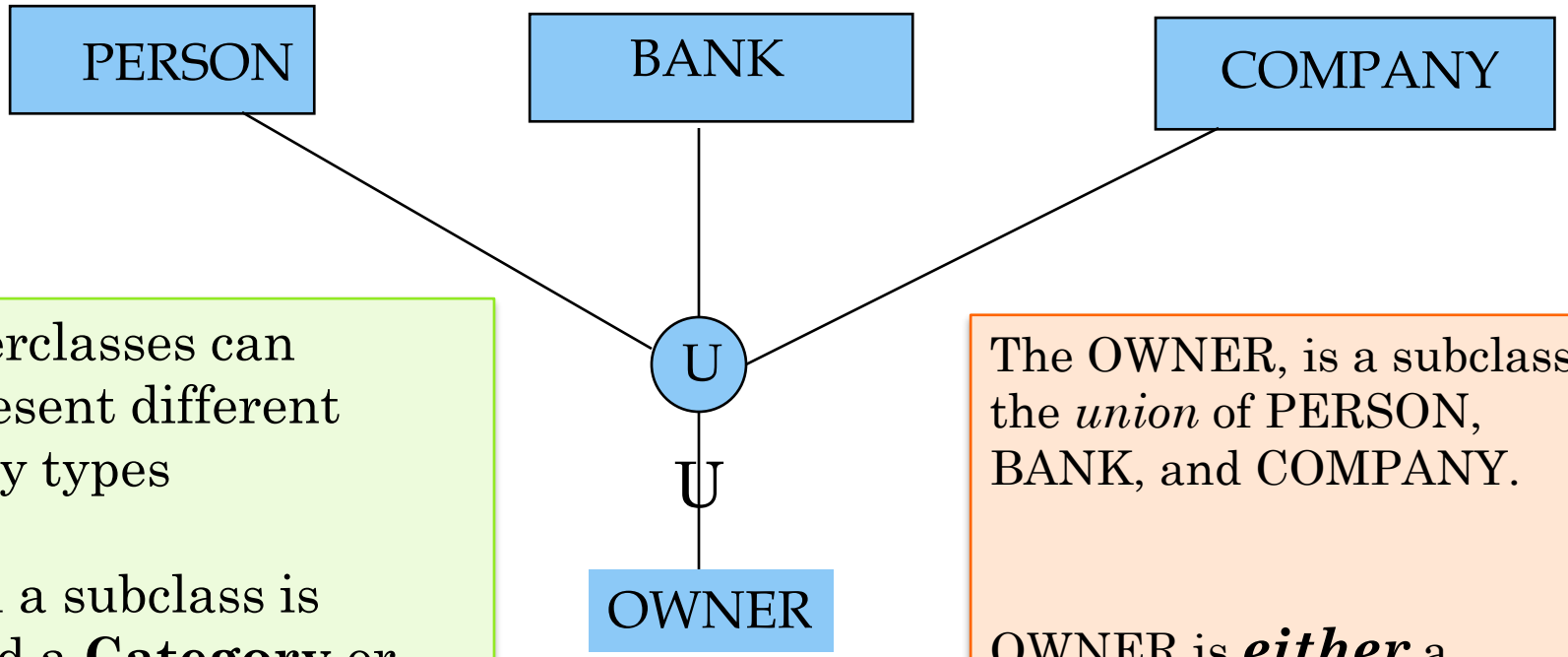


Figure 4.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

CATEGORIES (UNION TYPES)

- Models a class/subclass with ***more than one superclass*** of ***distinct*** entity types.
- Attribute inheritance is selective.



Superclasses can represent different entity types

Such a subclass is called a **Category** or **UNION TYPE**

The OWNER, is a subclass of the *union* of PERSON, BANK, and COMPANY.

OWNER is ***either*** a PERSON or a BANK or a COMPANY

Example
Vehicle registration database

A vehicle owner can be

- a PERSON,
- a BANK (holding a lien on a vehicle) or
- a COMPANY.

Category (union type)

- A vehicle OWNER represents a subset of the *union* of the three superclasses COMPANY, BANK, and PERSON
- A category member must exist in ***at least one*** of its superclasses

Difference from *shared subclass*, which is a:

- subset of the *intersection* of its superclasses
- shared subclass member must exist in ***all*** of its superclasses

