

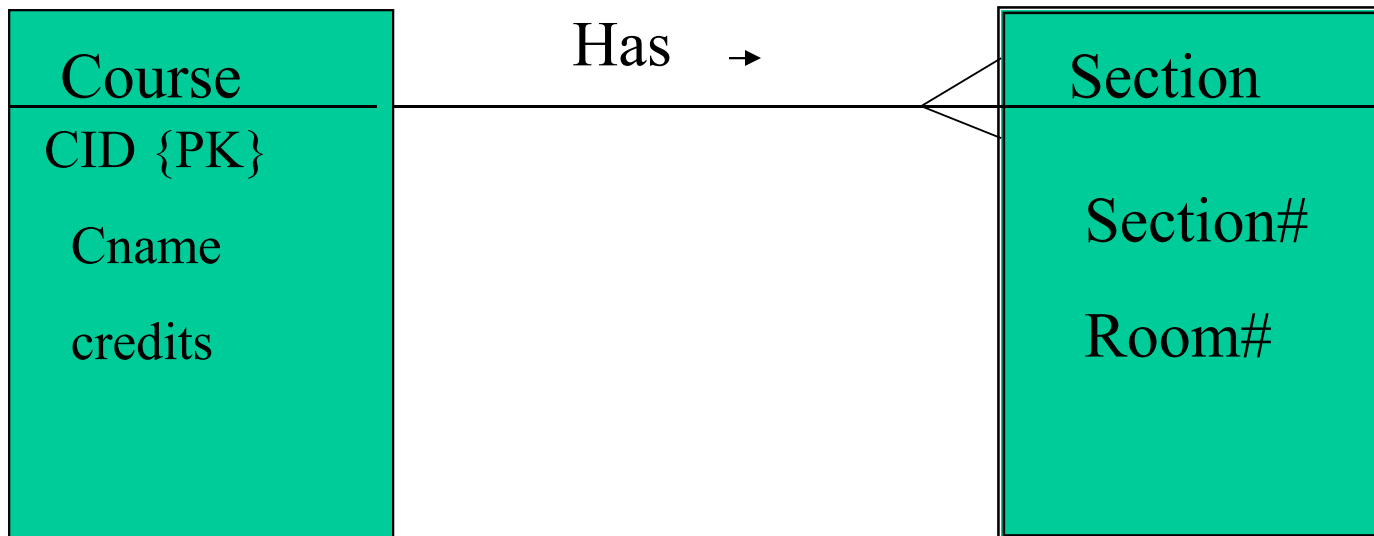
Enhanced Entity-Relationship Modeling

Strong and Weak Entity Types

- Strong entity: Each object is uniquely identifiable using primary key of that entity type.
- Weak entity: Each object cannot be uniquely identified using only the attributes associated with that entity type.
 - Entities that cannot exist in the database unless another entity also exists.

Weak Entity Examples

- Course: CID, Cname, Credits
 - Section: Section#, Room
- MedicalBldg: BuilidngID, Bname, Address
 - Office: Room#
- Web page and placeholders



Supertypes and Subtypes

- **Supertype:** A generic entity type that has a relationship with one or more subtypes
 - An entity type that includes one or more distinct subgroups which require to be represented in a data model.
- **Subtype:** A subgrouping of the entities in an entity type that has attributes distinct from those in other subgroupings
- **Attribute Inheritance:**
 - Subtype entities inherit values of all attributes of the supertype
 - An instance of a subtype is also an instance of the supertype

Figure 4-1 Basic notation for supertype/subtype notation

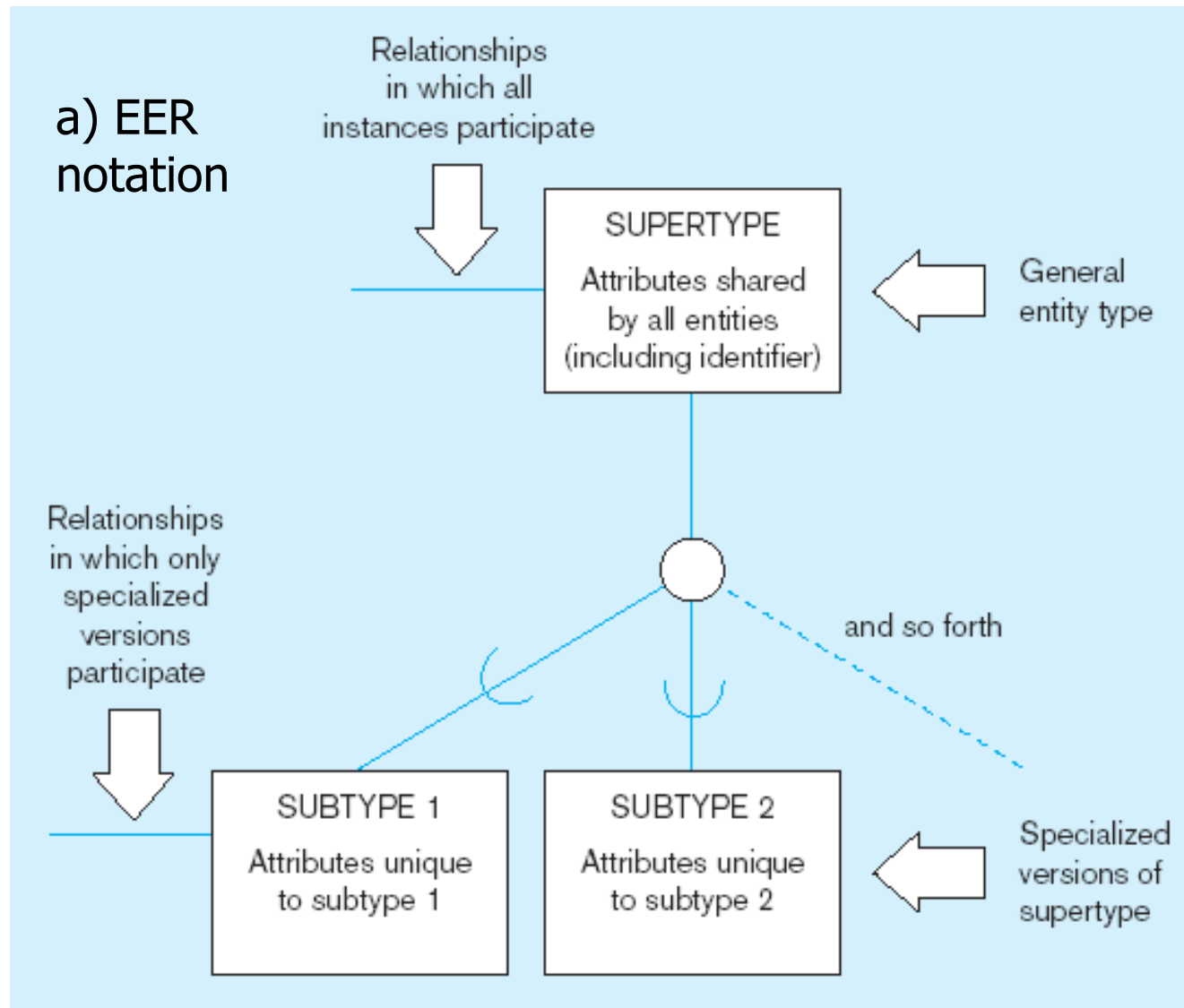
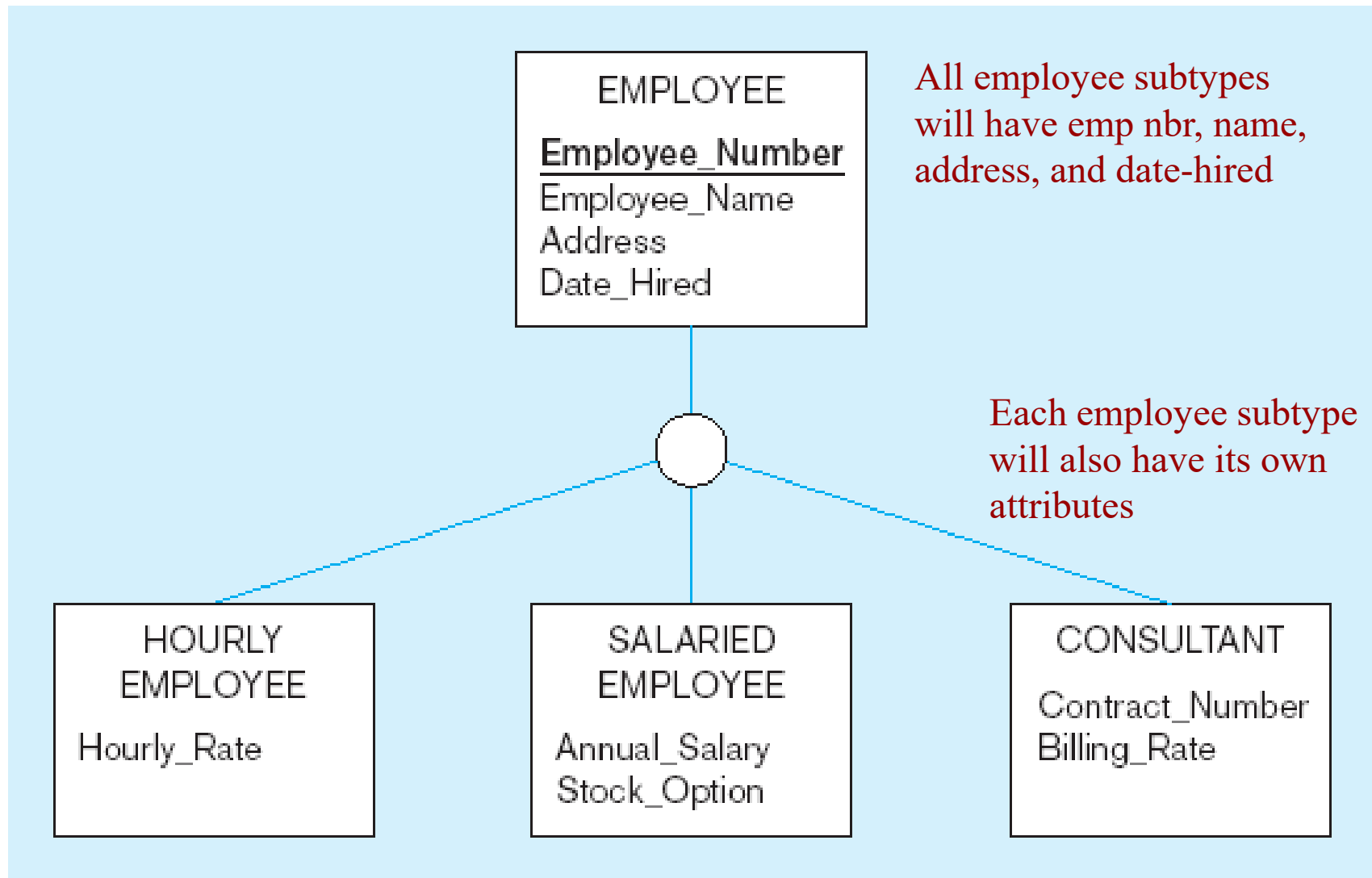


