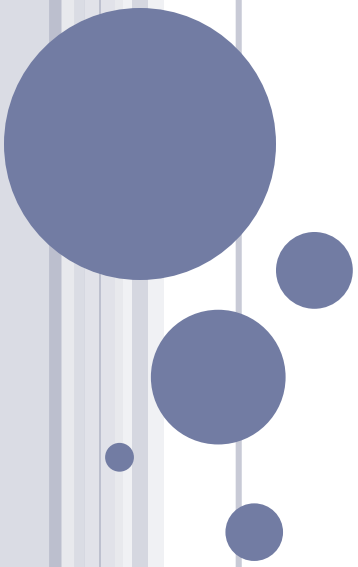
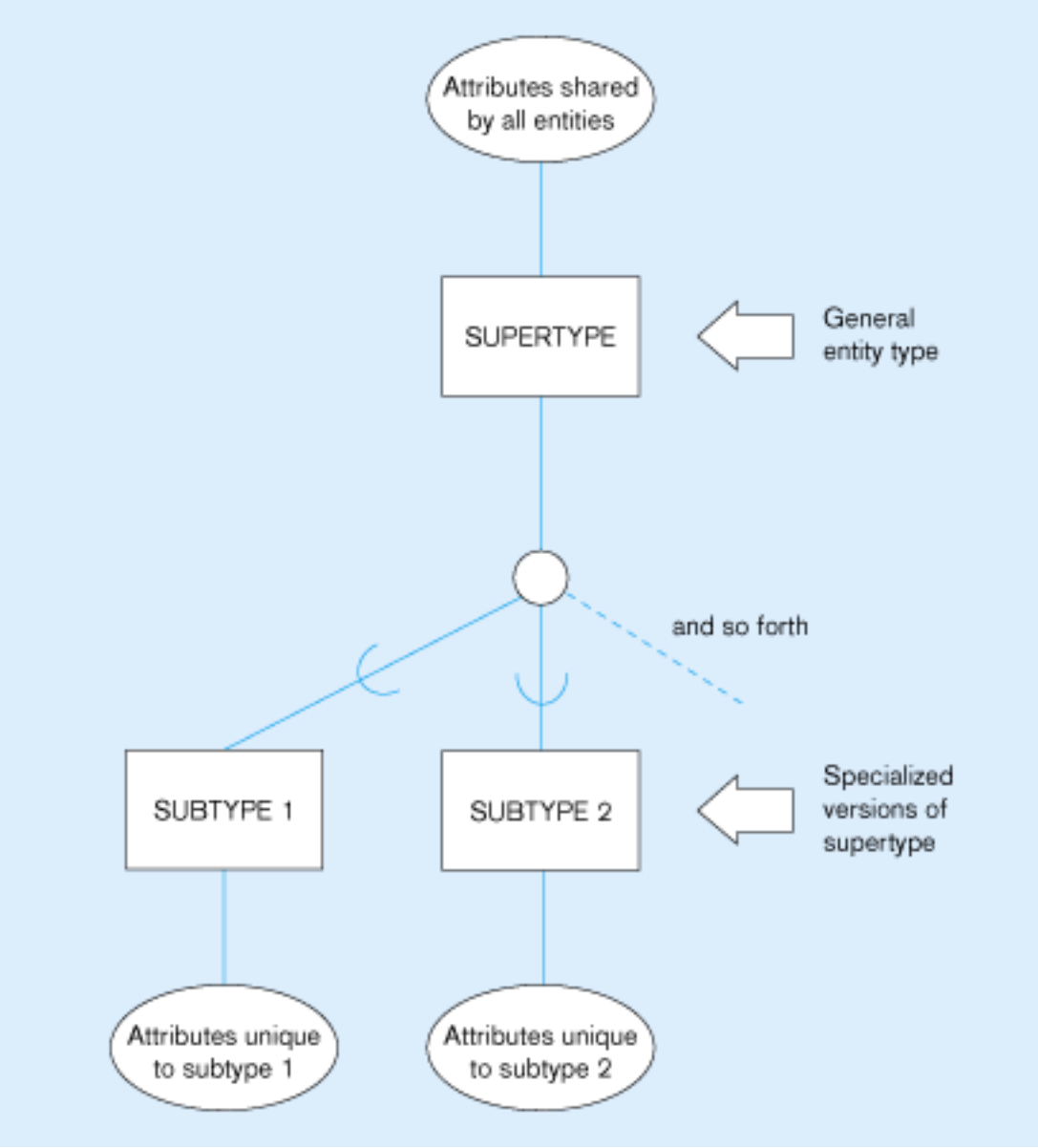


Enhanced Entity-Relationship (EER) Modeling

Introduction

- Enhanced Entity Relationship (EER) model is an extension of the original ER model
- Why do we need EER Model ?
- Which concepts and relationships cannot be captured by ER Model ?

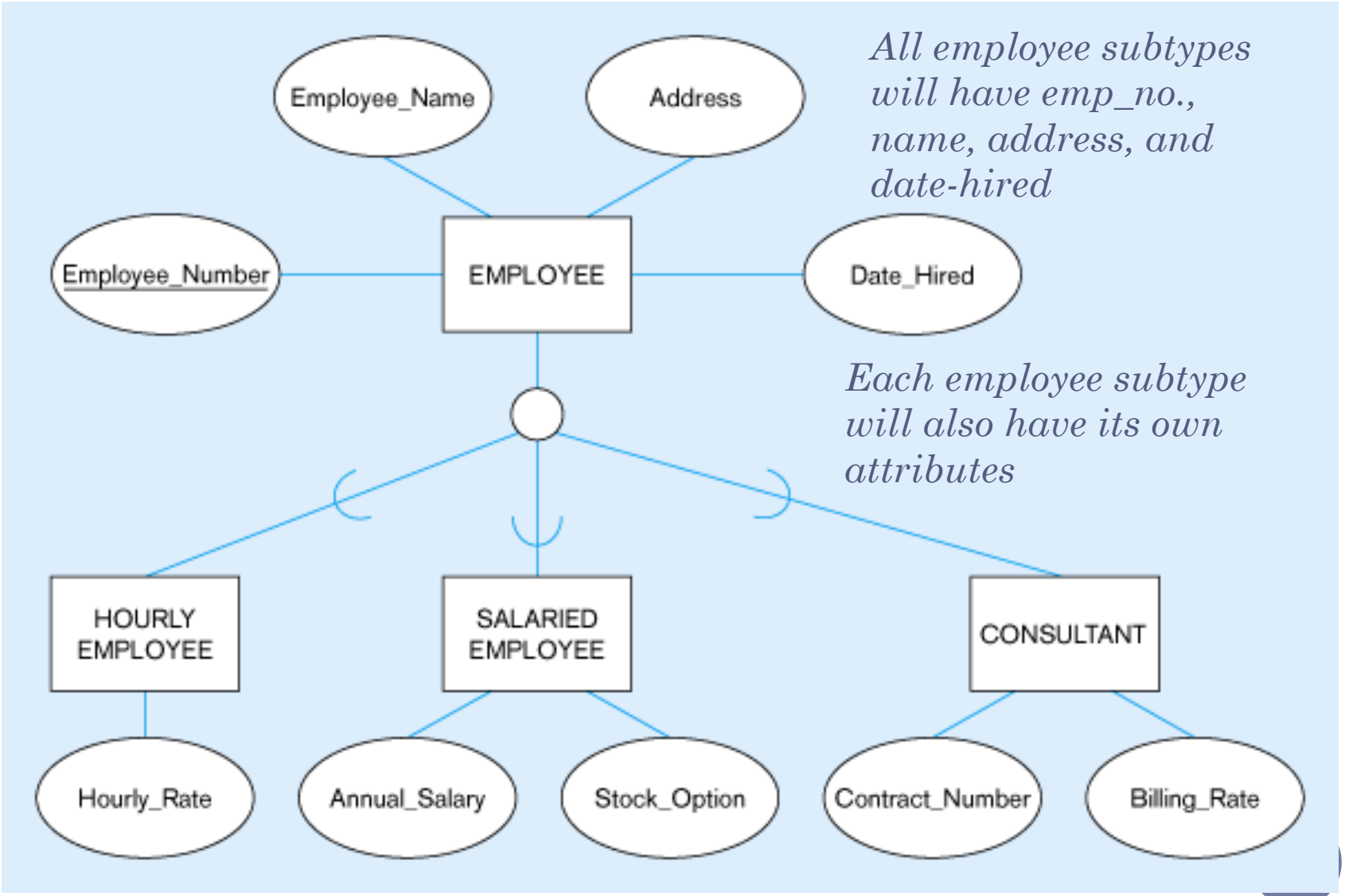
Basic notation for supertype/subtype relationships



An example: EMPLOYEE supertype

- Suppose that an organisation has 3 types of employees:
- **Hourly:**
  - Employee_Number, Employee_Name, Address, Date_Hired, Hourly_Rate
- **Salaried:**
  - Employee_Number, Employee_Name, Address, Date_Hired, Annual_Salary, Stock_Option
- **Contract consultants:**
  - Employee_Number, Employee_Name, Address, Date_Hired, Contract_Number, Billing_Rate

Employee supertype with three subtypes

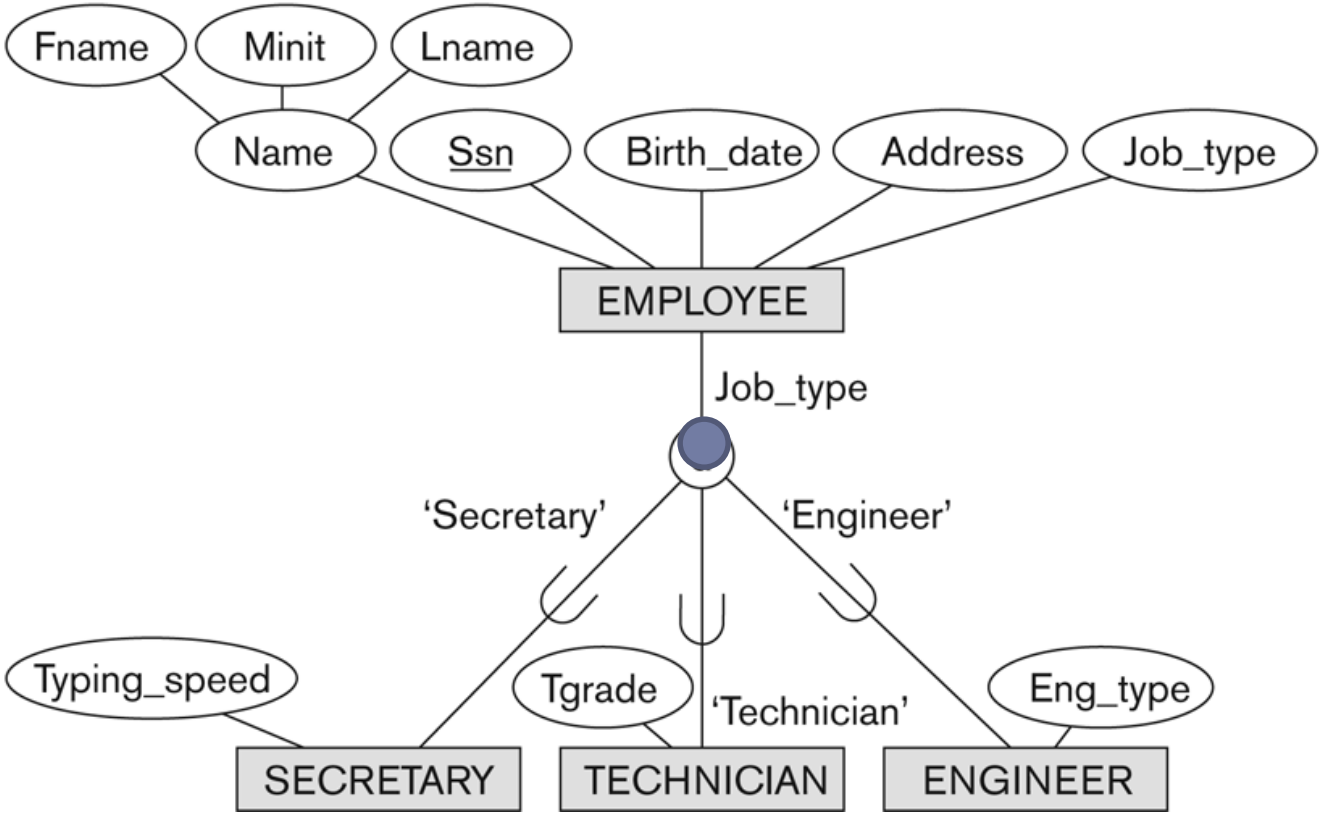


Subclasses and Superclasses

- An entity type may have different sub-groupings
  - Example: EMPLOYEE may be grouped into:
    - SECRETARY, ENGINEER, TECHNICIAN, …
    - MANAGER
    - SALARIED_EMPLOYEE, HOURLY_EMPLOYEE