VEHICLE REGISTRATION DATABASE

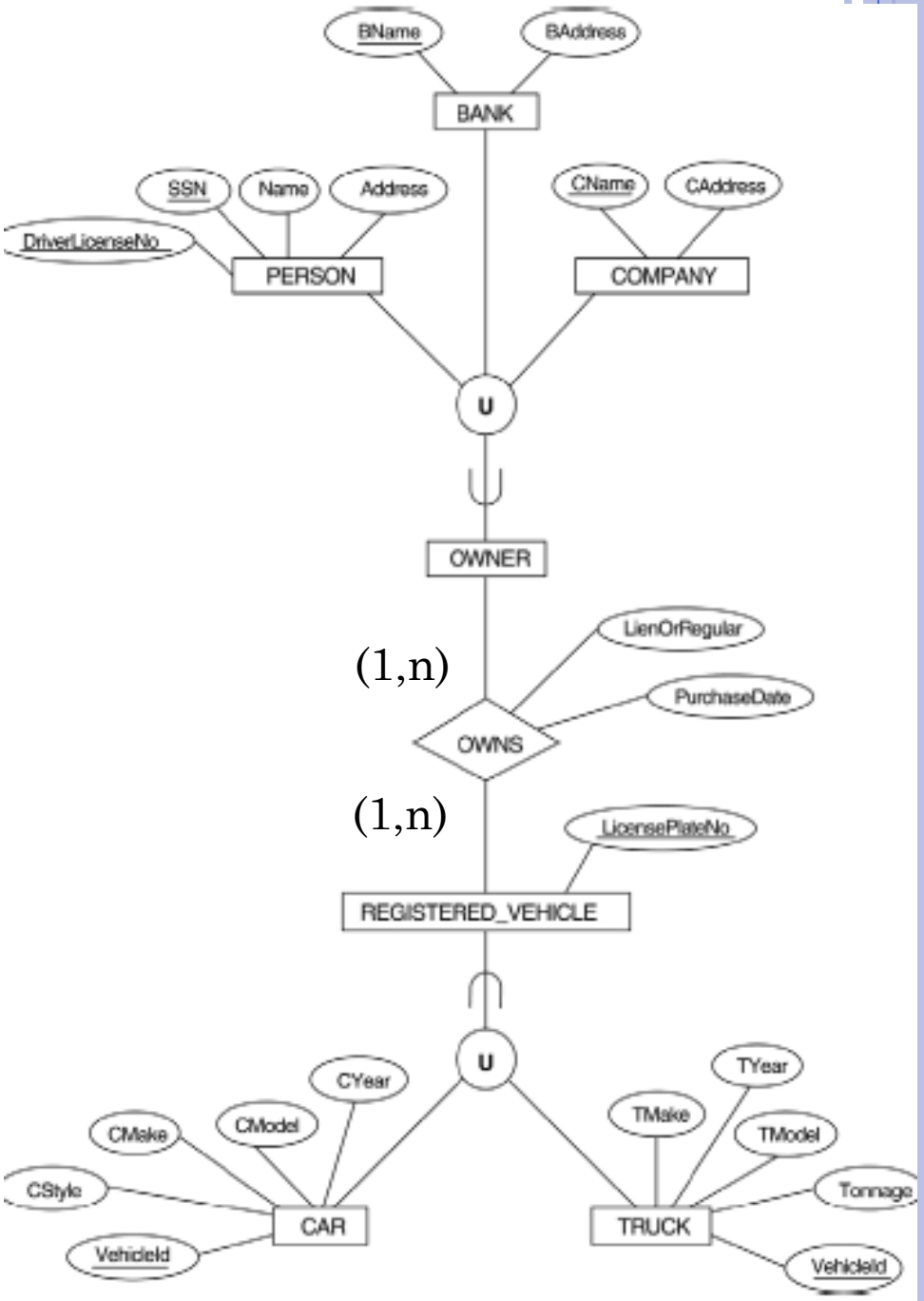
*A category can be **total** or **partial***

Total holds union of all entities in superclass

Partial holds subset of the union

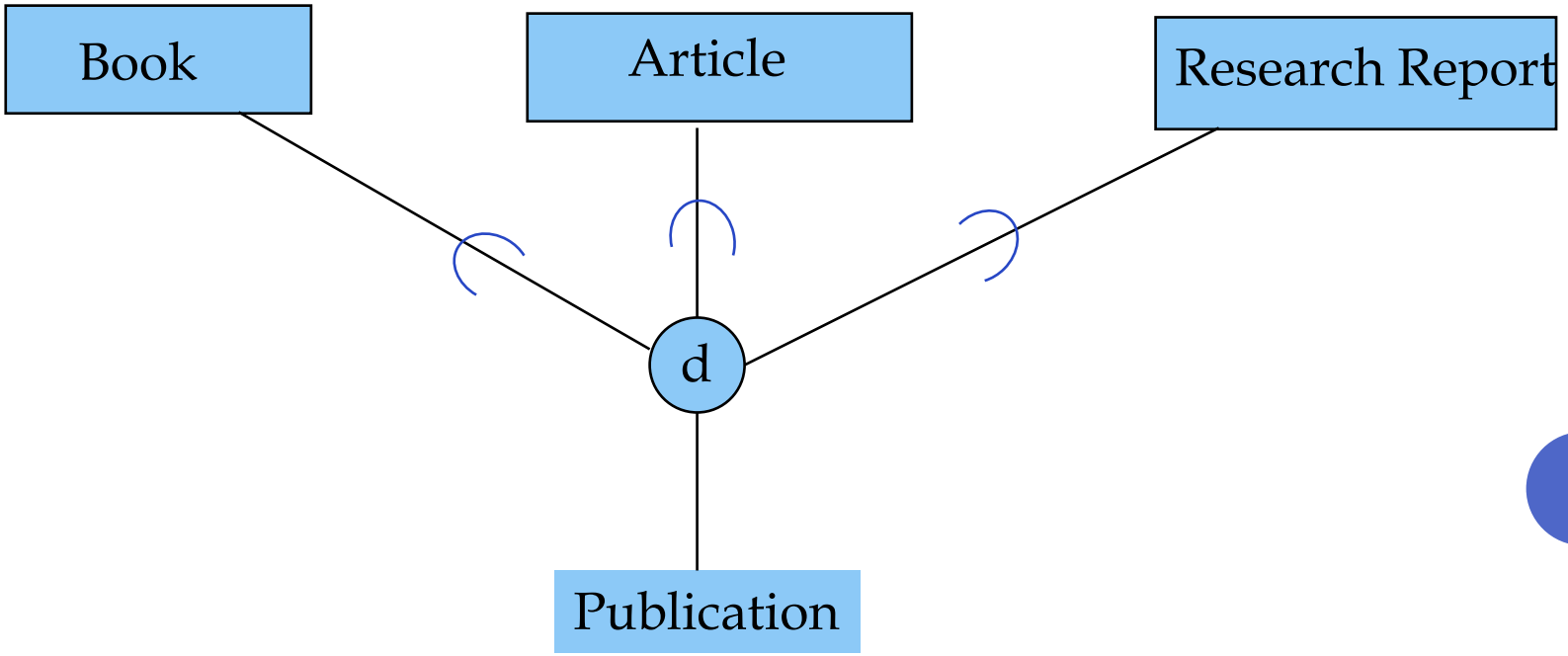
*If category is **total** then it can be represented by **Inheritance***