Figure 4-2 Employee supertype with three subtypes



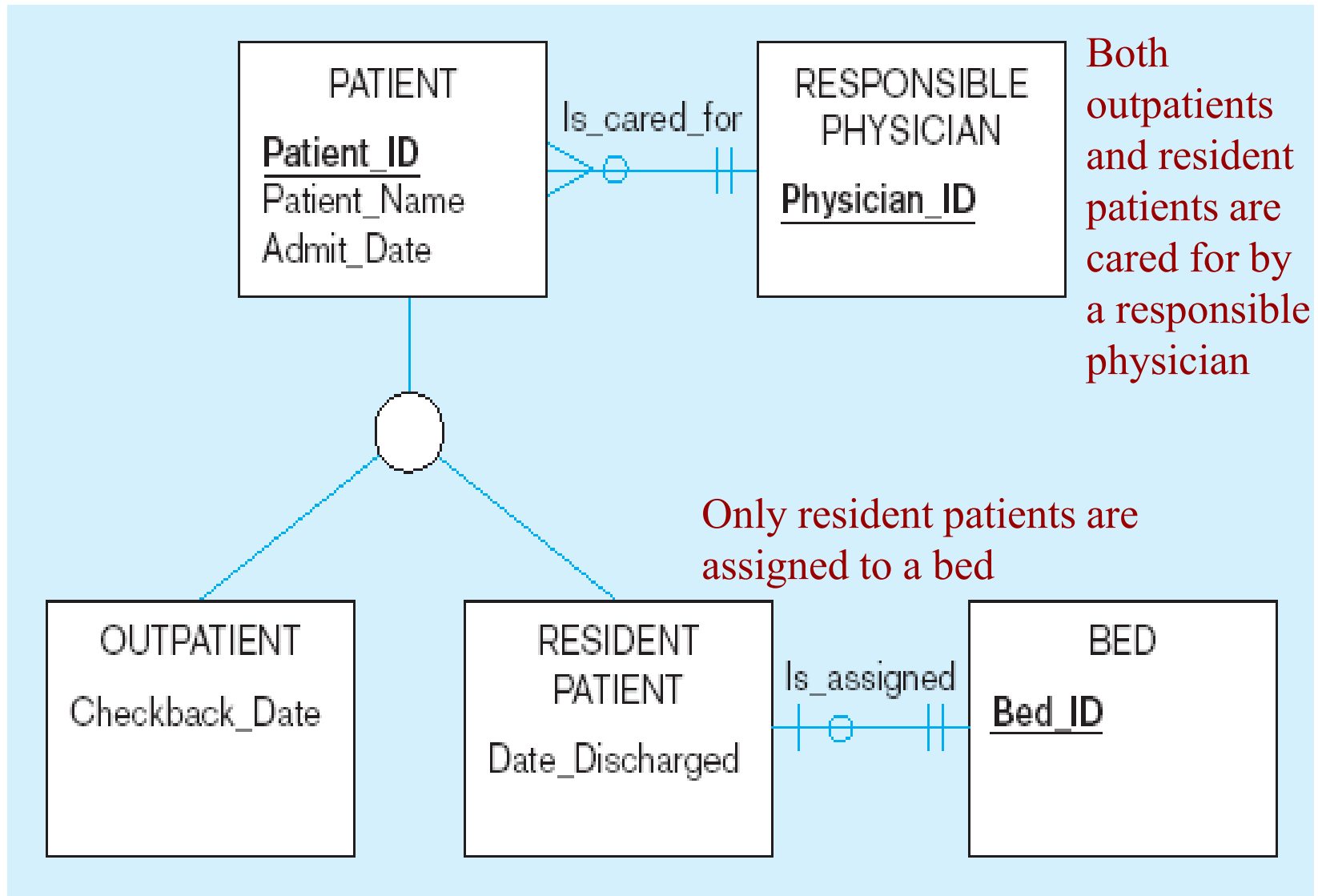
- Employee:
 - EID, Ename, Salary, Position, TypingSpeed, ManagerBonus, SalesArea, CarAllowance
- Employee:EID, Ename, Salary, Position
 - Manager: managerBonus
 - SalesPerson: SalesArea, CarAllowance
 - Secretary: TypingSpeed

- Faculty
 - Professor
 - Lecturer
 - GraduateAssistant

Relationships and Subtypes

- Relationships at the *supertype* level indicate that all subtypes will participate in the relationship
- The instances of a *subtype* may participate in a relationship unique to that subtype. In this situation, the relationship is shown at the subtype level