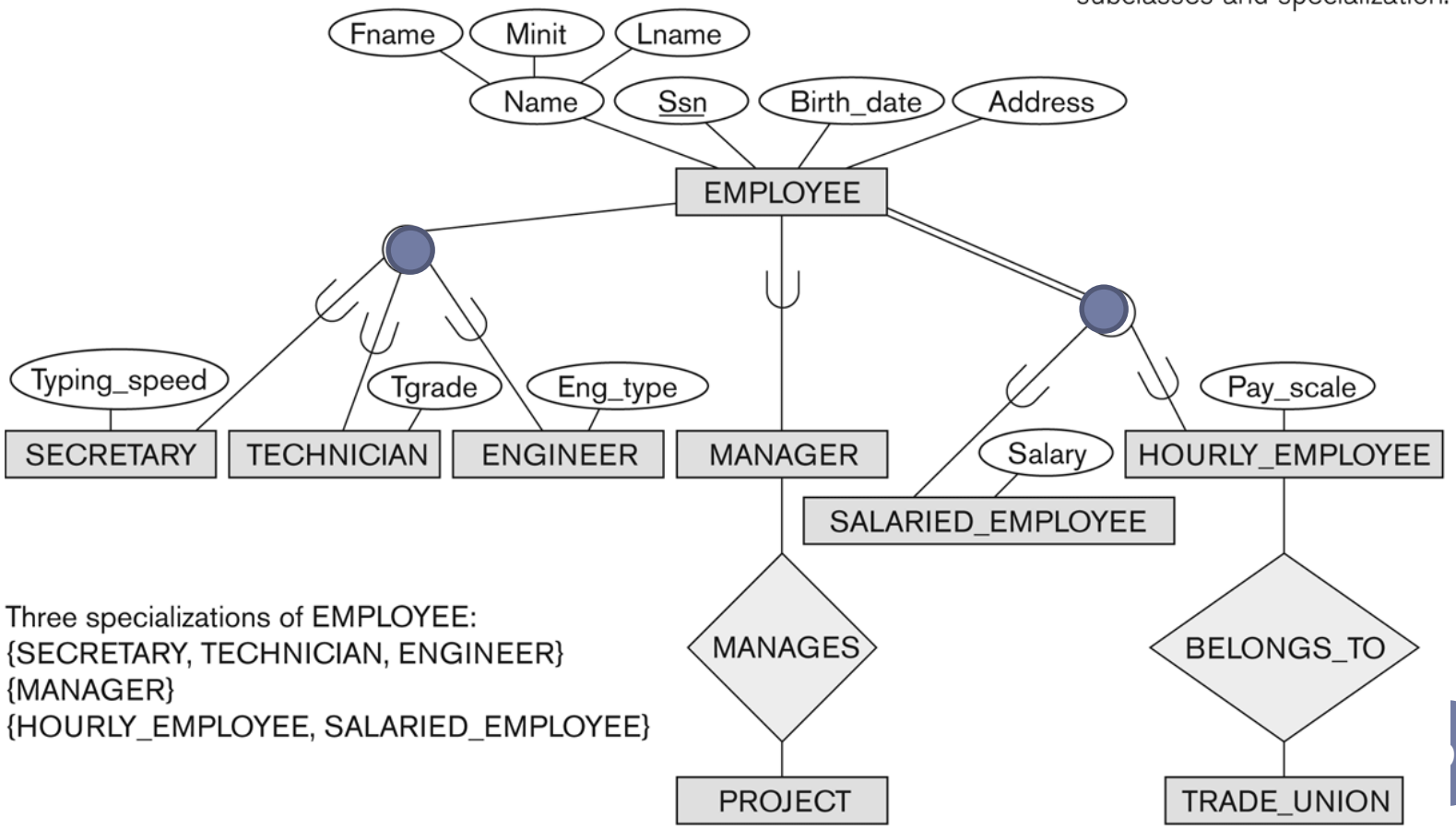
Subclasses and Superclasses

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



Subclasses and Superclasses

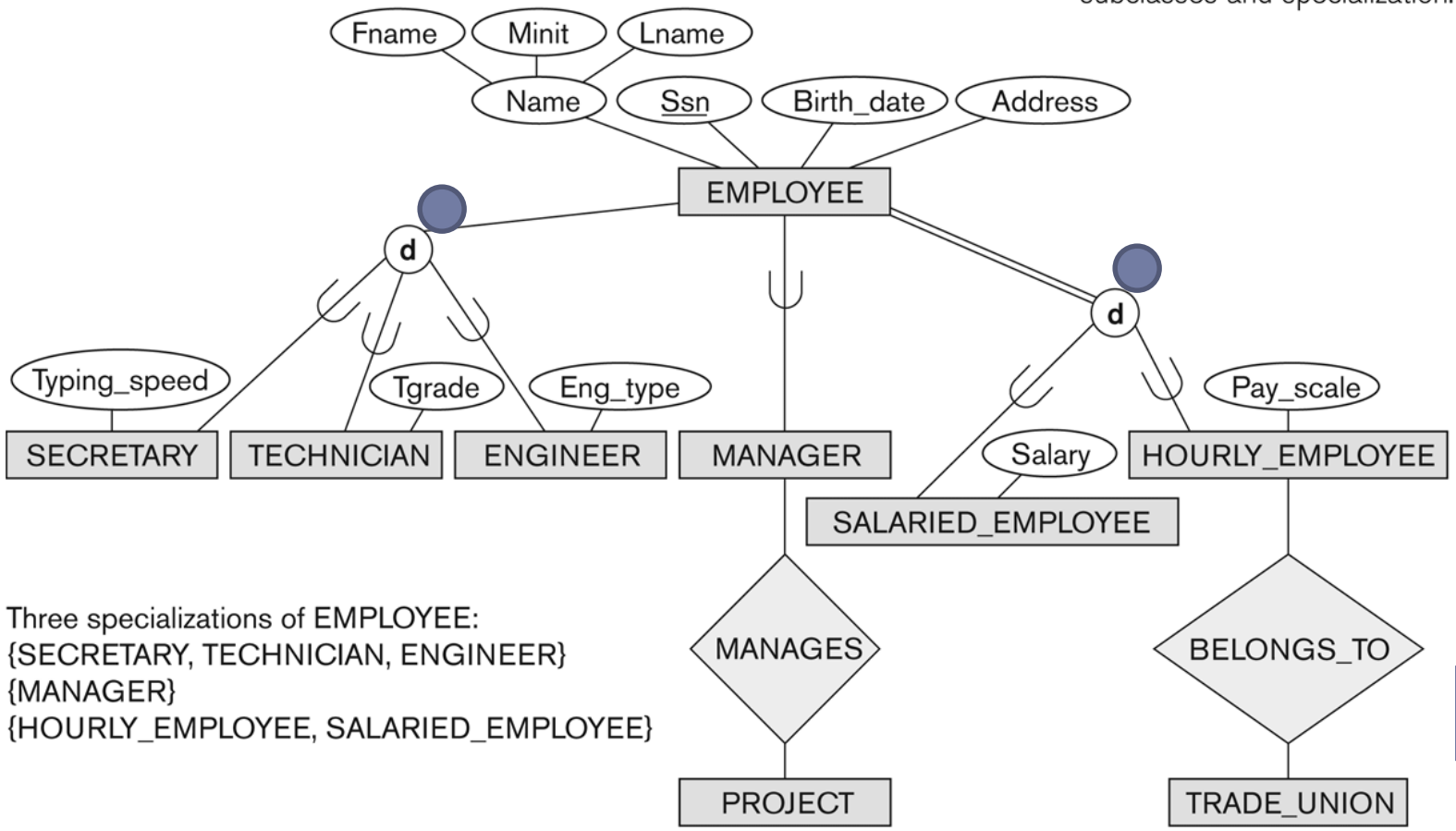
Figure 4.1
EER diagram notation to represent subclasses and specialization.



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.