What is the difference between VEHICLE and REGISTERED_VEHICLE ?



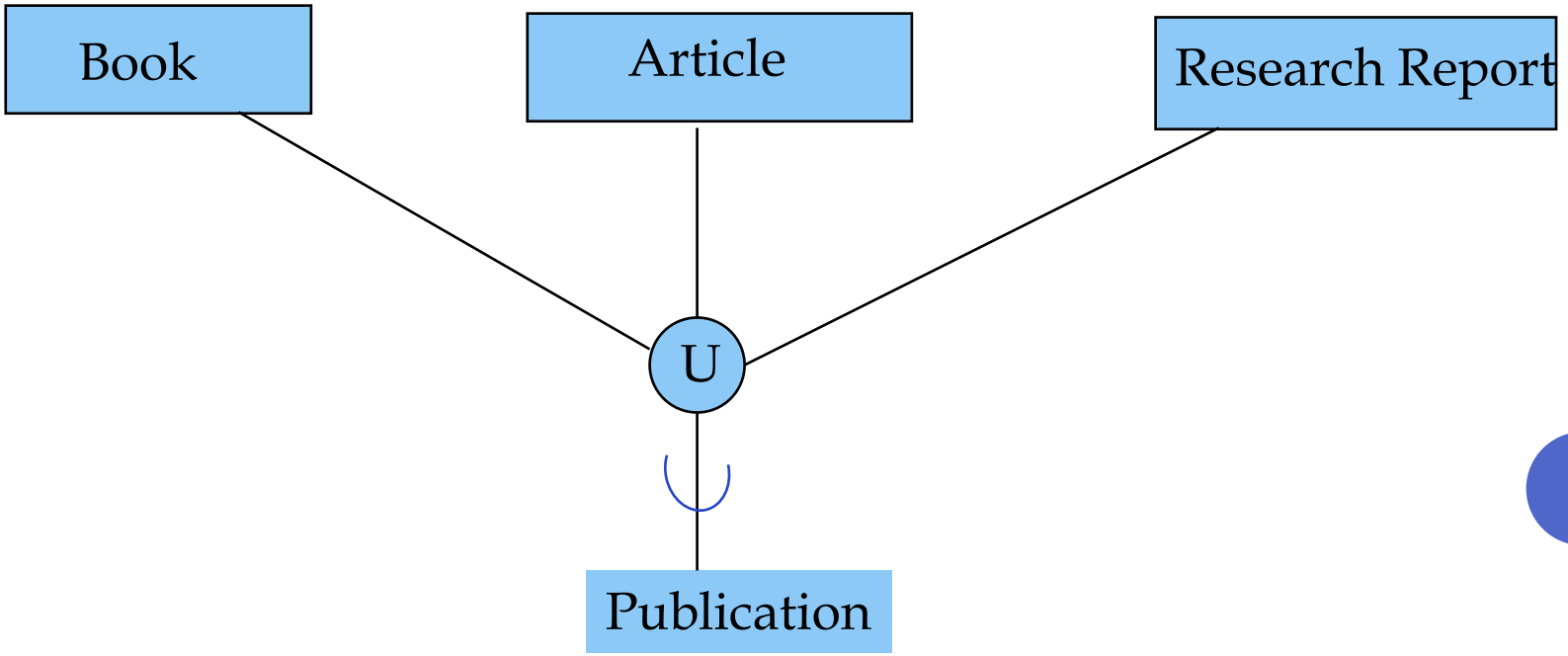
Example: Publication

- We have a list of books, research reports and articles.
- All of them are published by some publisher.
- How to represent ?
  - Inheritance
  - Union Type (category)