Figure 4-3 Supertype/subtype relationships in a hospital

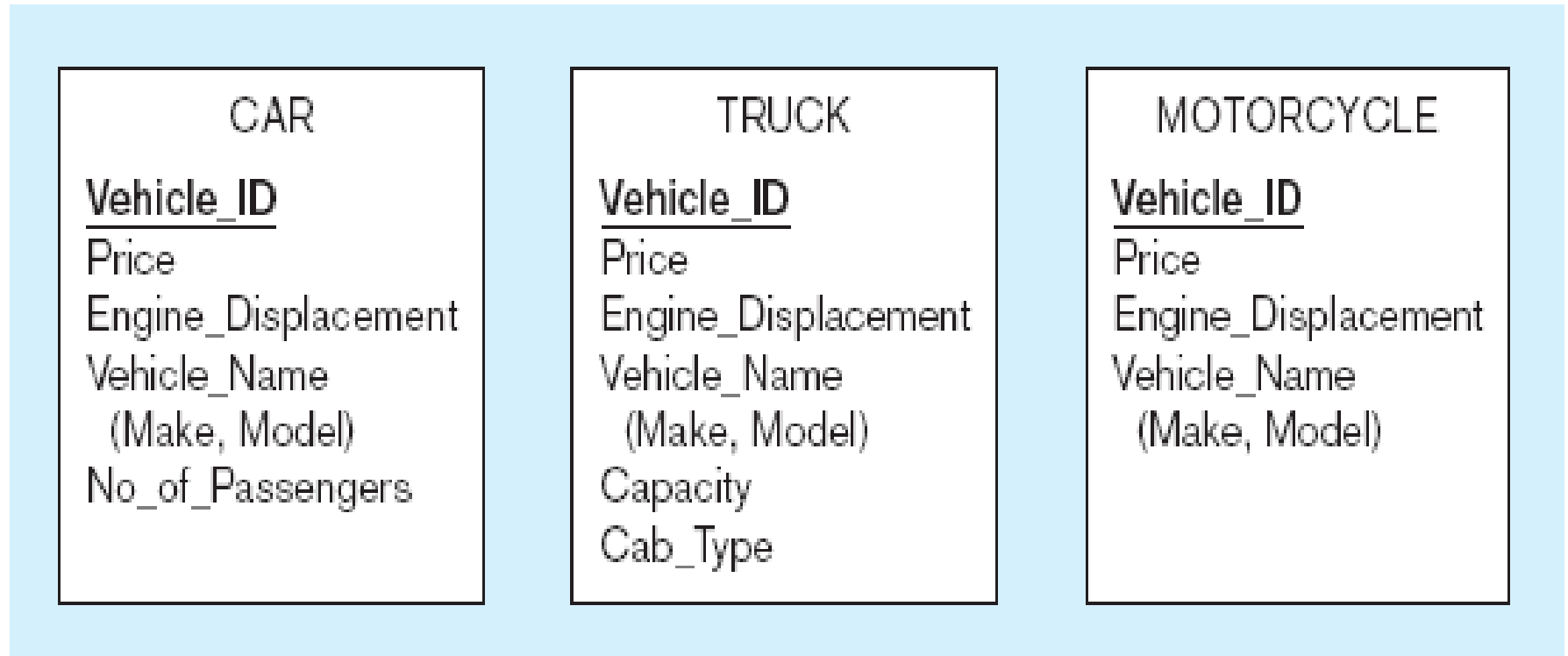


Generalization and Specialization

- ***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

Figure 4-4 Example of generalization

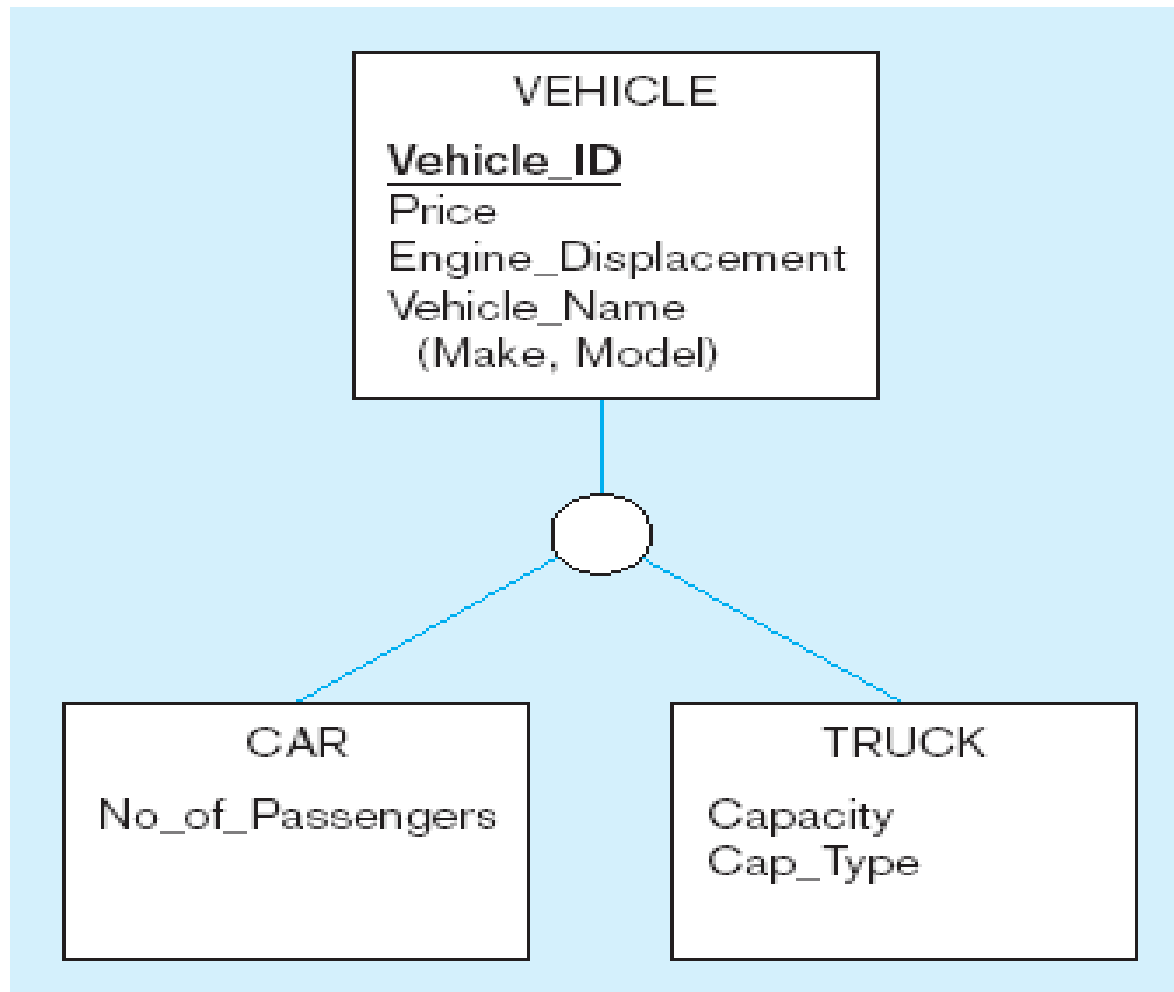
a) Three entity types: CAR, TRUCK, and MOTORCYCLE



All these types of vehicles have common attributes

Figure 4-4 Example of generalization (cont.)

b) Generalization to VEHICLE supertype



So we put
the shared
attributes in
a supertype

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

Figure 4-5 Example of specialization

a) Entity type PART

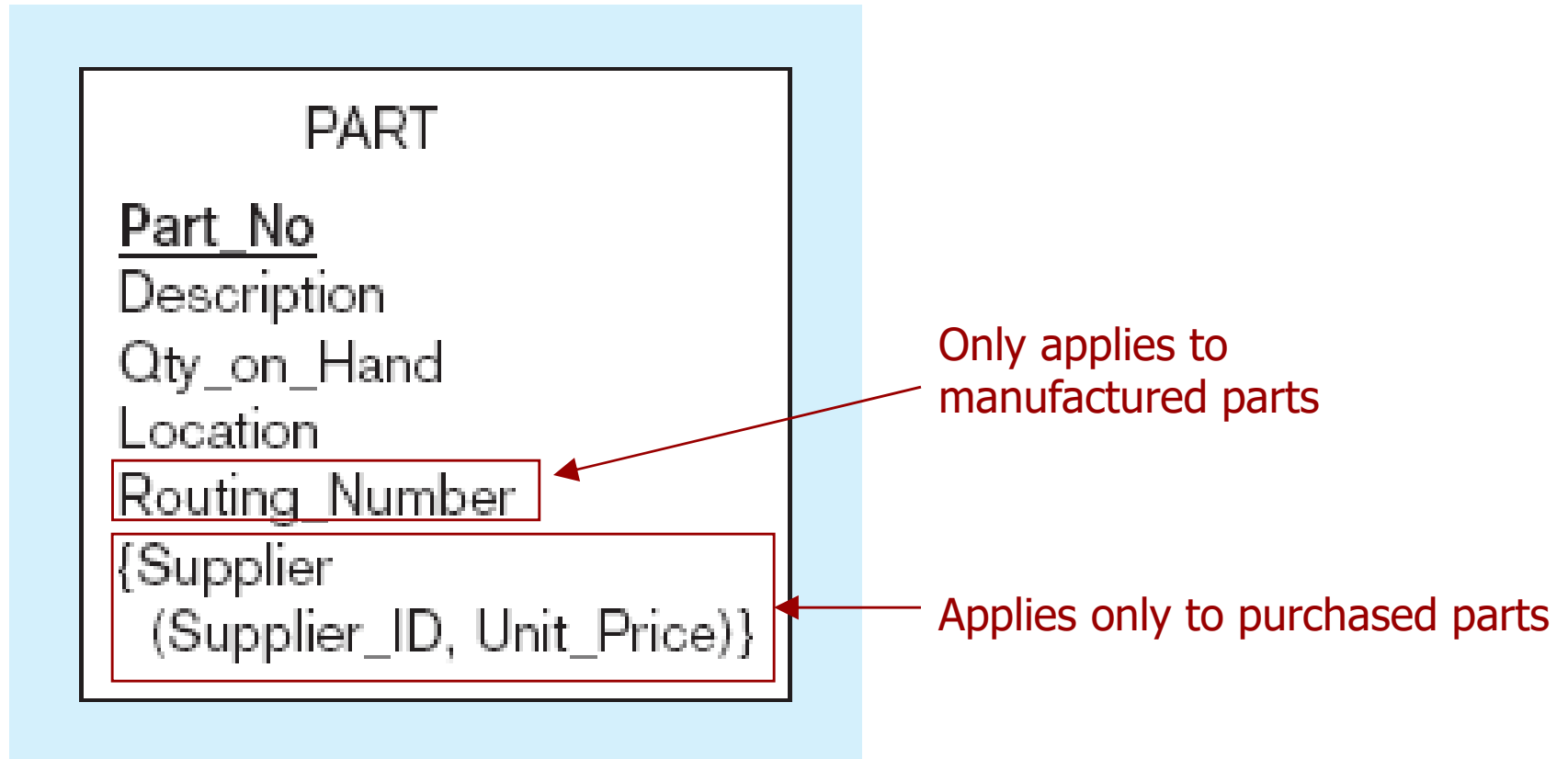
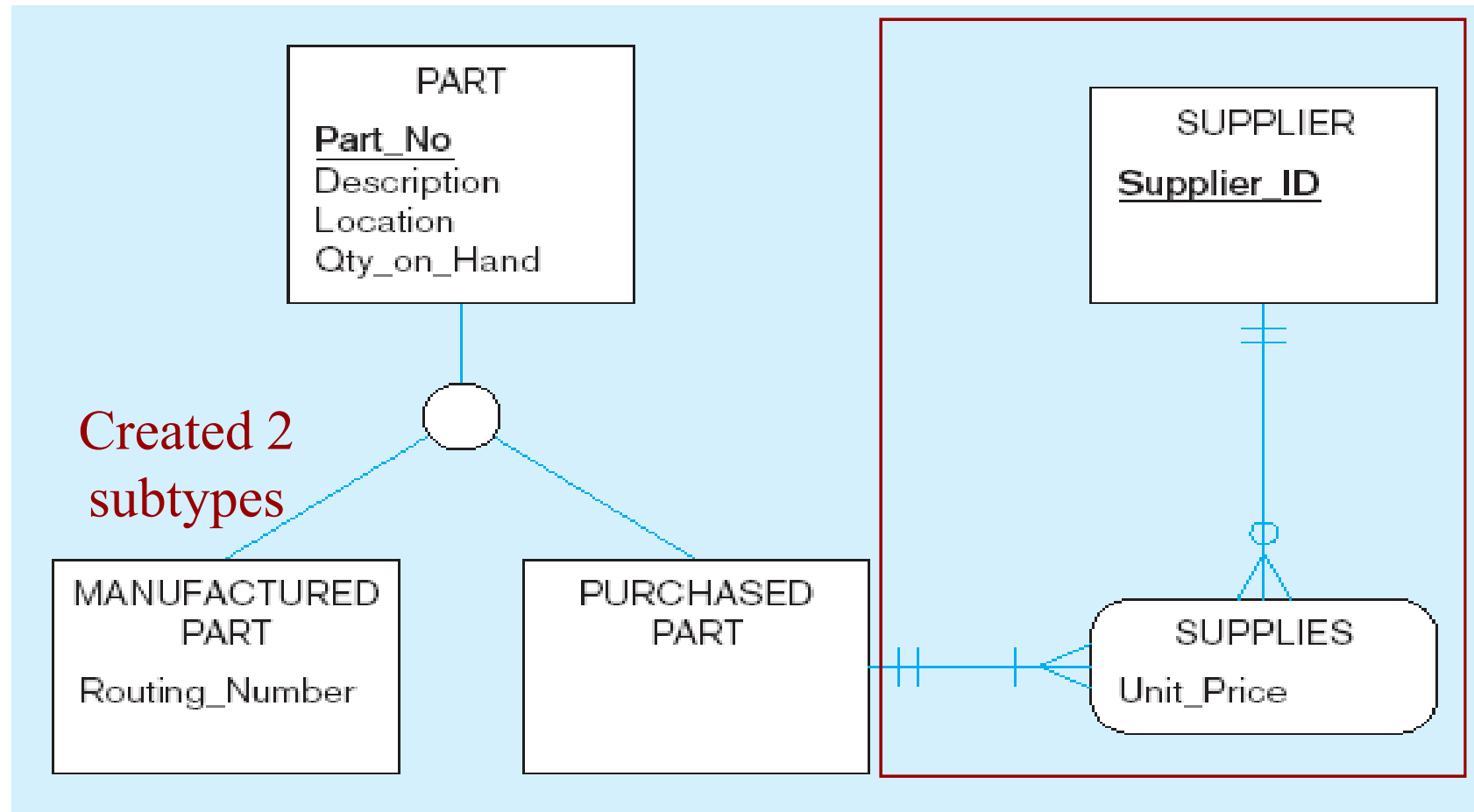