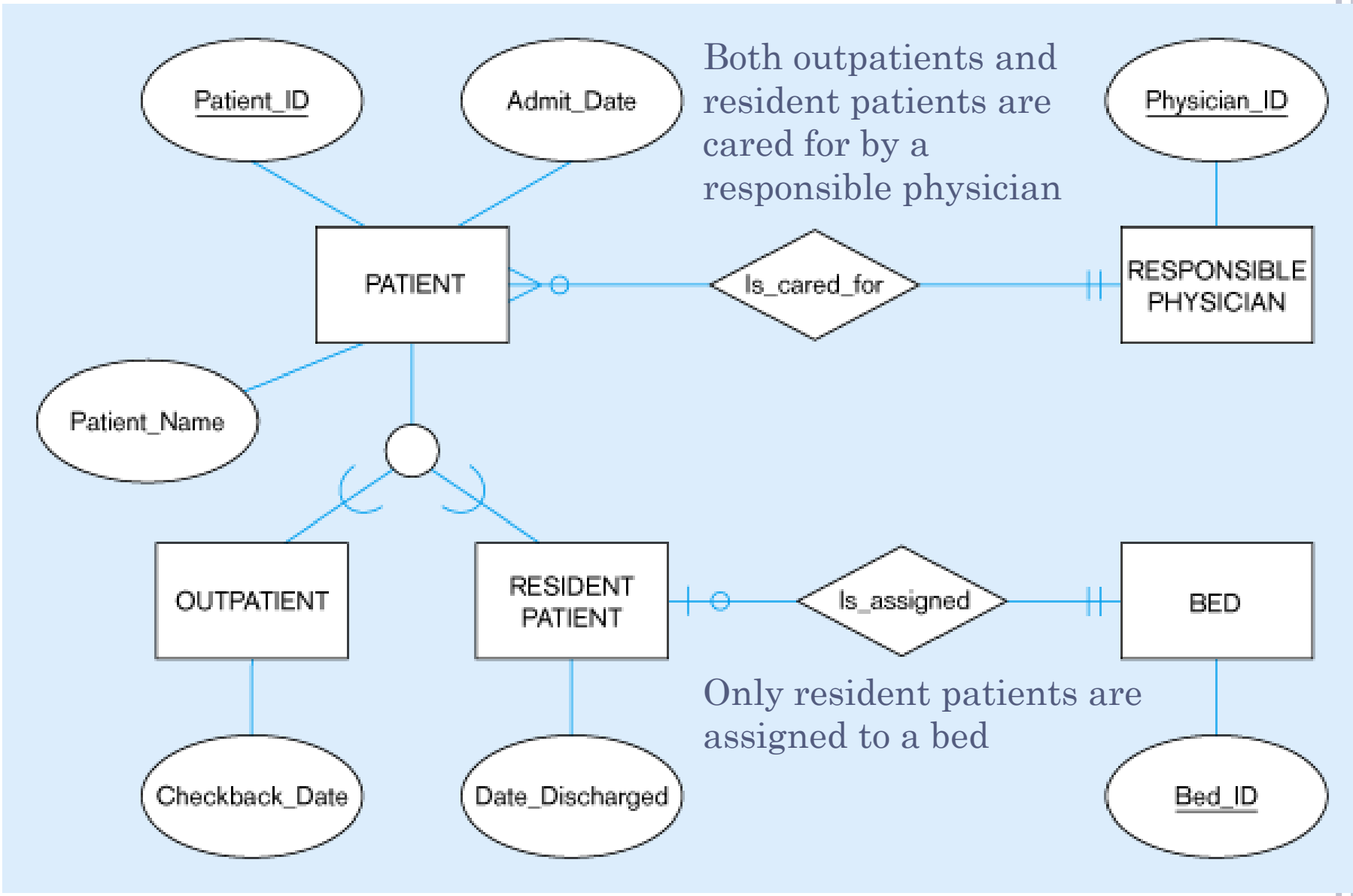


Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

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

Supertype/subtype relationships of patients



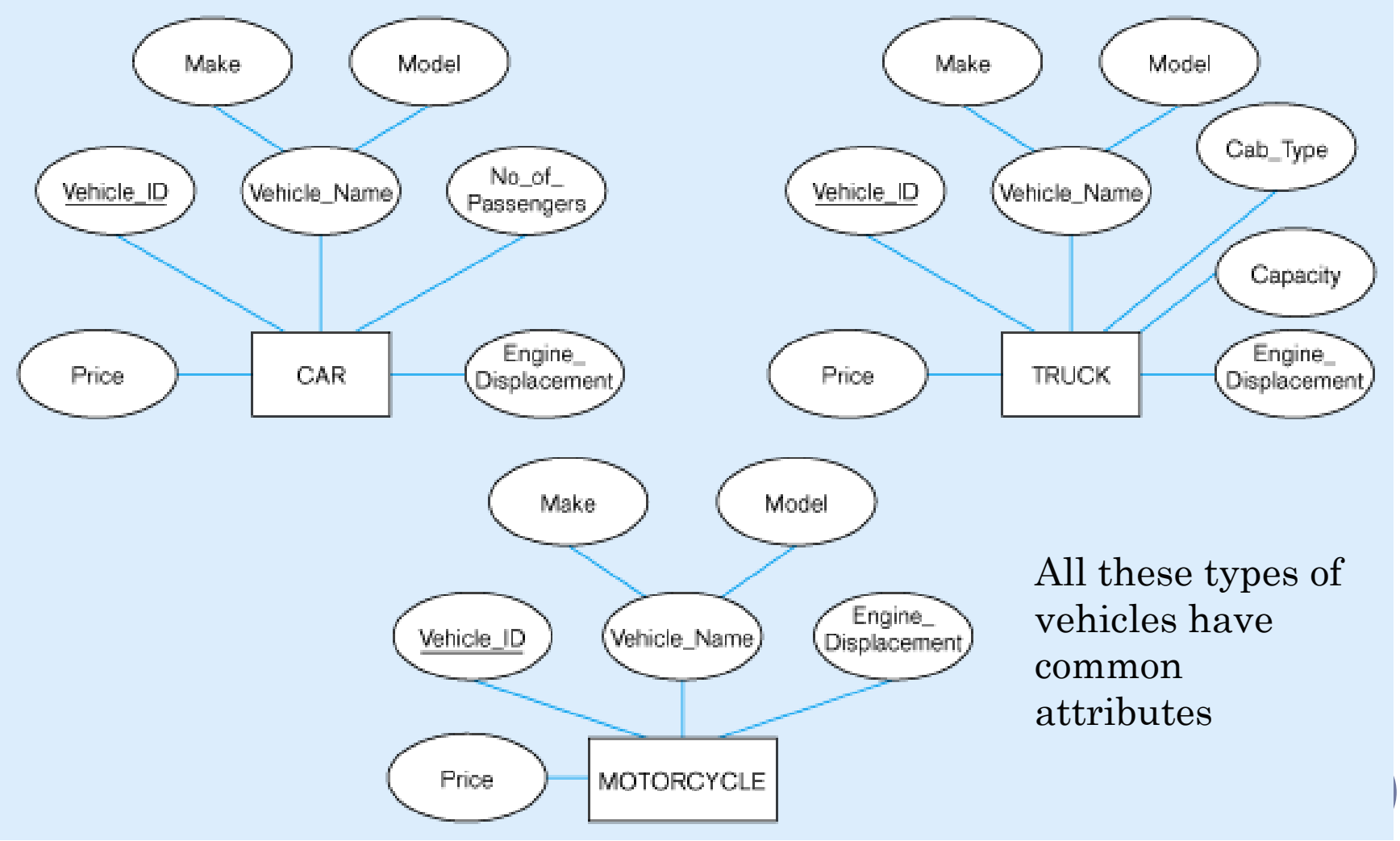
Generalization and specialization

- **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, and forming supertype/subtype relationships.
  - TOP-DOWN