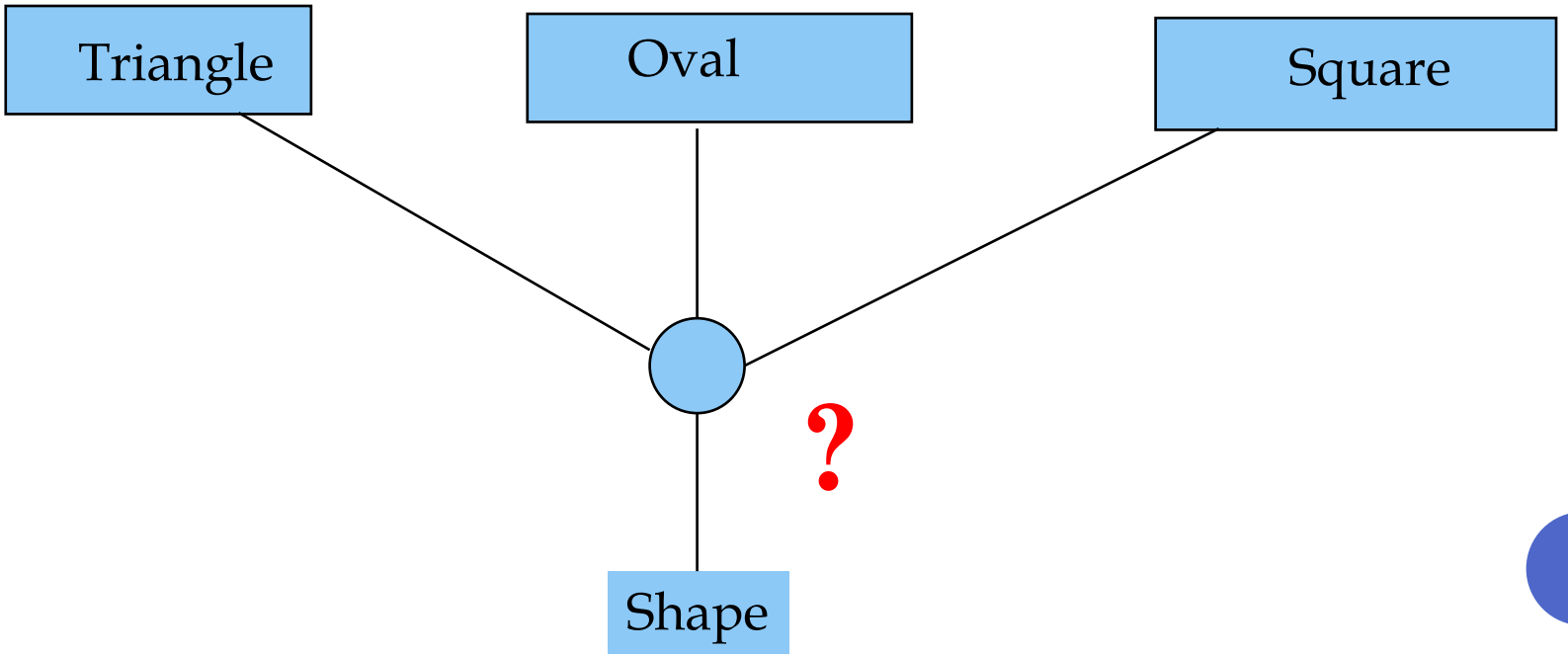
Example 2: Publication

- We have a list of books, research reports and articles.
- Some of them are published while others are still under the process of acceptance.
- How to represent ?
  - Inheritance
  - Union Type (category)



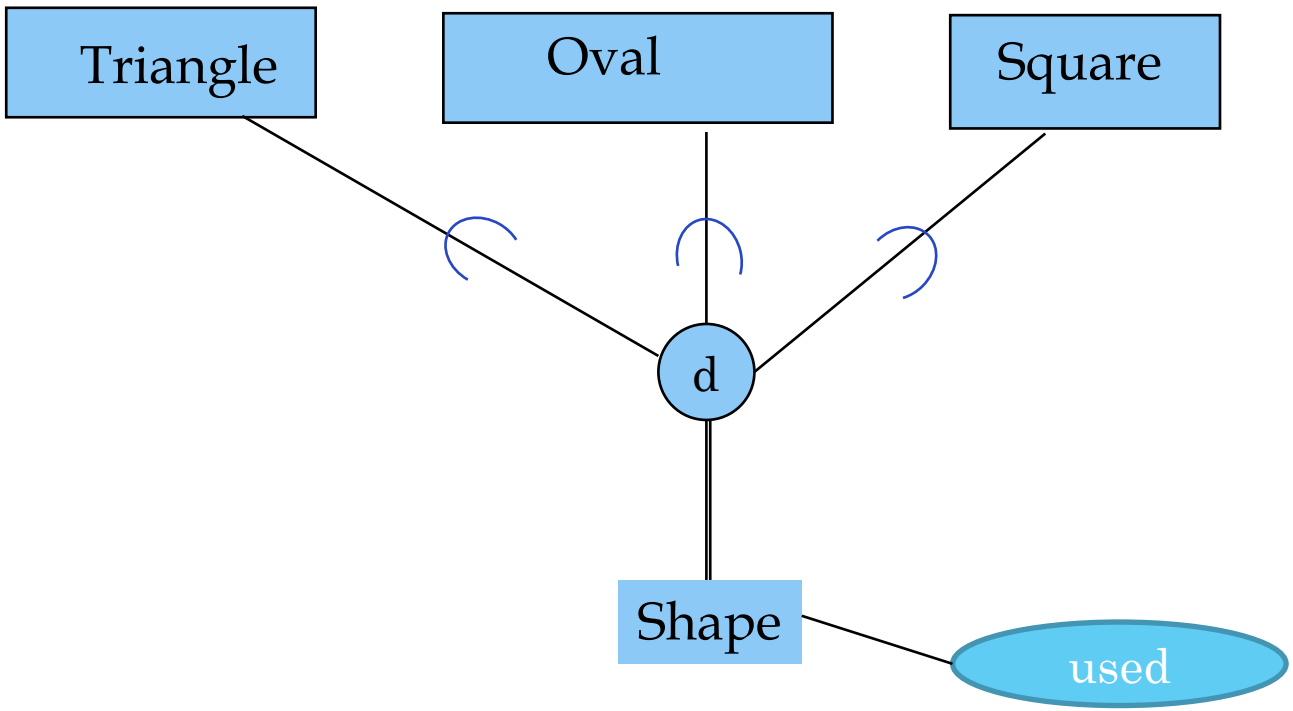
Example 3: Shape

- How to represent ?
  - Inheritance
  - Union Type (category)