Figure 4-5 Example of specialization (cont.)

b) Specialization to MANUFACTURED PART and PURCHASED PART



Note: multivalued attribute was replaced by an associative entity relationship to another entity

Constraints in Supertype/ Completeness Constraint

- **Completeness Constraints**: Whether an instance of a supertype *must* also be a member of at least one subtype
 - Total Specialization Rule: Yes (double line)
 - Partial Specialization Rule: No (single line)

Figure 4-6 Examples of completeness constraints

a) Total specialization rule

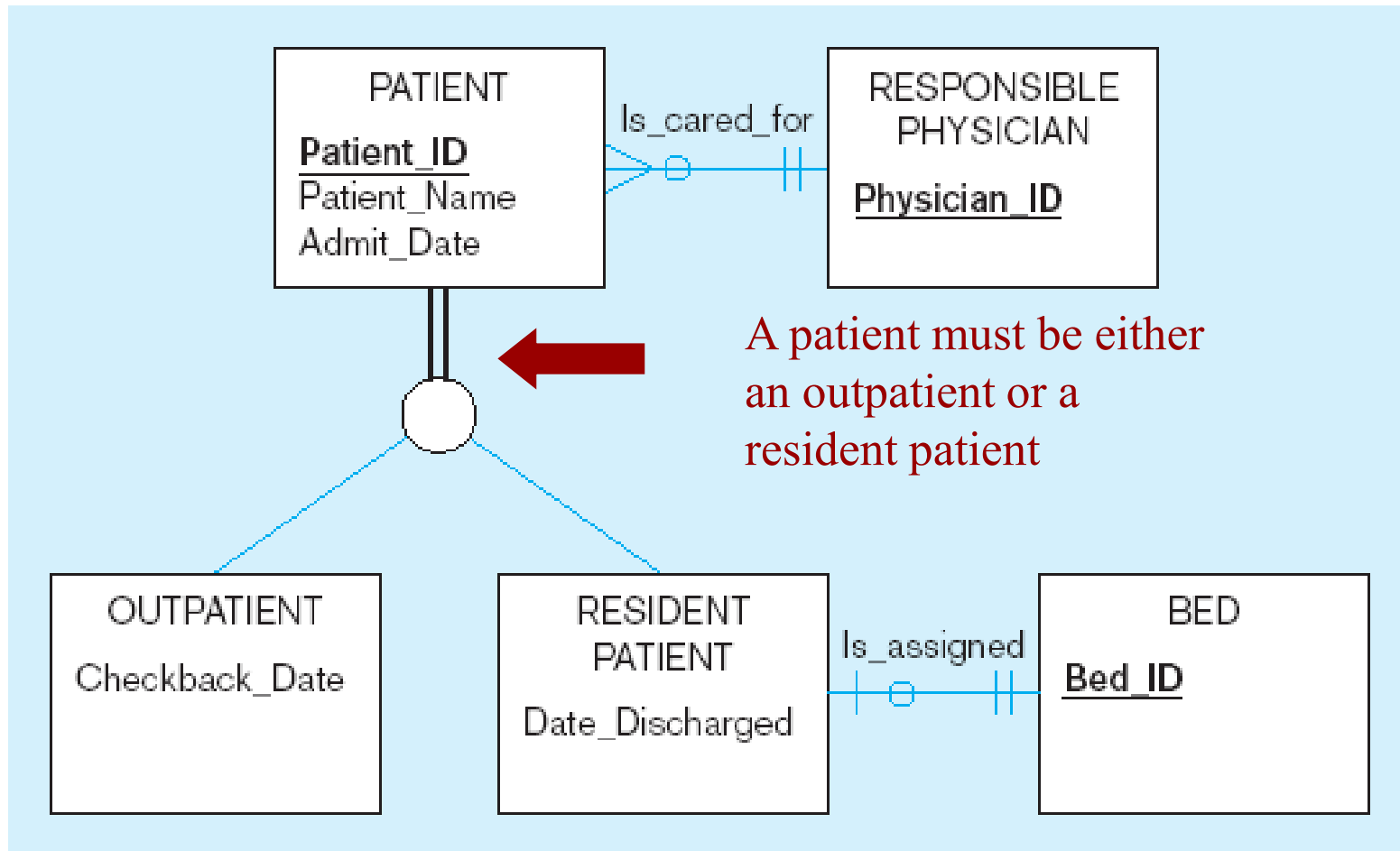
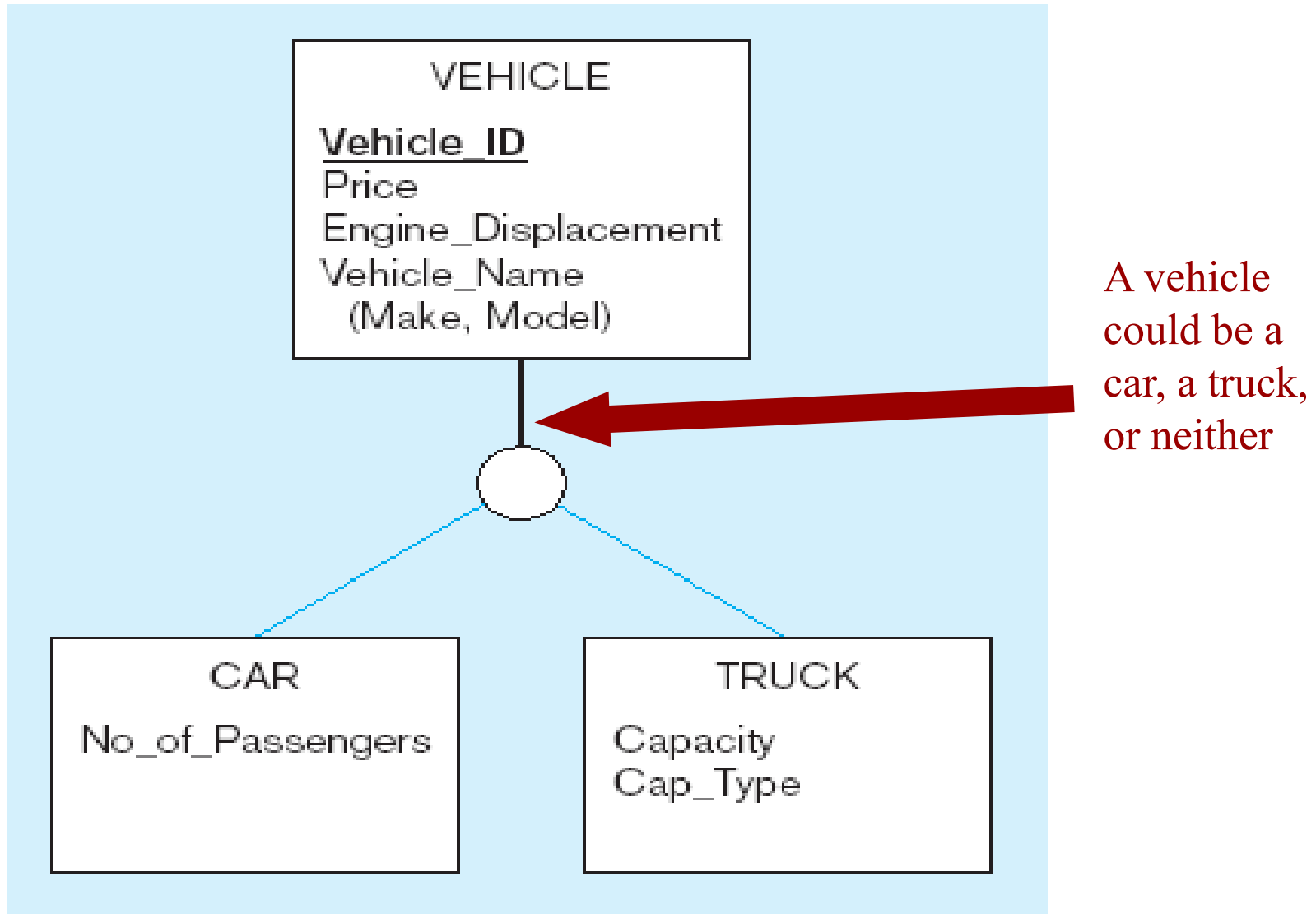