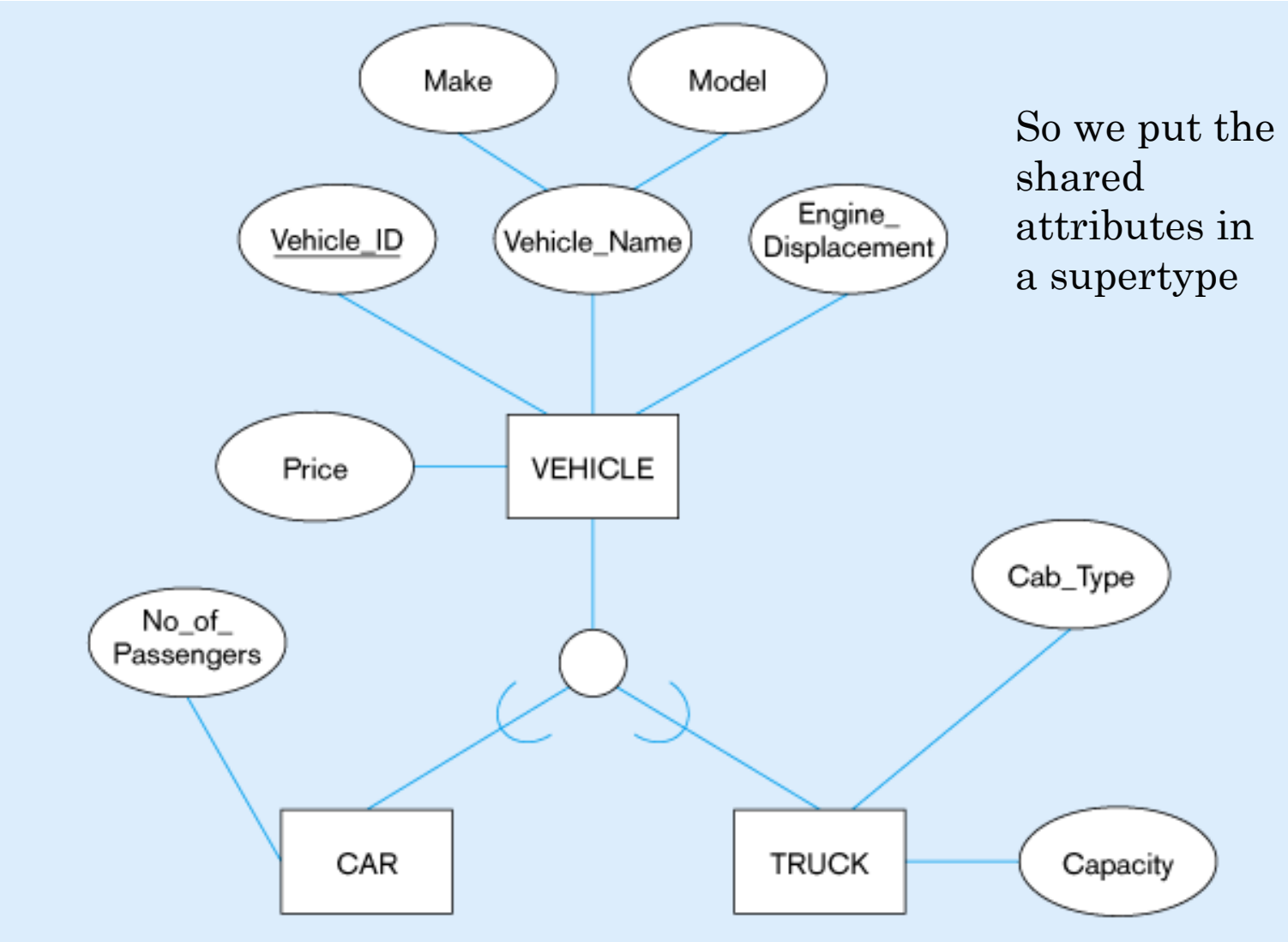
Example of Generalization

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

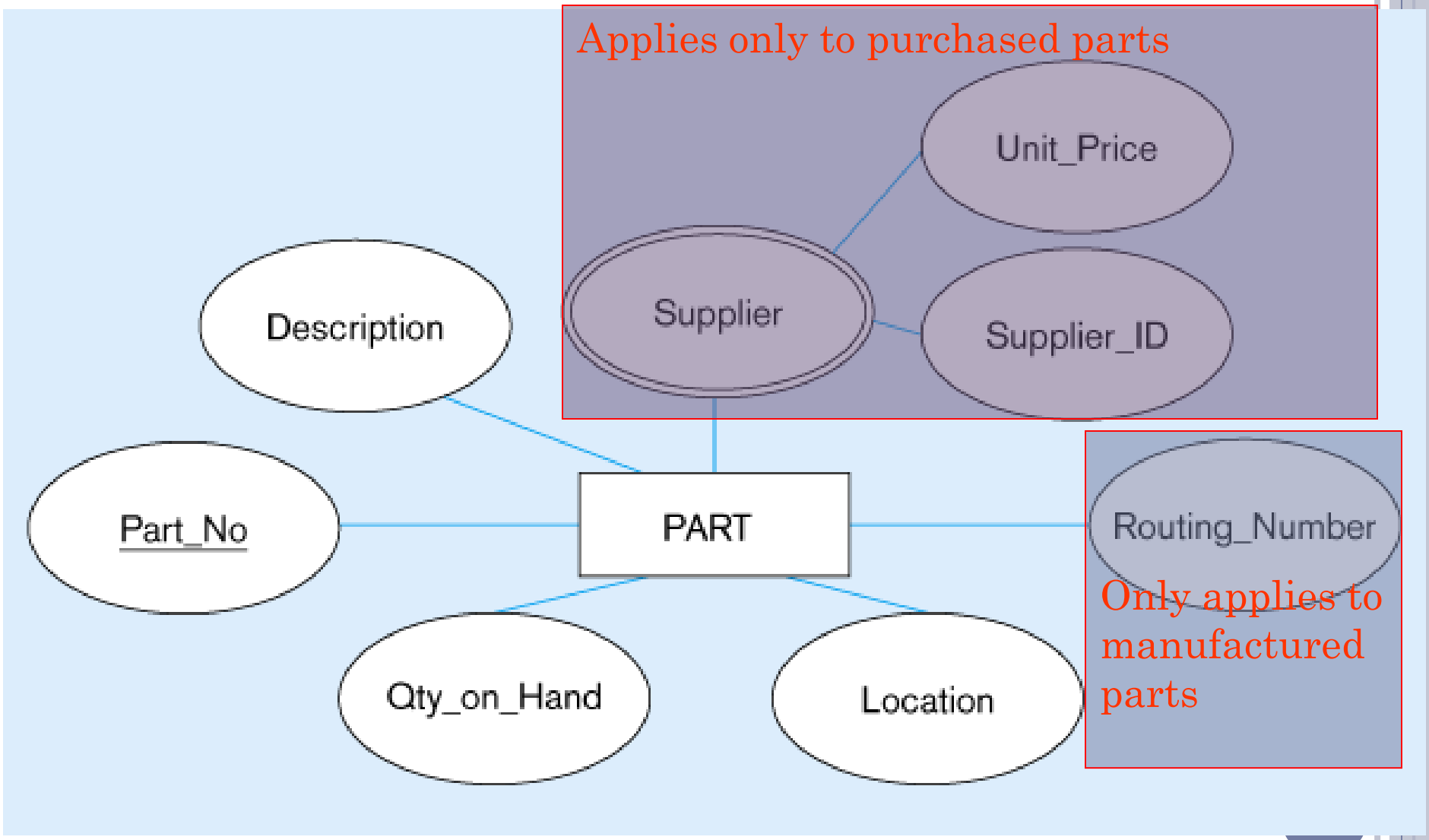
Note: no subtype for motorcycle, since it has no unique attributes

Specialisation

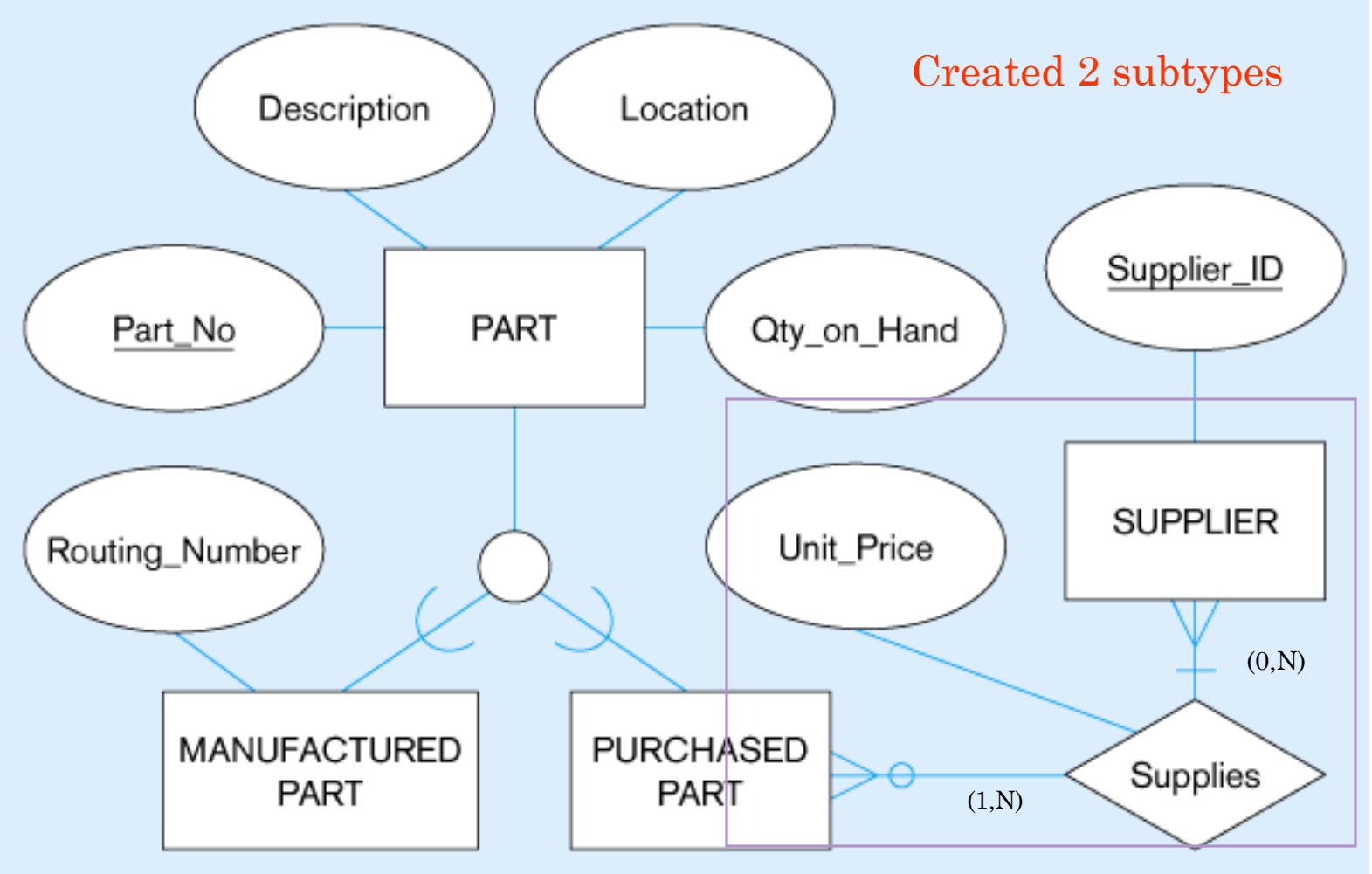
- An entity type PART has attributes
  - Part_No, Description, Unit_price, Location, Qty_On_Hand, Routing_Number and Supplier
  - There may be more than one supplier
- Some parts are internally Manufactured Parts while others are externally Purchased Parts
- Some parts are obtained from both sources
- 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



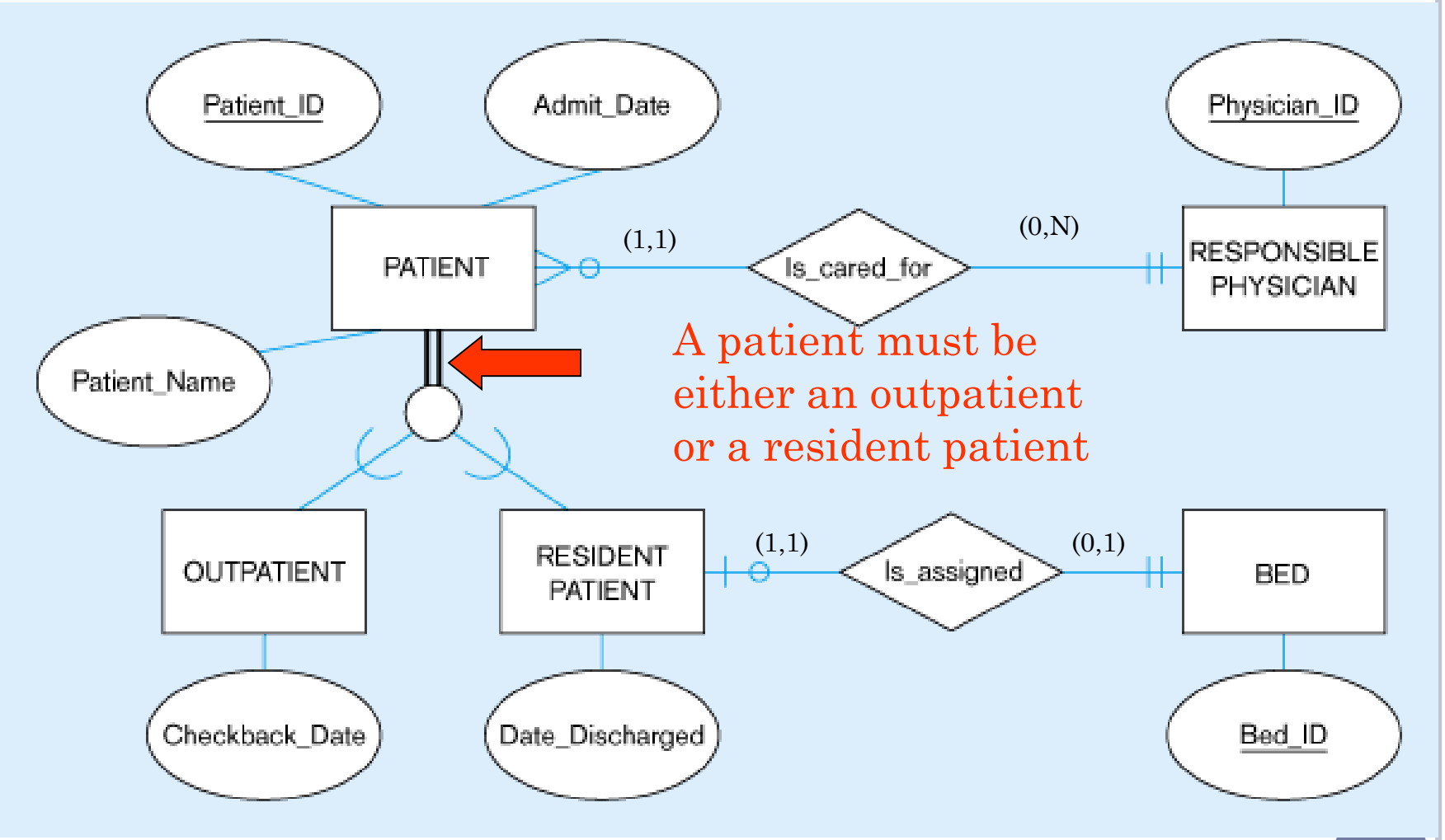
Note: multivalued attribute was replaced by a relationship to another entity