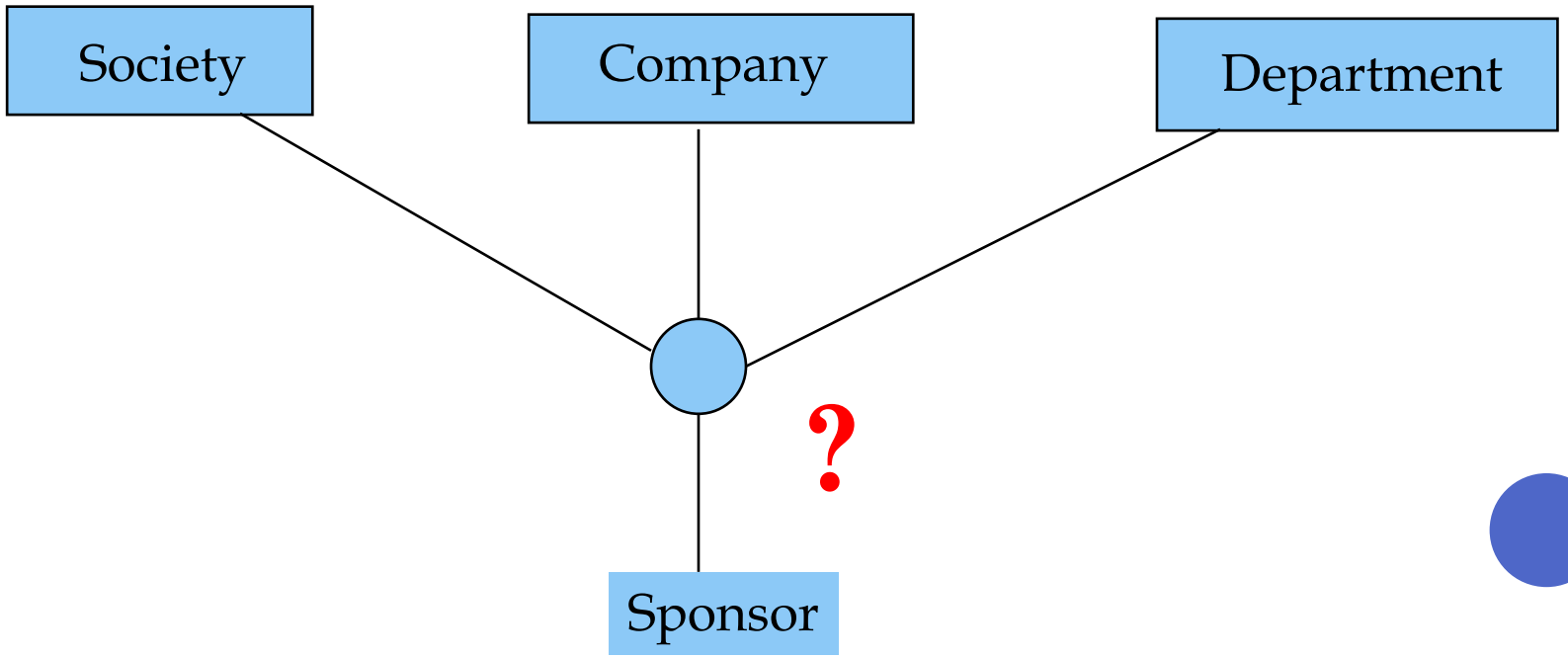
Example 3: Shape

- You are building a puzzle game for kids. A user has to fill a given pattern with some basic shapes.
- How to represent ?
  - Inheritance
  - Union Type (category)