Figure 4-6 Examples of completeness constraints (cont.)

b) Partial specialization rule



Constraints in Supertype/ Disjointness constraint

- **Disjointness Constraints:** Whether an instance of a supertype may *simultaneously* be a member of two (or more) subtypes
 - Disjoint Rule: An instance of the supertype can be only ONE of the subtypes
 - Overlap Rule: An instance of the supertype could be more than one of the subtypes

Figure 4-7 Examples of disjointness constraints

a) Disjoint rule

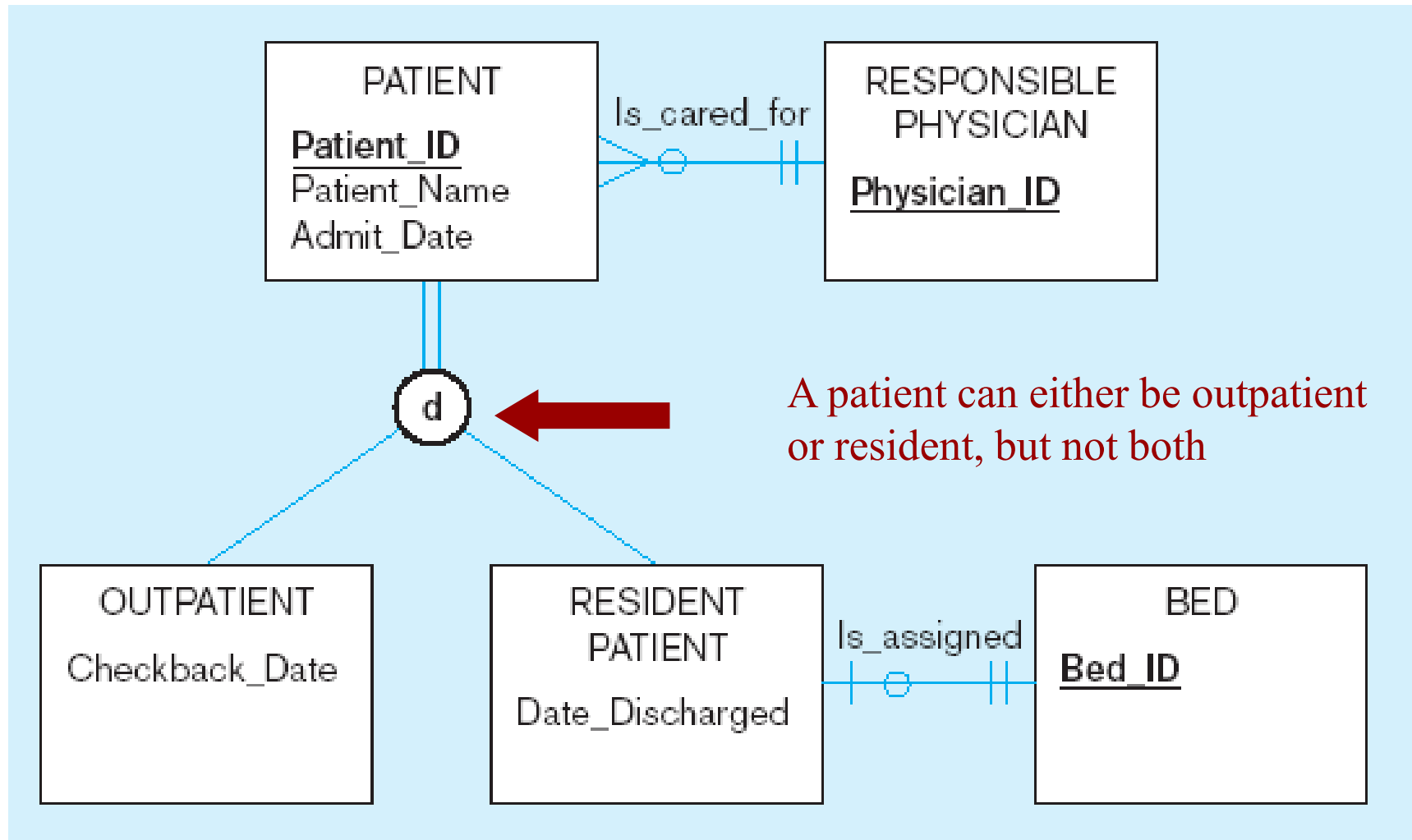


Figure 4-7 Examples of disjointness constraints (cont.)