Completeness constraints

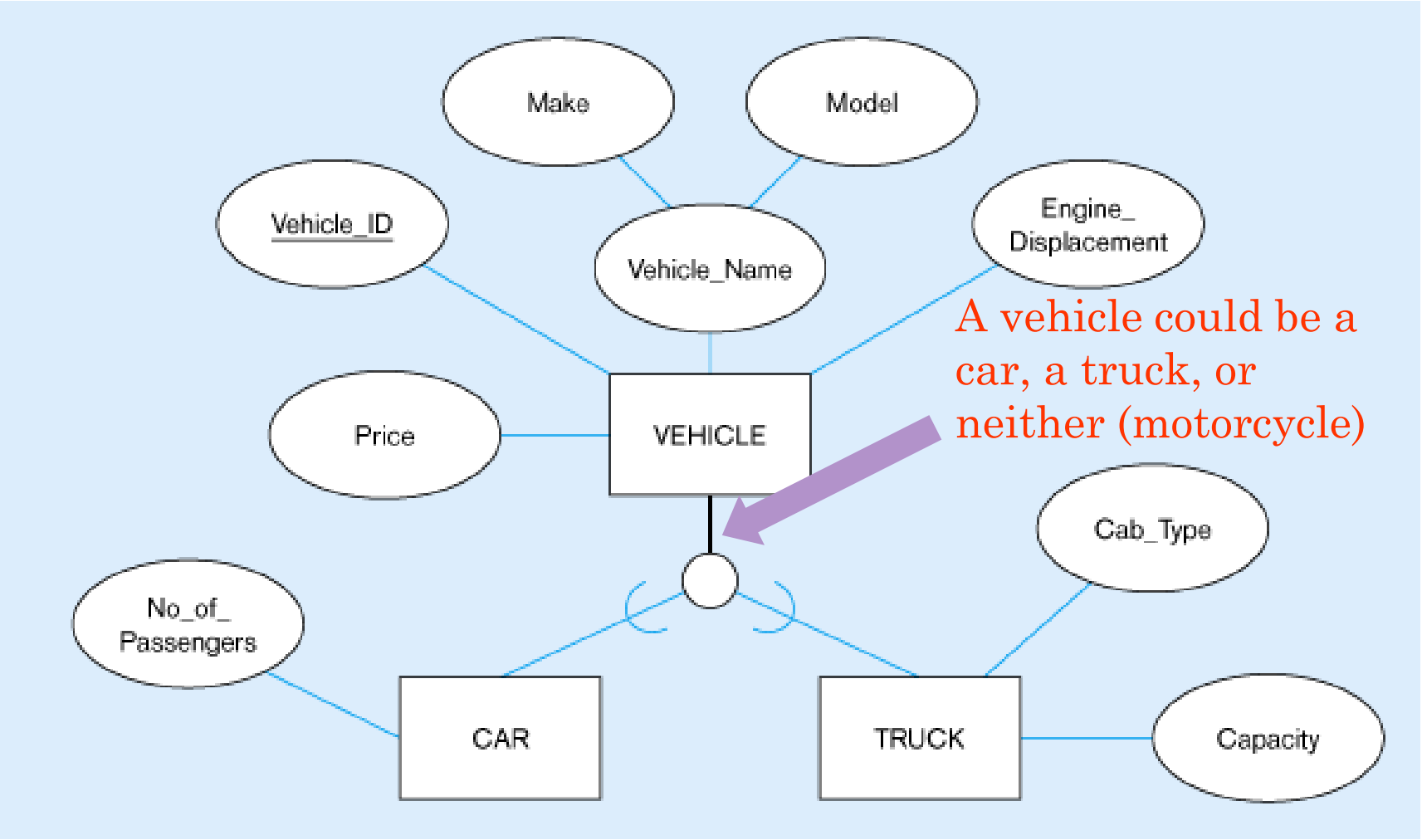
- Total Specialization: An entity instance of a supertype ***must*** also be a member of at least one subtype.

- Partial Specialization : An entity instance of the supertype is allowed not to belong to any subtype.