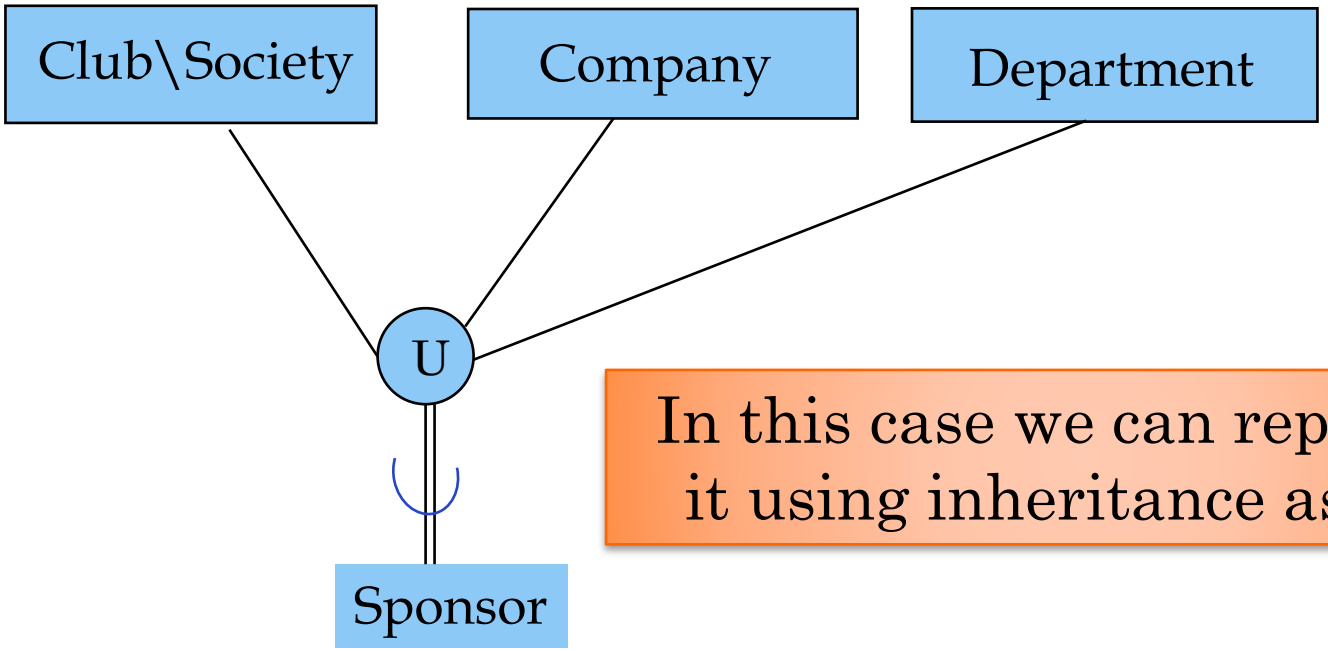
Example 4: Softec Sponsers

- We keep track of the companies, societies and departments around the Pakistan who can sponsor our events at NU-FAST
- How to represent ?
  - Inheritance
  - Union Type (category)



Example 4: Softec Sponsers

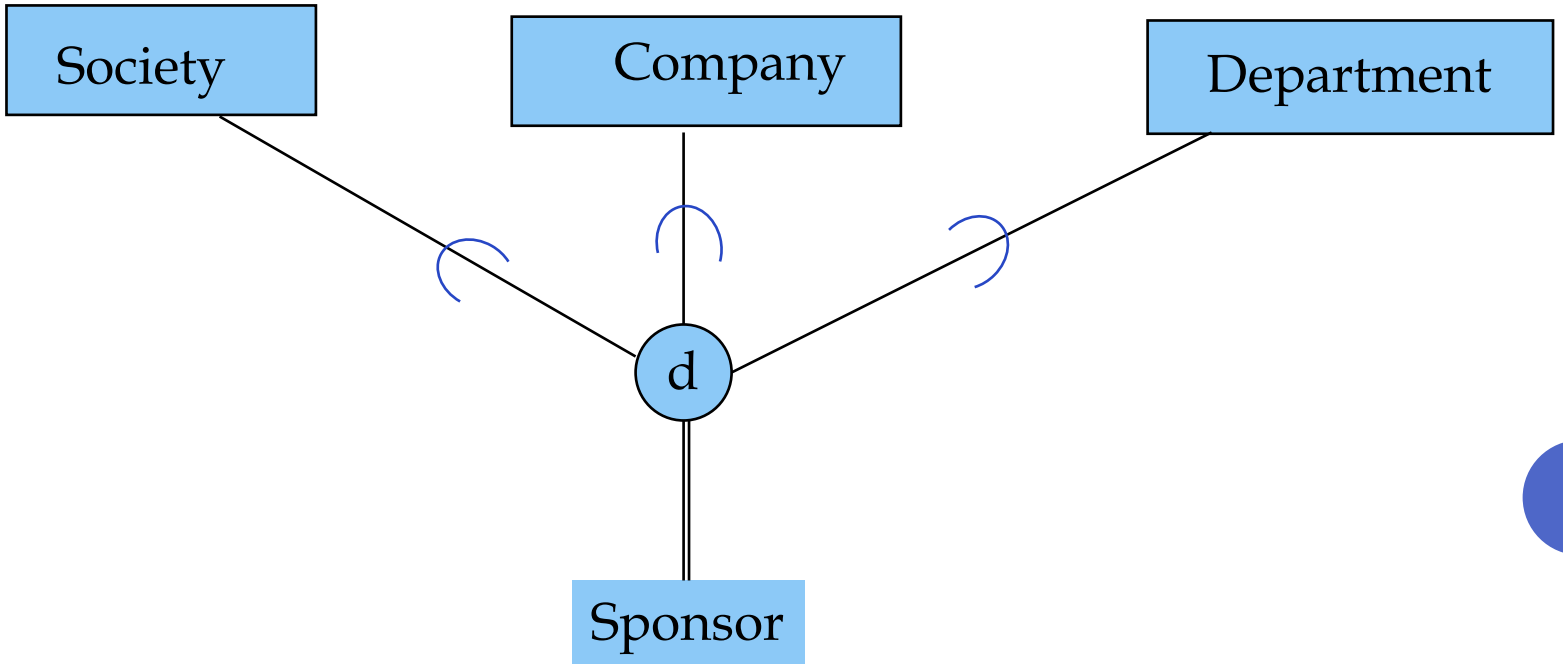
- If we have total union.
  - That means every company, dept and society in our database must be a Sponsor.