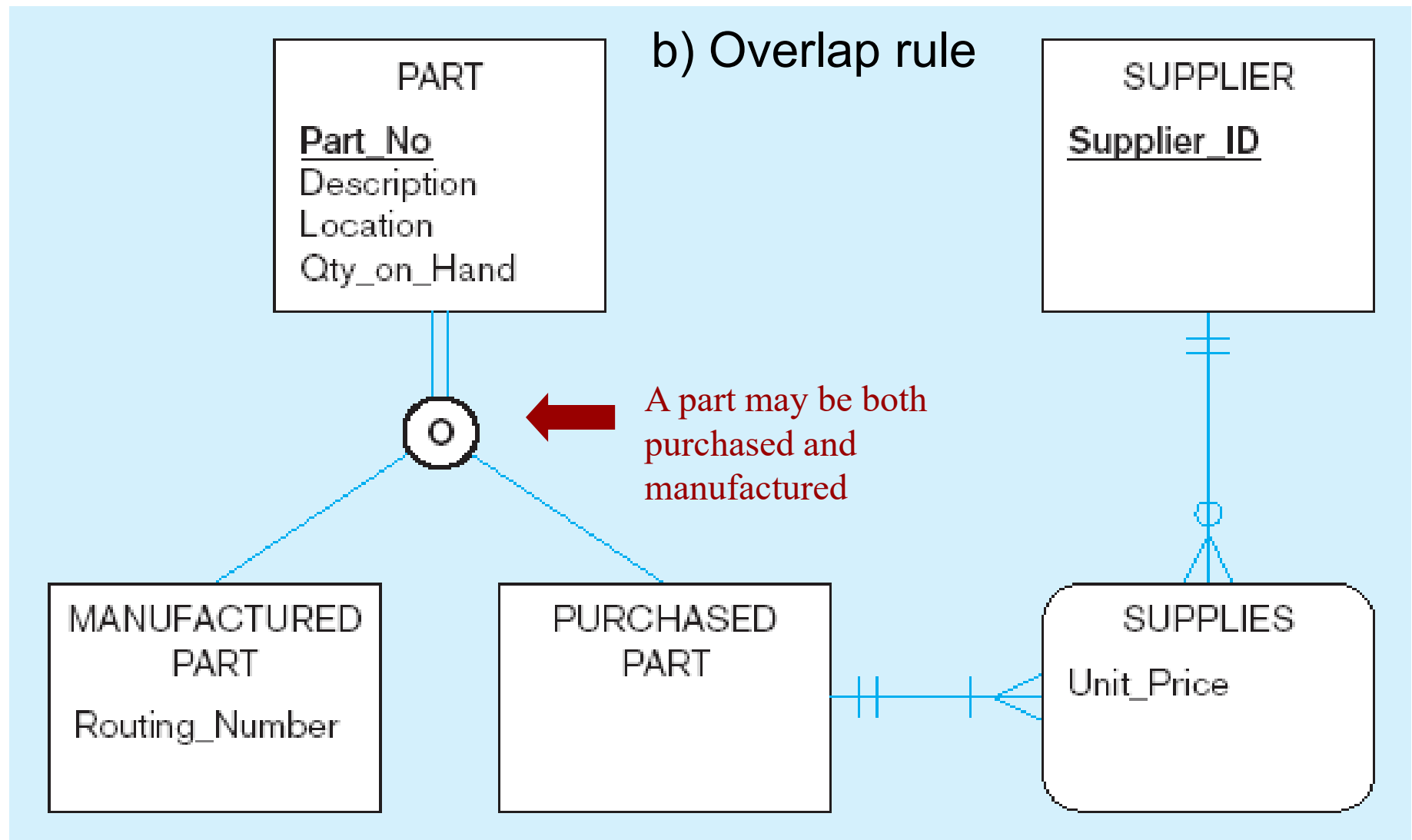
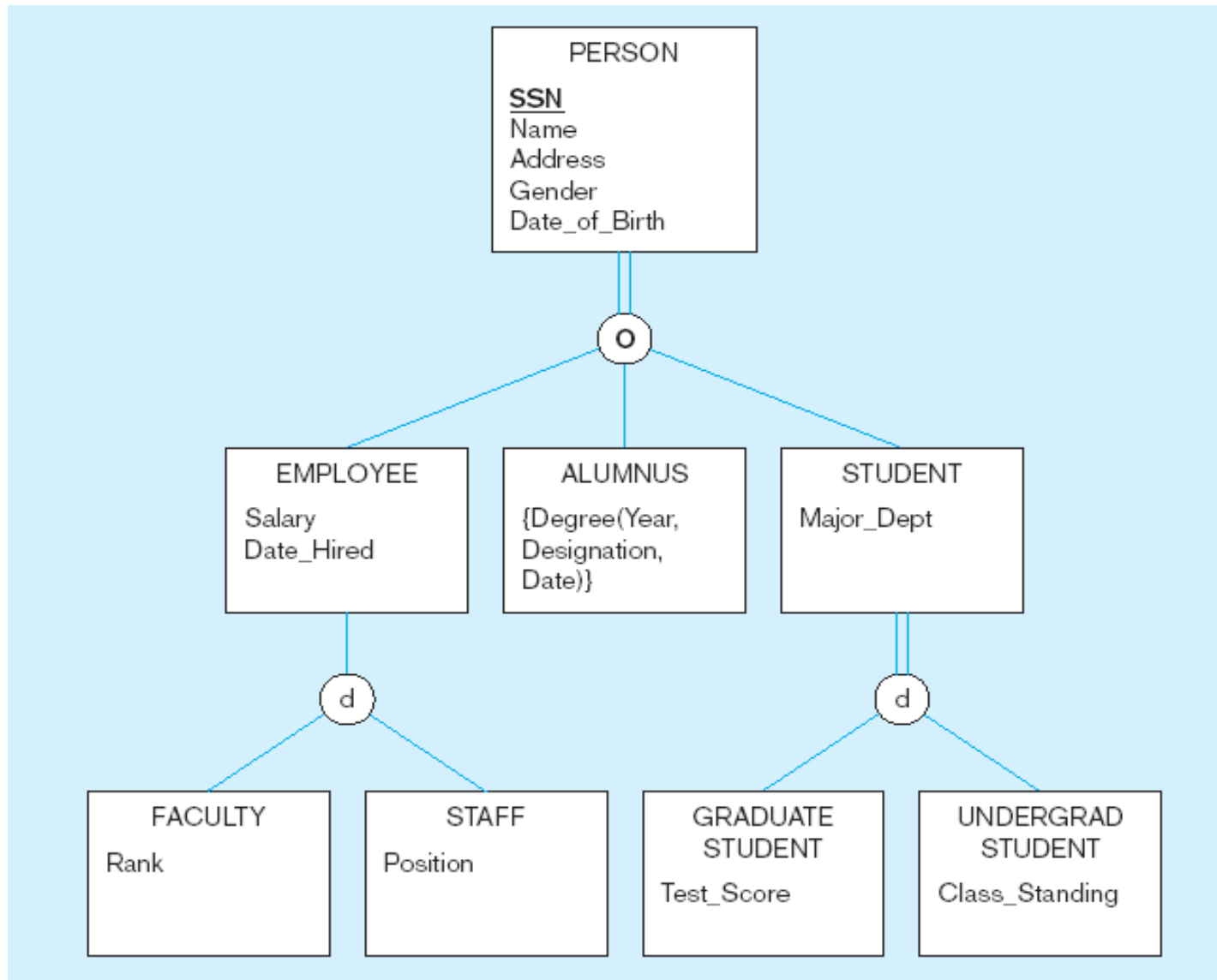


Figure 4-10 Example of supertype/subtype hierarchy



Entity Clusters

- EER diagrams are difficult to read when there are too many entities and relationships
- Solution: Group entities and relationships into *entity clusters*
- **Entity cluster:** Set of one or more entity types and associated relationships grouped into a single abstract entity type

Figure 4-13a
Possible entity
clusters for Pine
Valley Furniture in
Microsoft Visio