Examples of completeness constraints
Total specialization rule



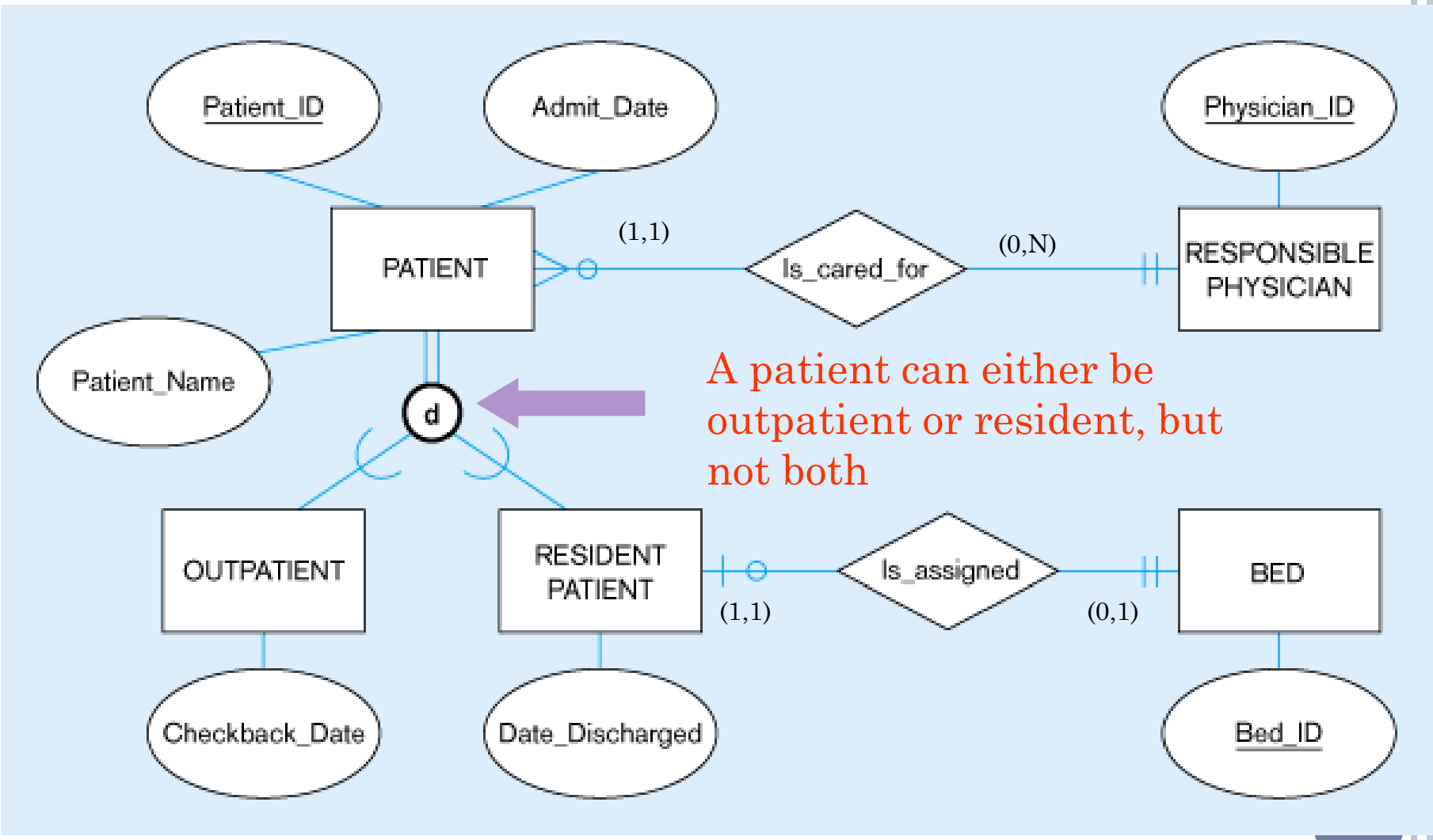
Partial specialization rule



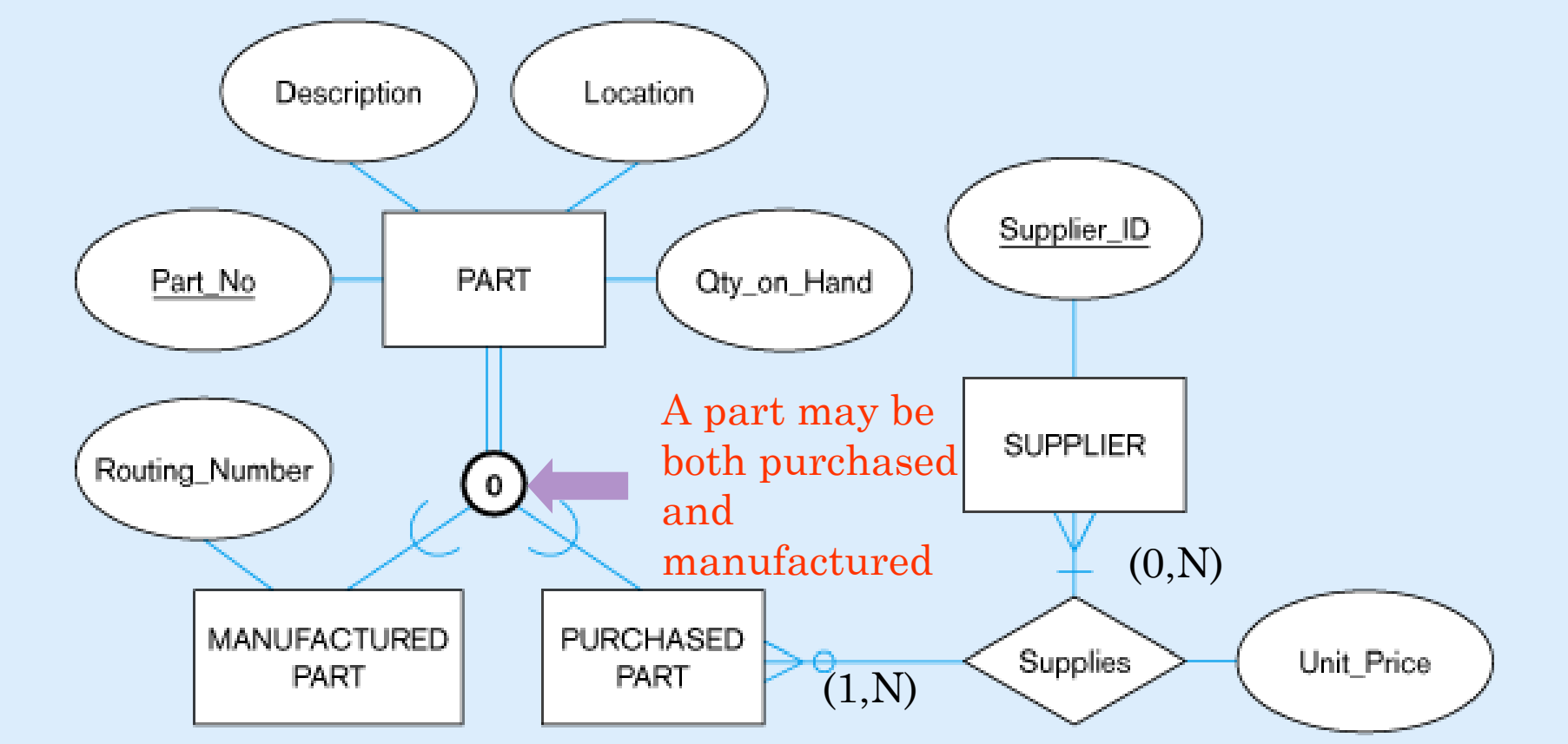
Disjointness constraint

- 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 disjointness constraints



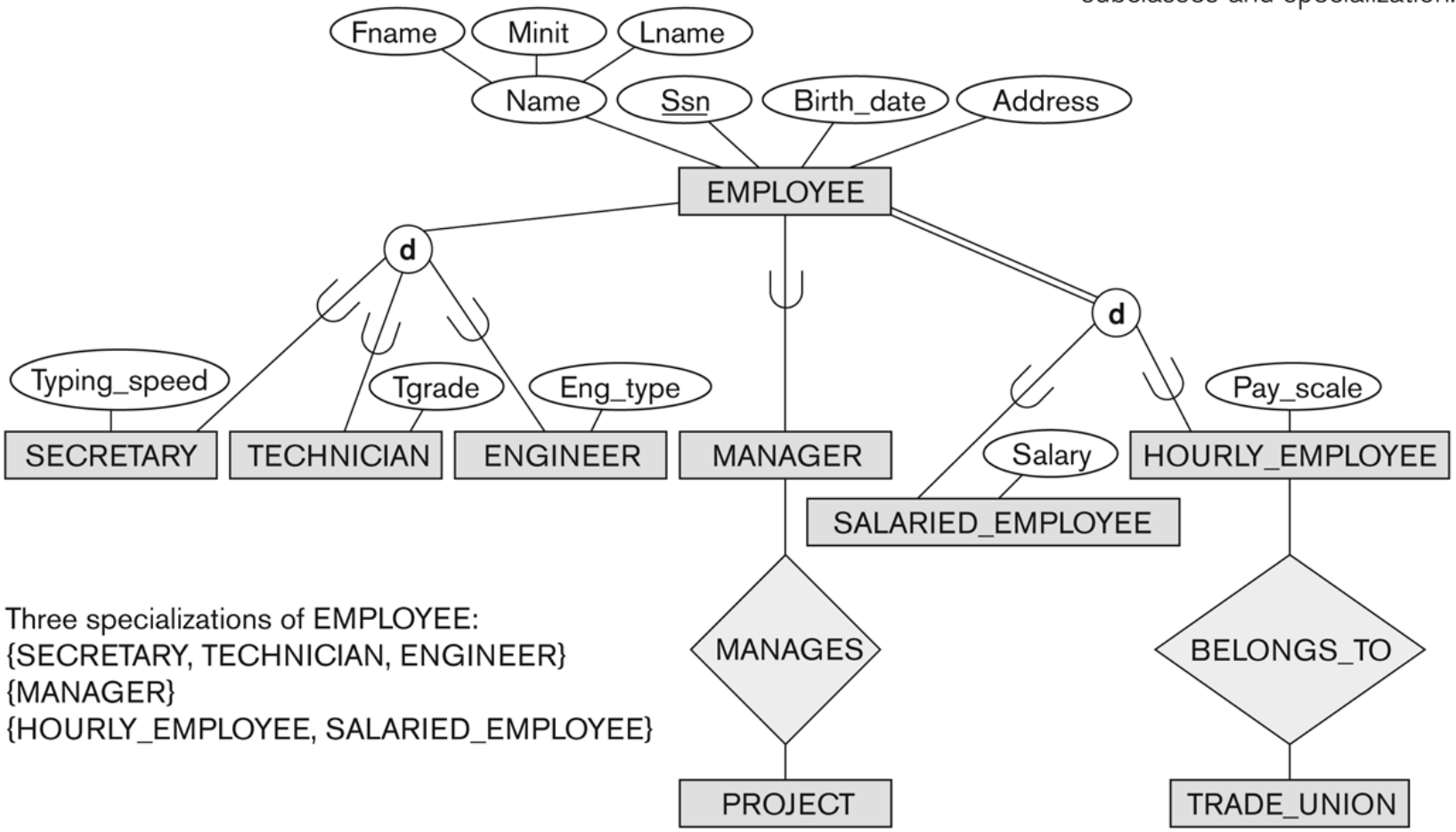
Overlap rule



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.



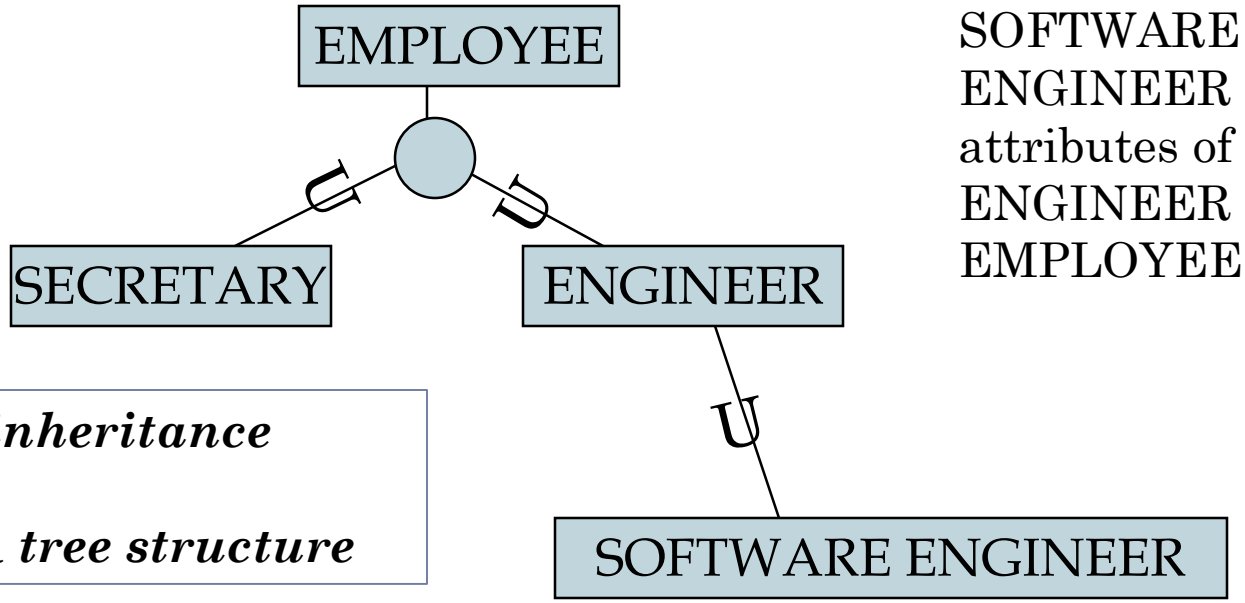
Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

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



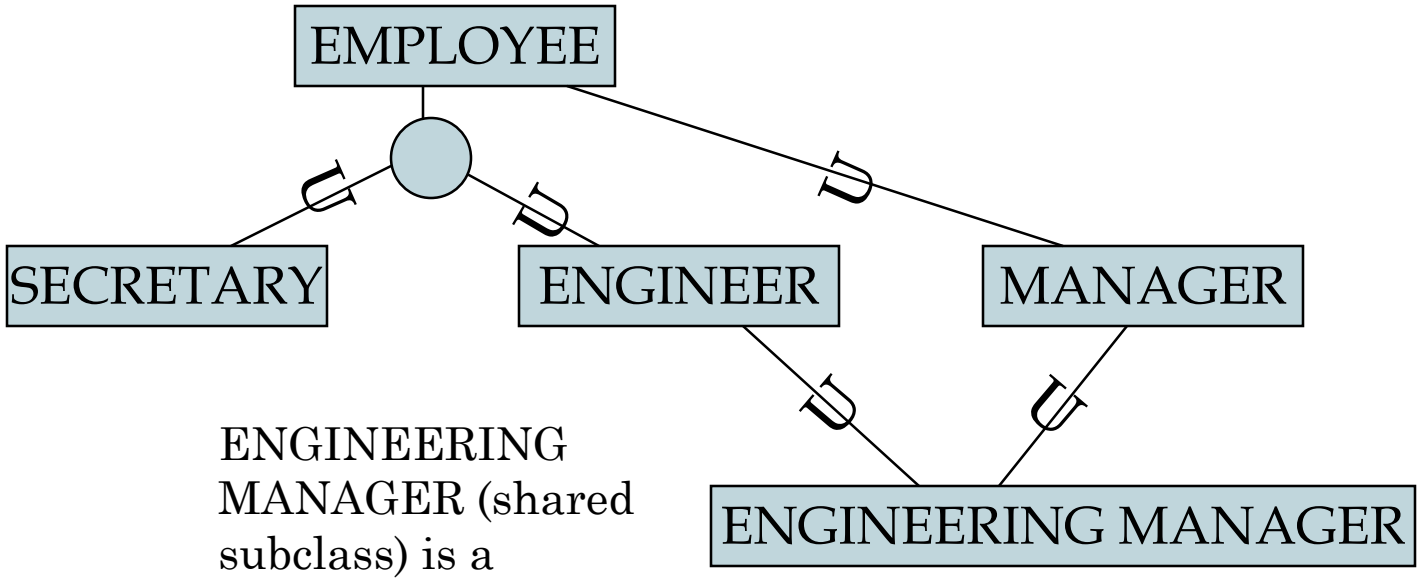
SOFTWARE ENGINEER has all the attributes of an ENGINEER and EMPLOYEE

single inheritance

Forms a ***tree structure***

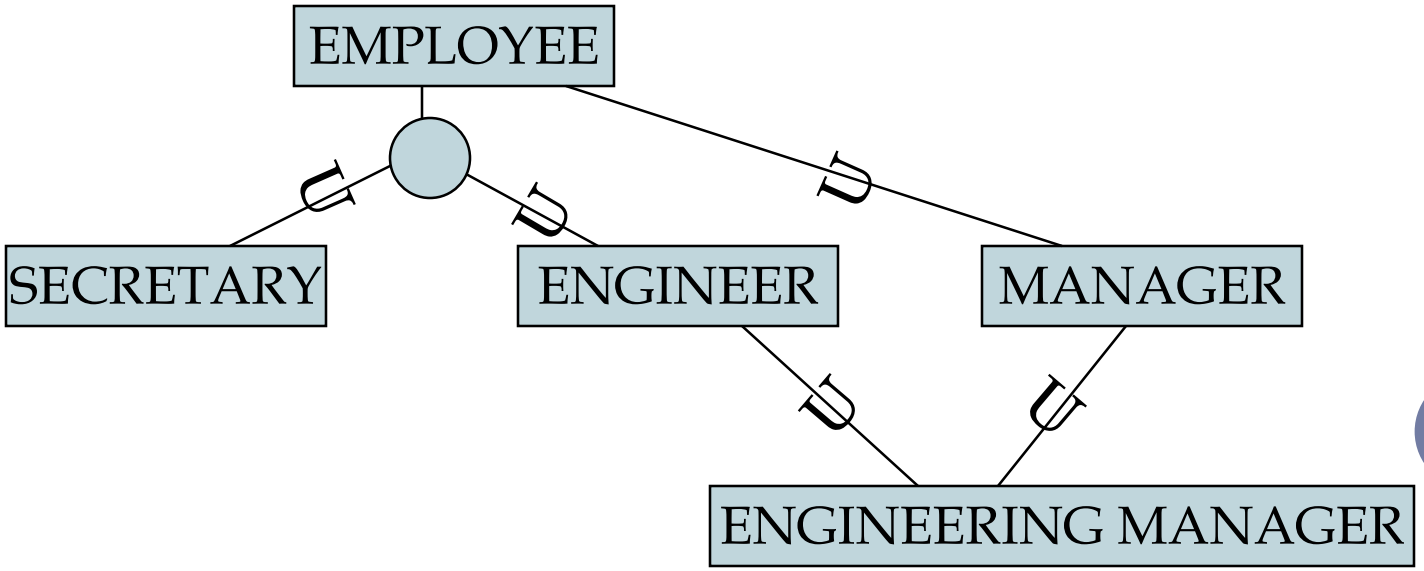
LATTICES(SHARED SUBCLASS)