In this case we can represent it using inheritance as well

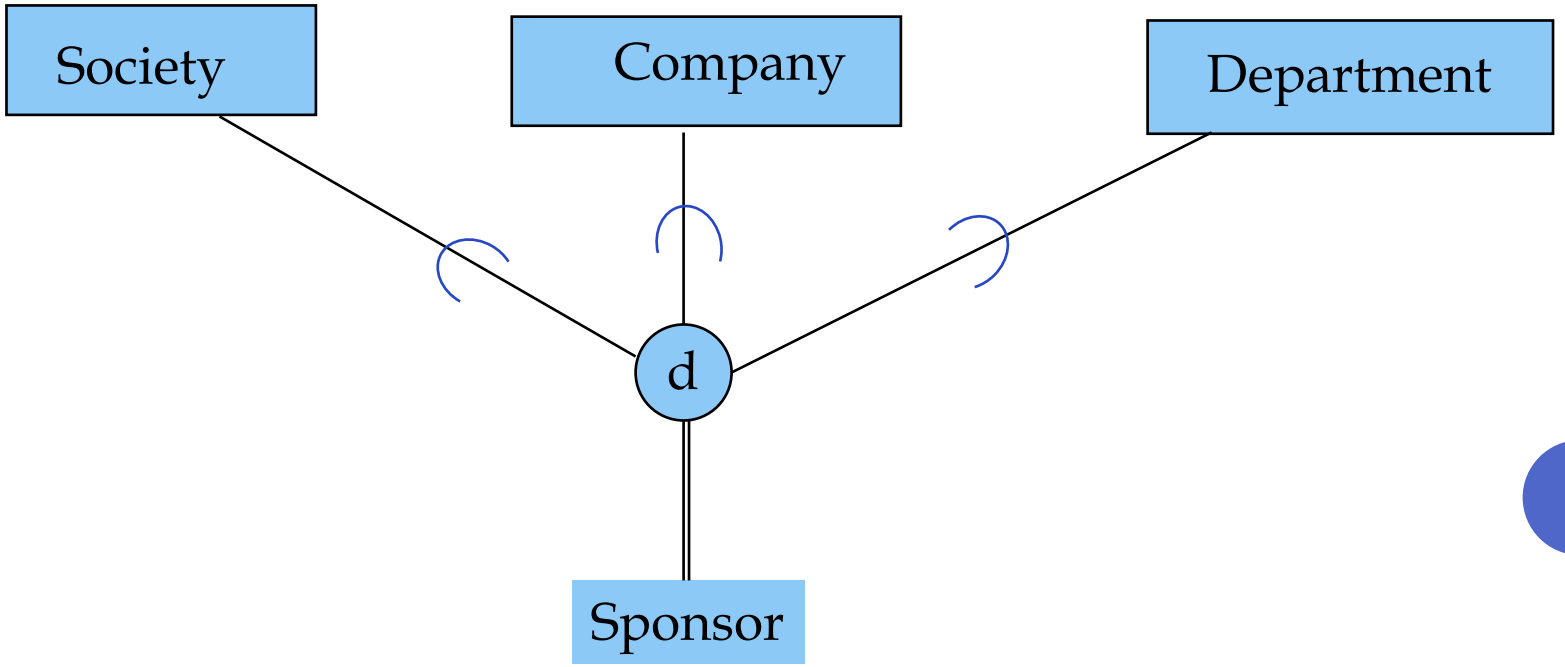
Example 4: Softec Sponsers

- If we have total union.
  - That means every company, dept and society in our database must be a Sponsor.
- In this case we can represent it using inheritance as well