Related
groups of
entities could
become
clusters

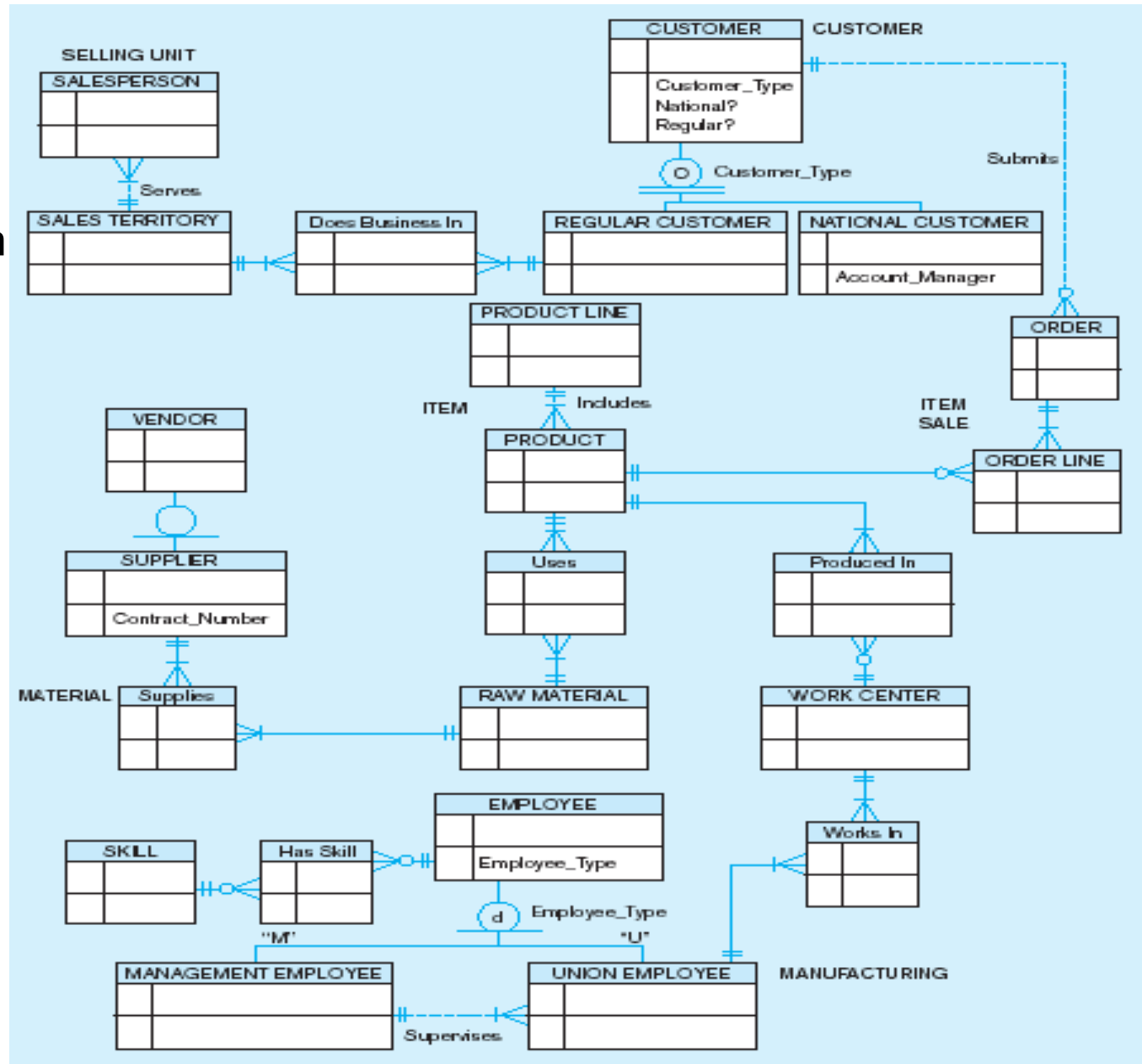
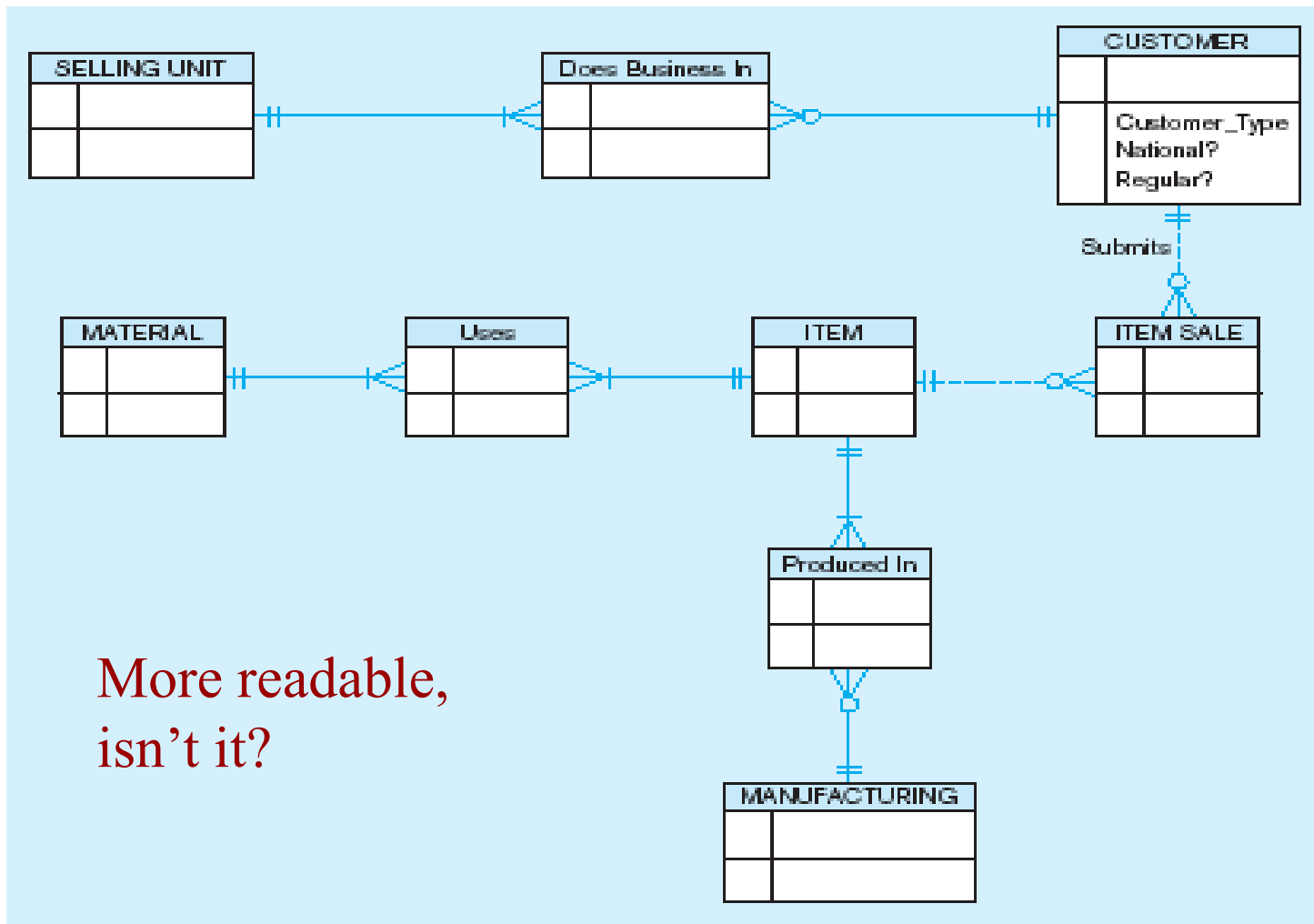
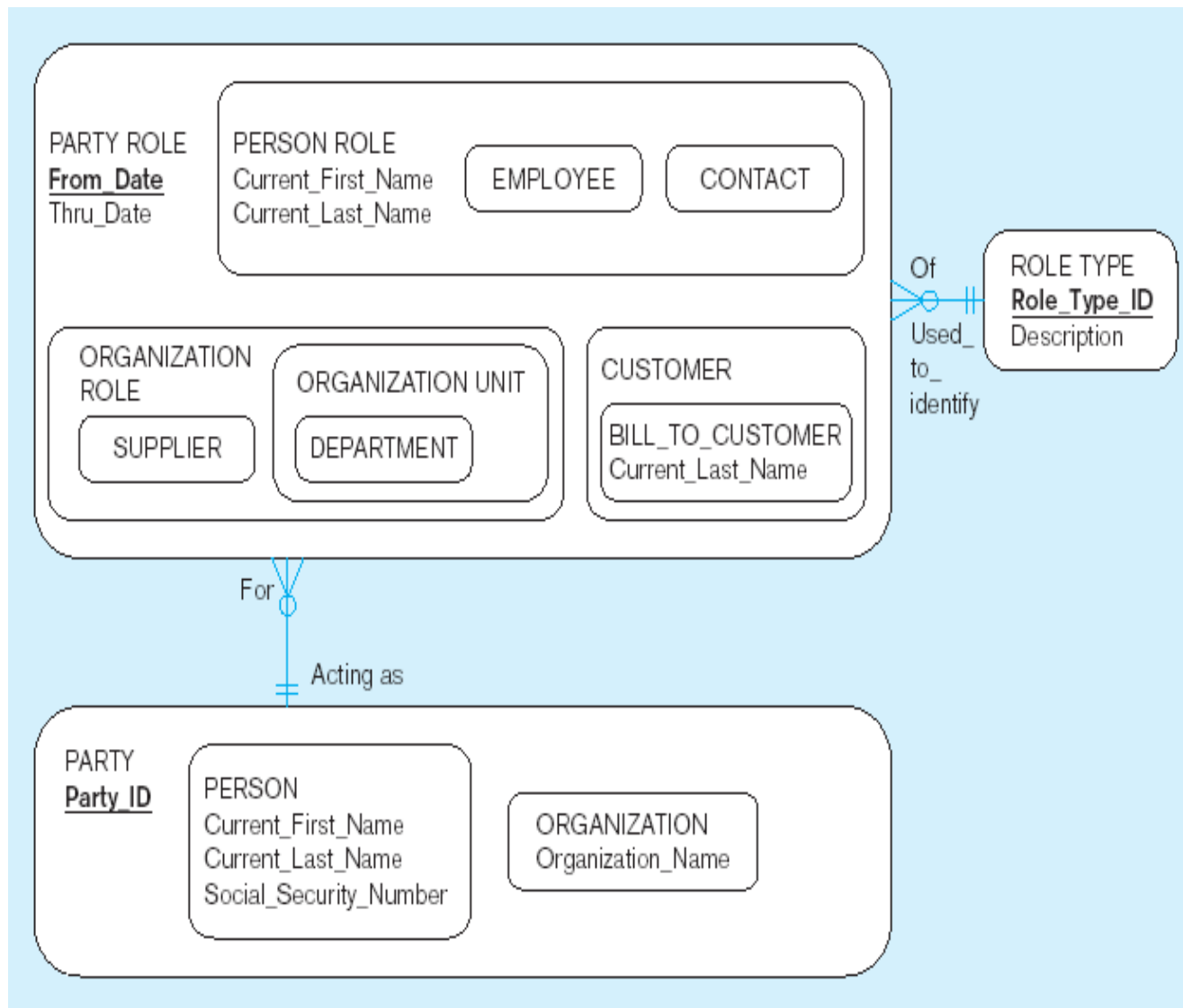


Figure 4-13b EER diagram of PVF entity clusters





Packaged data models provide generic models that can be customized for a particular organization's business rules

Business rules

- Statements that *define* or *constrain* some aspect of the business
- Classification of business rules:
 - Derivation—rule derived from other knowledge, often in the form of a formula using attribute values
 - Structural assertion—rule expressing static structure. Includes attributes, relationships, and definitions
 - Action assertion—rule expressing constraints/control of organizational actions