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



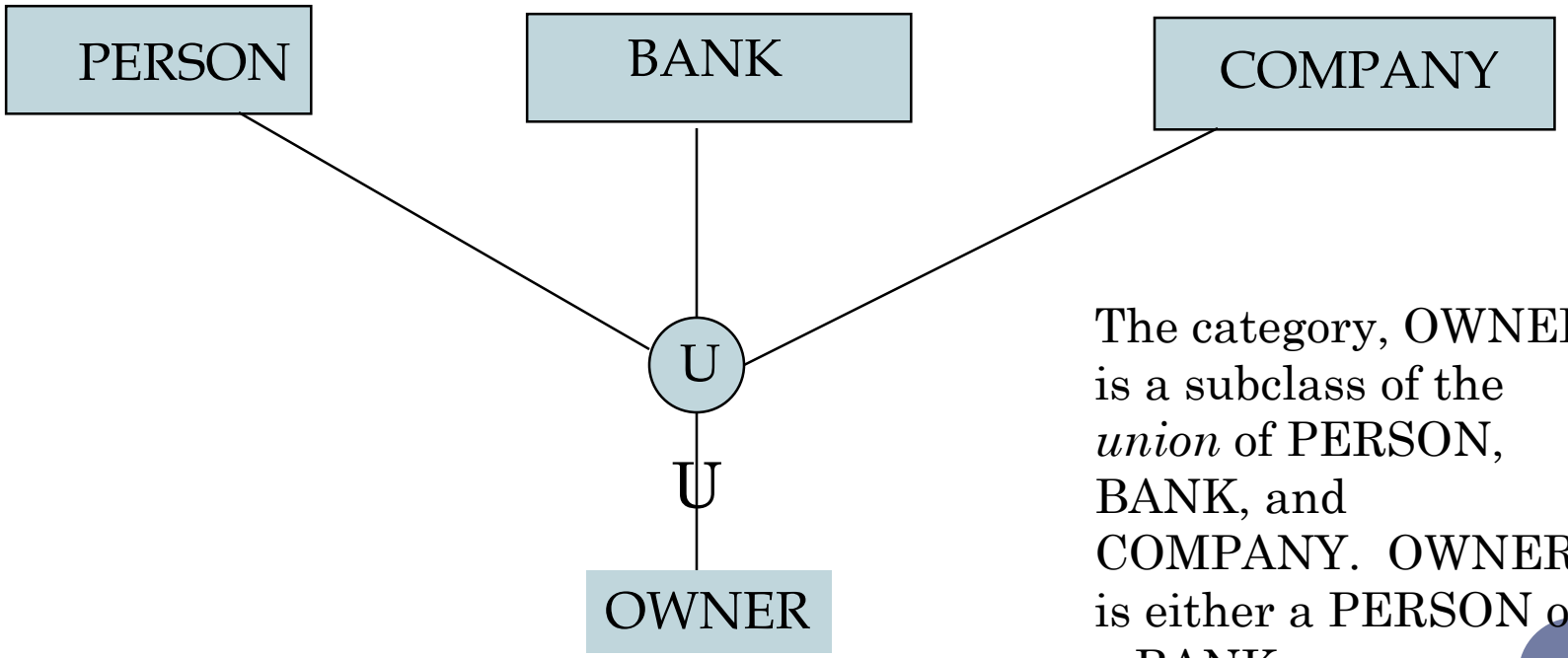
SHARED SUBCLASS

- 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



CATEGORIES (UNION TYPES)

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



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

Categories (UNION TYPES)

- In some cases, we need to model a *single superclass/subclass relationship* with *more than one* superclass
- Superclasses can represent different entity types
- Such a subclass is called a category or UNION TYPE

CATEGORIES (UNION TYPES)

- **Example:** In vehicle registration database, a vehicle owner can be a PERSON, a BANK (holding a lien on a vehicle) or a COMPANY.
  - 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

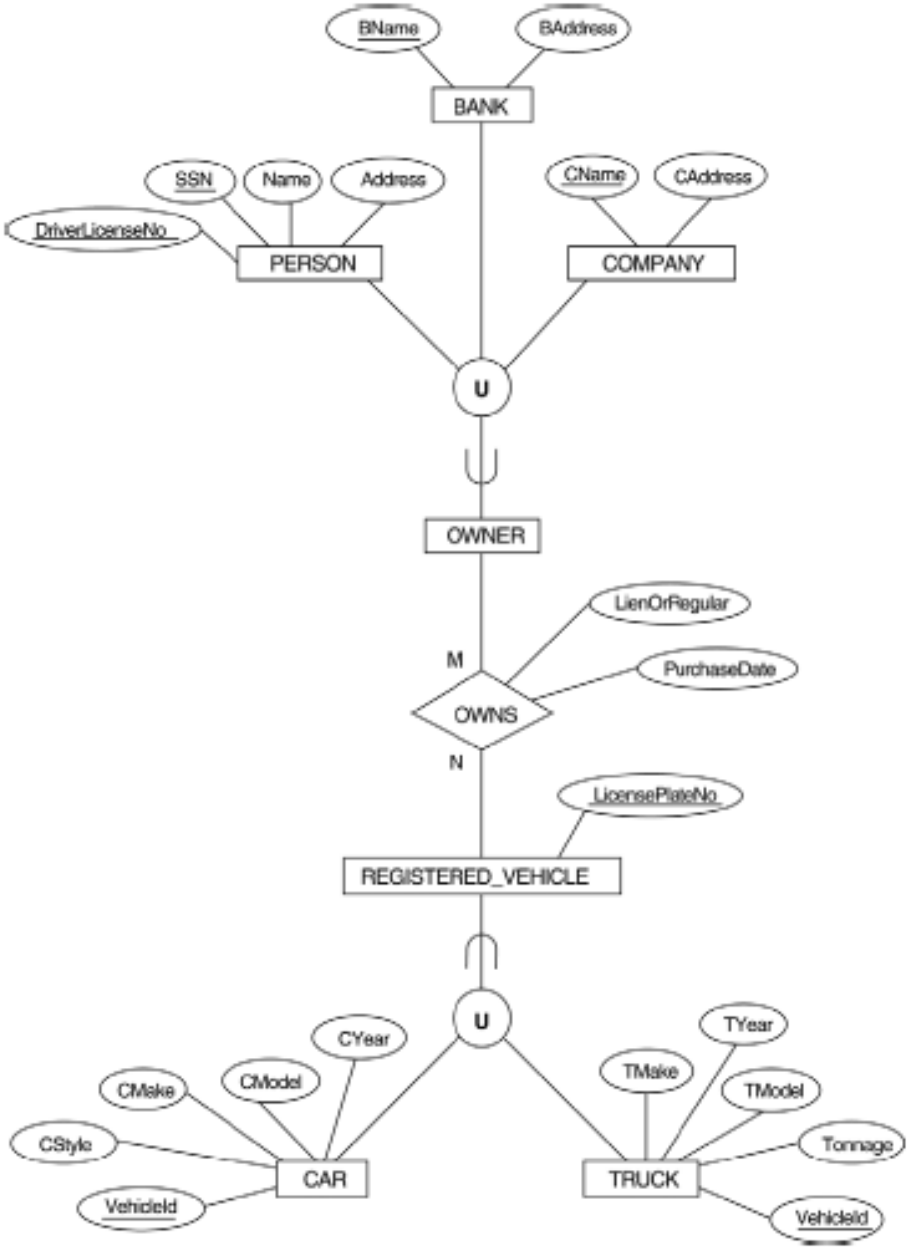
OWNER, REGISTERED_VEHICLE

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 total specialization or generalization

What is the difference between VEHICLE and REGISTERED_VEHICLE ?