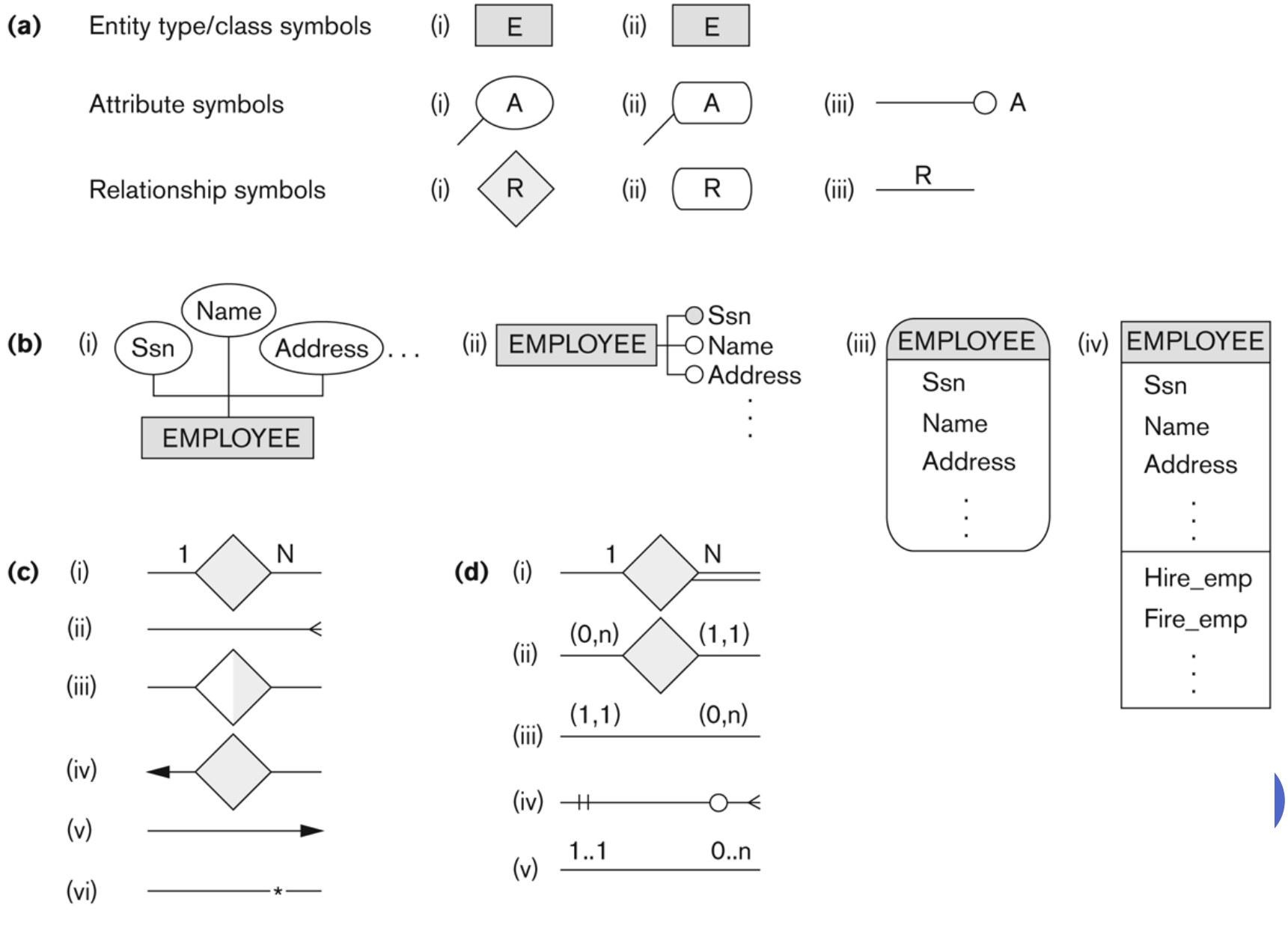


Example 4: Softec Sponsers

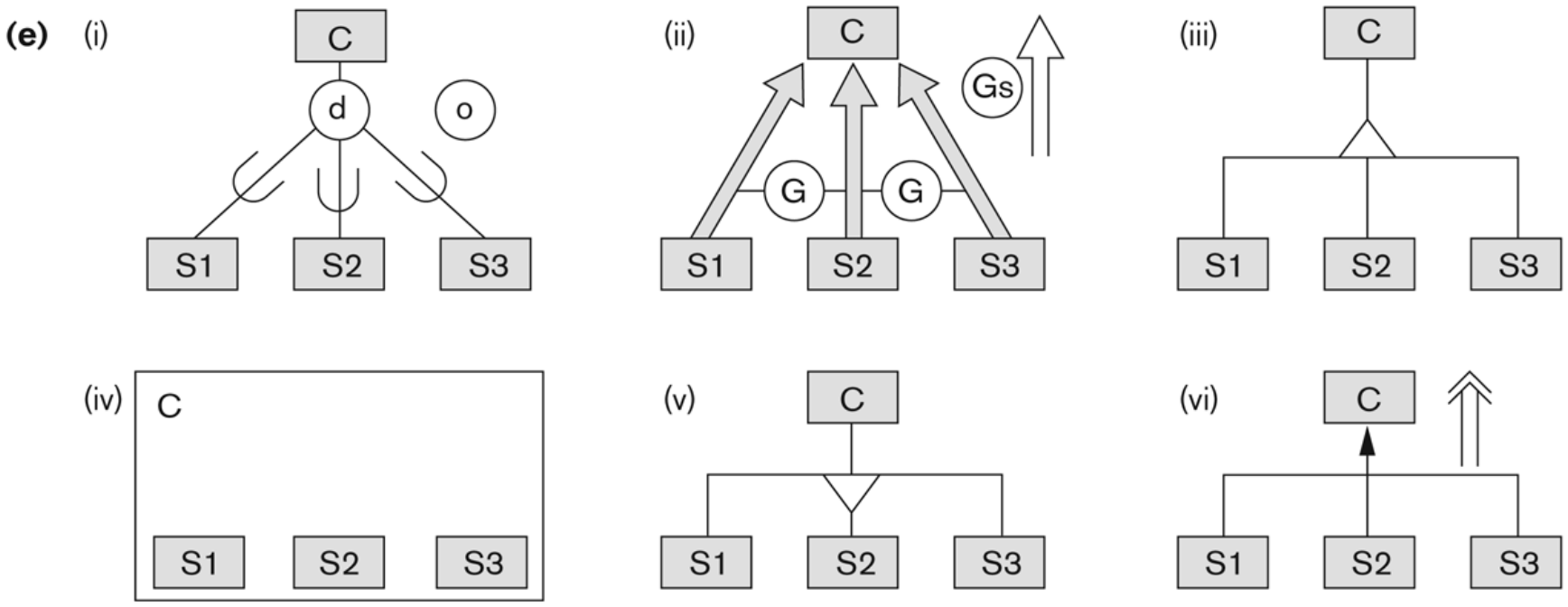
- Which is better ?
  - Union or Inheritance
  - If the Super-classes are very related conceptually then inheritance is better and
  - if Super-classes are very different like person and bank then union is better.



Alternative Diagrammatic Notations



Alternative Diagrammatic Notations



General Conceptual Modeling Concepts

- GENERAL DATA ABSTRACTIONS
  - CLASSIFICATION and INSTANTIATION
  - AGGREGATION and ASSOCIATION (relationships)
  - GENERALIZATION and SPECIALIZATION
  - IDENTIFICATION

- CONSTRAINTS
  - CARDINALITY (Min and Max)
  - COVERAGE (Total vs. Partial, and Exclusive (disjoint) vs. Overlapping)

Summary

- Introduced the EER model concepts
  - Class/subclass relationships
  - Specialization and generalization
  - Inheritance
- These augment the basic ER model concepts introduced in Chapter 3
- EER diagrams and alternative notations were presented