Figure 4-20 Business Rule 1: For a faculty member to be assigned to teach a section of a course, the faculty member must be qualified to teach the course for which that section is scheduled

In this case, the action assertion is a ***R***estriction

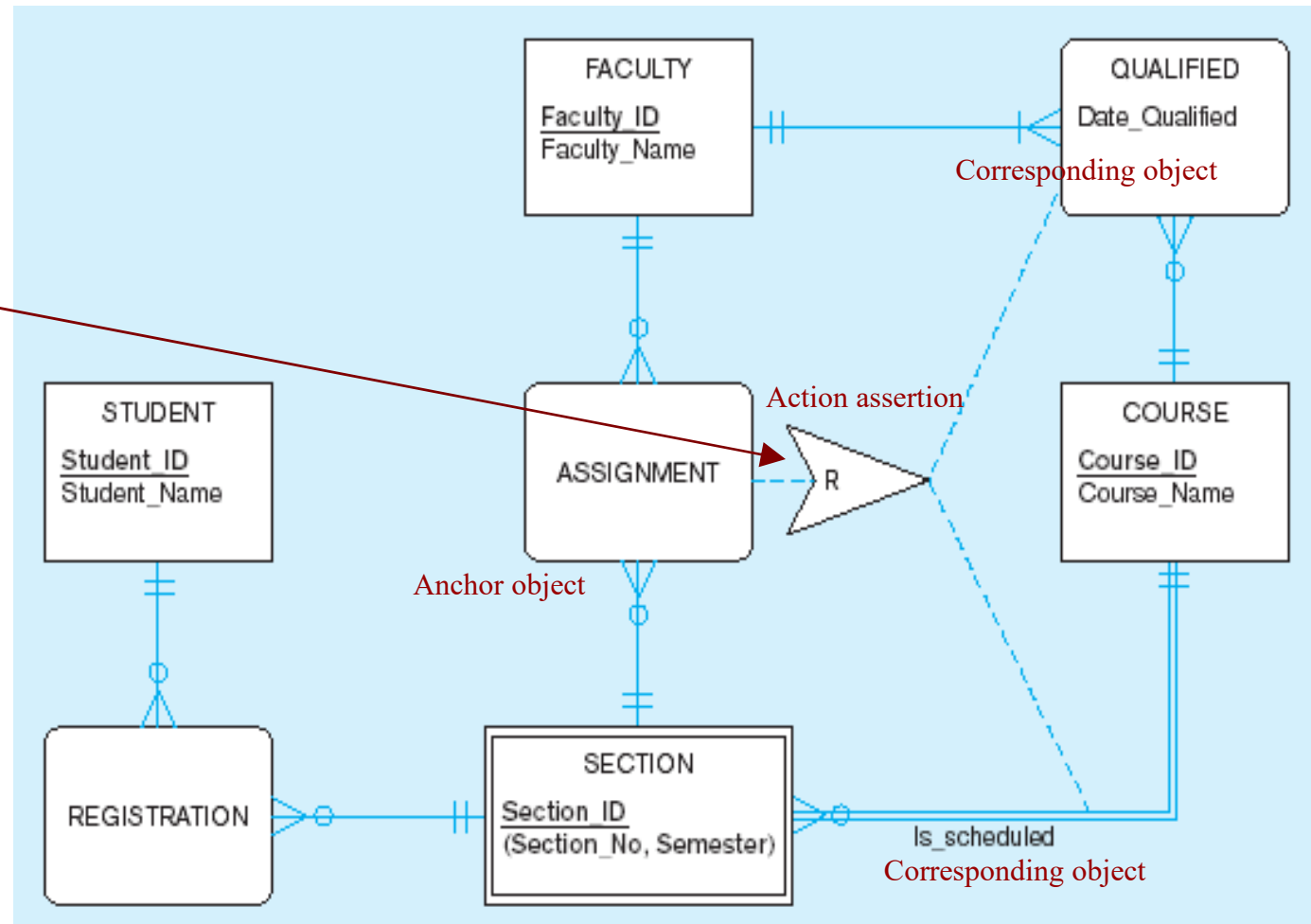
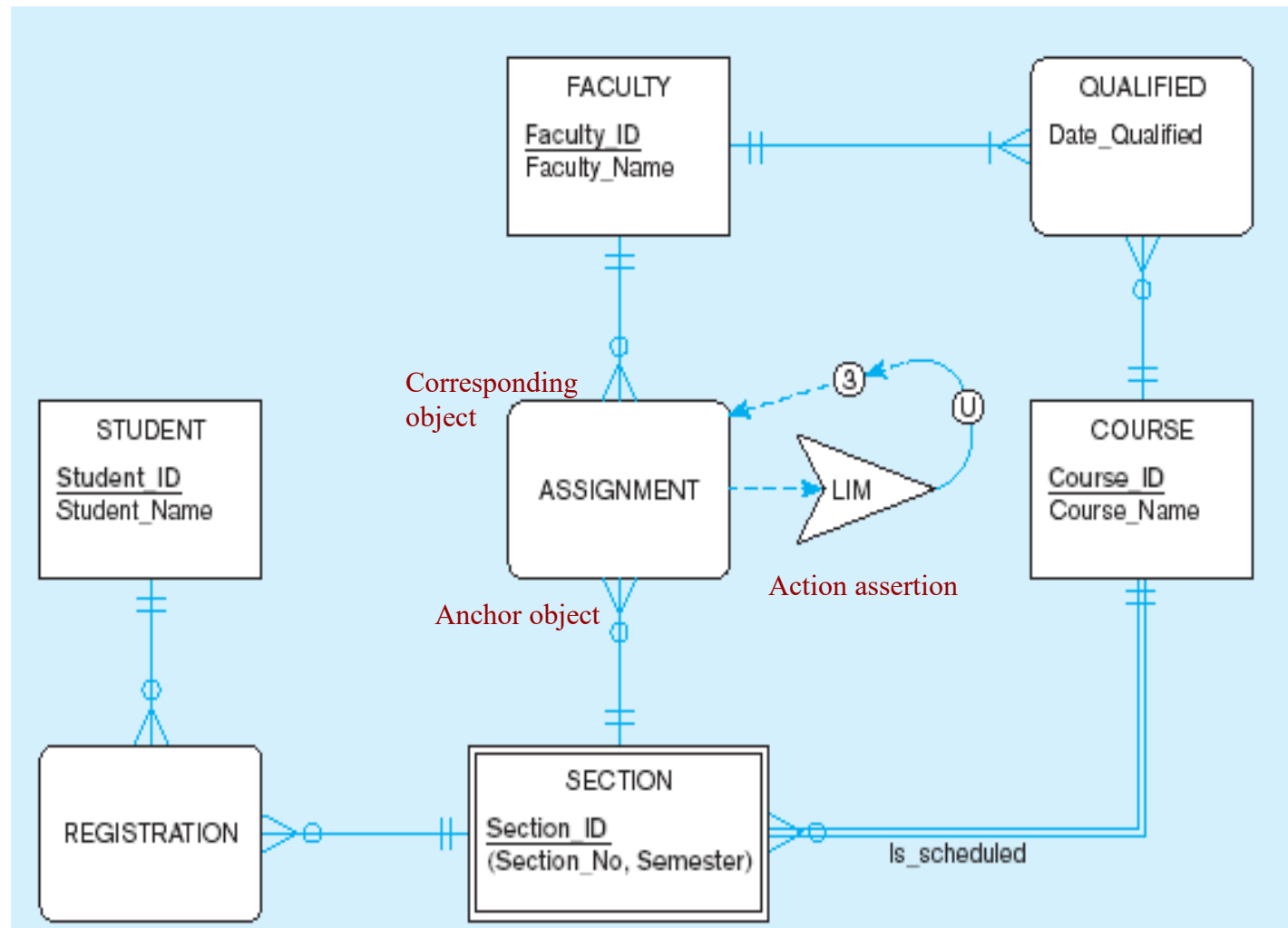


Figure 4-21 Business Rule 2: For a faculty member to be assigned to teach a section of a course, the faculty member must not be assigned to teach a total of more than three course sections

In this case, the
action assertion
is an
*Upper LIM*it



Introduction to Relational Model

- Data is logically structured within relations.
- Each relation is a table (file) with named columns (attributes, fields) and rows (records).

Properties of a Relation

- Simple attribute
 - No composite, no multivalued attribute
- Each relation must have a primary key:
 - Simple or composite key
 - May have other keys (candidate keys)
 - Key cannot be null
 - Cannot be duplicated

Integrity Constraints

- Domain constraints
- Entity integrity:
 - Primary key cannot be null, cannot be duplicated
- Referential integrity
- Other constraints

Relational Database Design

- Strong entity: Create a table that includes all simple attributes
 - Composite
- Weak entity: add owner primary key
- Multi-valued attribute: Create a table for each multi-valued attribute
 - Key + attribute
- Relationship:
 - 1:1, 1:M
 - Relationship table: for partial participation to avoid null
 - Foreign key
 - M:M: relationship table
 - N-ary relationship: relationship table
 - Recursive relationship
- Attribute of relationship
- Superclass and subclass

Online Shopping Cart