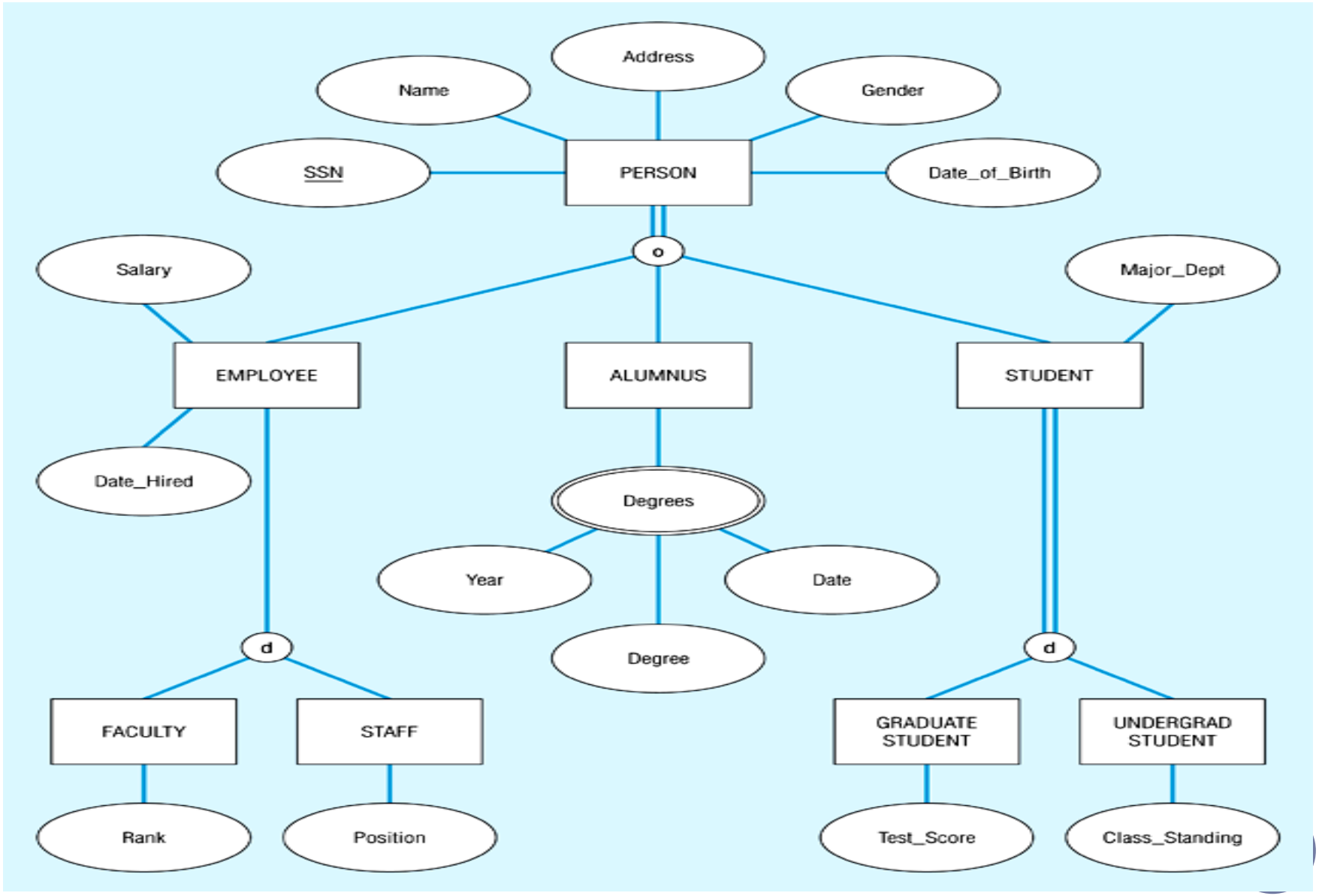


Defining supertype/subtype hierarchies

Let us model Human resources in a University

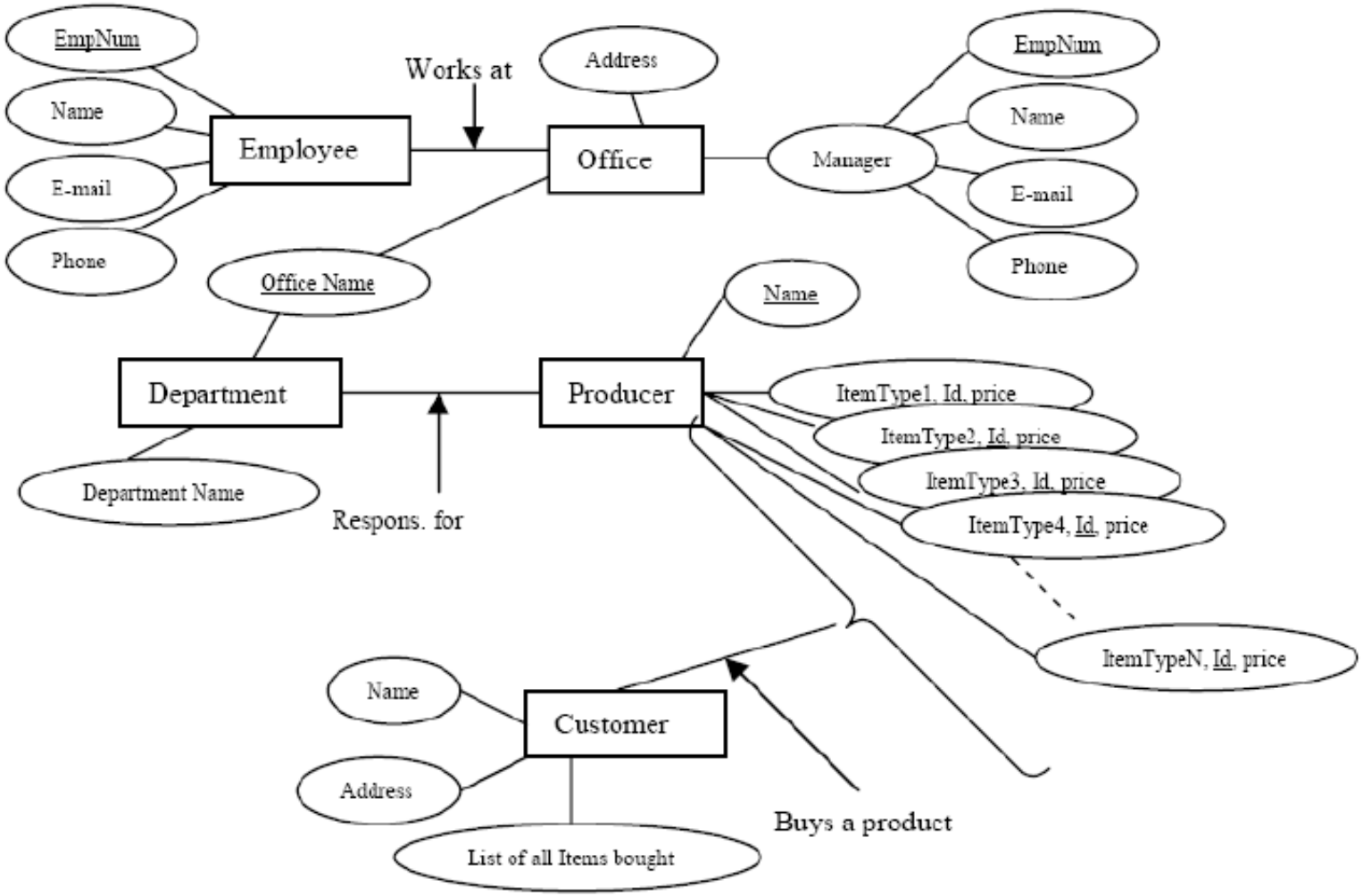
- We have three types of resources: EMPLOYEE, STUDENT and ALUMNUS (already graduated). All three types have attributes such as SSN, Name, Address, Gender and BirthDate.
- A person may belong to more than one subtype such as ALUMNUS and EMPLOYEE. Alumnus have degrees. And Employee gets salary.
- The two 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.

Example of supertype/subtype hierarchy



The diagram is supposed to model a company buying products from producers and selling to customers.

Find errors ??



Problem 2: ER for Notown Records

- Each musician has an SSN, name, address, phone. Poor musicians often share the same address, and no address has more than one phone.
- Each instrument that is used in songs recorded at Notown has a name (e.g., guitar, flute) and a musical key (e.g., C, B-flat).
- Each album has a title, a copyright date, a format (e.g., CD or MC), and an album identifier.
- Each song recorded at Notown has a title and an author.
- Each musician may play several instruments, and an instrument may be played by several musicians.
- Each album has a number of songs on it, but no song may appear on more than one album.
- Each song is performed by one or more musicians, and a musician may perform a number of songs.
- Each album has exactly one musician who acts as its producer. A musician may produce several albums, of course.

Alternative diagrammatic notations

- ER/EER diagrams are a specific notation for displaying the concepts of the model diagrammatically
- DB design tools use many alternative notations for the same or similar concepts
- One popular alternative notation uses *UML class diagrams*

UML Example for Displaying Specialization / Generalization

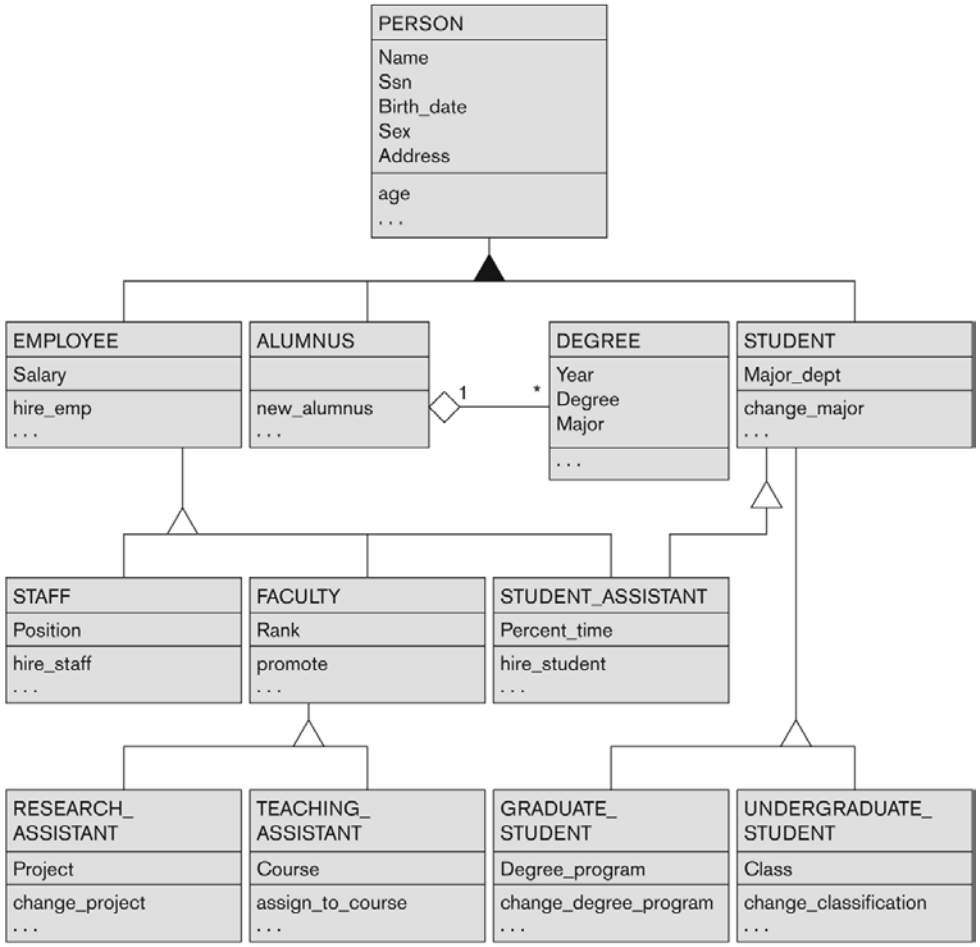


Figure 4.10
A UML class diagram corresponding to the EER diagram in Figure 4.7, illustrating UML notation for specialization/generalization.

Alternative Diagrammatic Notations

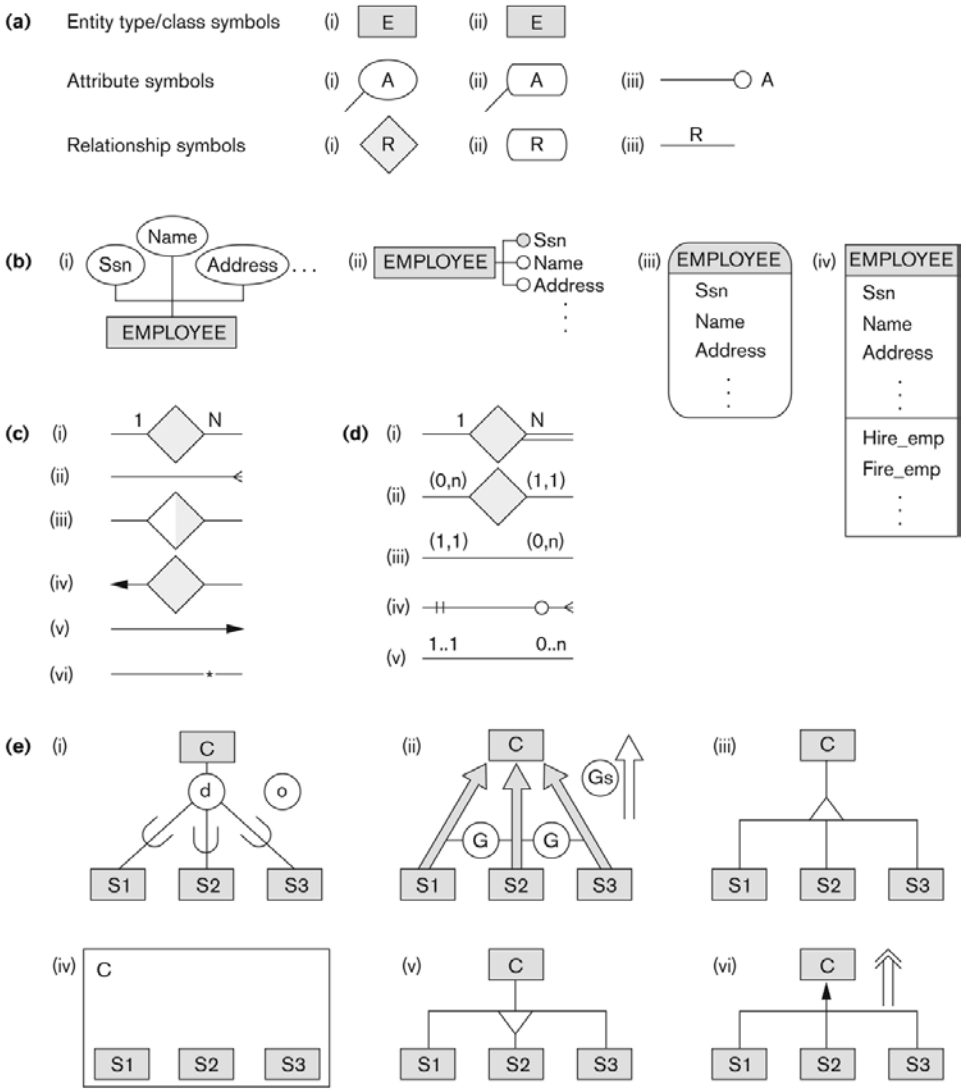


Figure A.1
Alternative notations. (a) Symbols for entity type/class, attribute, and relationship. (b) Displaying attributes. (c) Displaying cardinality ratios. (d) Various (min, max) notations. (e) Notations for displaying specialization/generalization.

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)

Ontologies

- Use conceptual modeling and other tools to develop "a specification of a conceptualization"
  - **Specification** refers to the language and vocabulary (data model concepts) used
  - **Conceptualization** refers to the description (schema) of the concepts of a particular field of knowledge and the relationships among these concepts
- Many medical, scientific, and engineering ontologies are being developed as a means of standardizing concepts and terminology

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