

The Enhanced ER Model

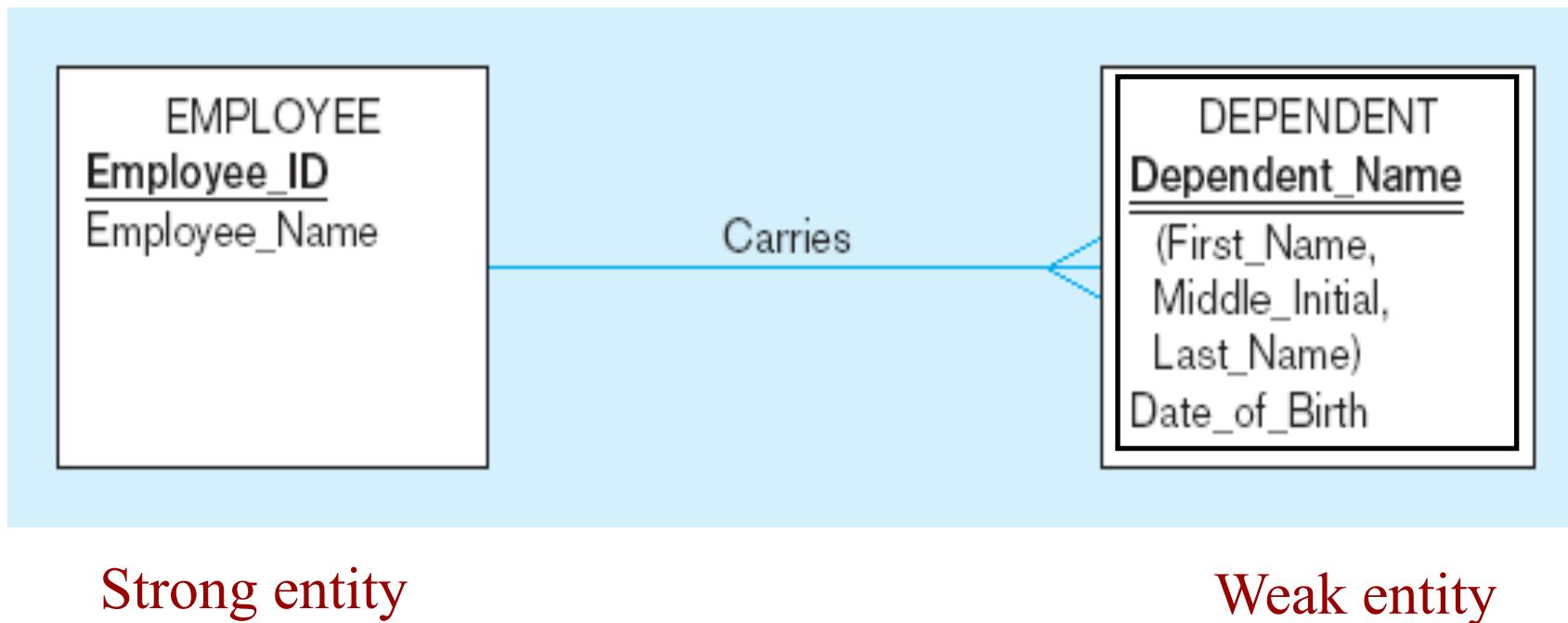
Agenda

- Strong vs Weak entity
- Associative entity
- Enhanced ER (EER)
- Cluster

Strong vs. Weak Entities, and Identifying Relationships

- **Strong entities**
 - exist independently of other types of entities
 - **has its own unique identifier**
 - *identifier underlined with single-line*
- **Weak entity**
 - dependent on a strong entity (identifying owner)...cannot exist on its own
 - **does not have a unique identifier** (only a partial identifier)
 - *Partial identifier underlined with double-line*
 - *Entity box has double line*
- **Identifying relationship**
 - links strong entities to weak entities

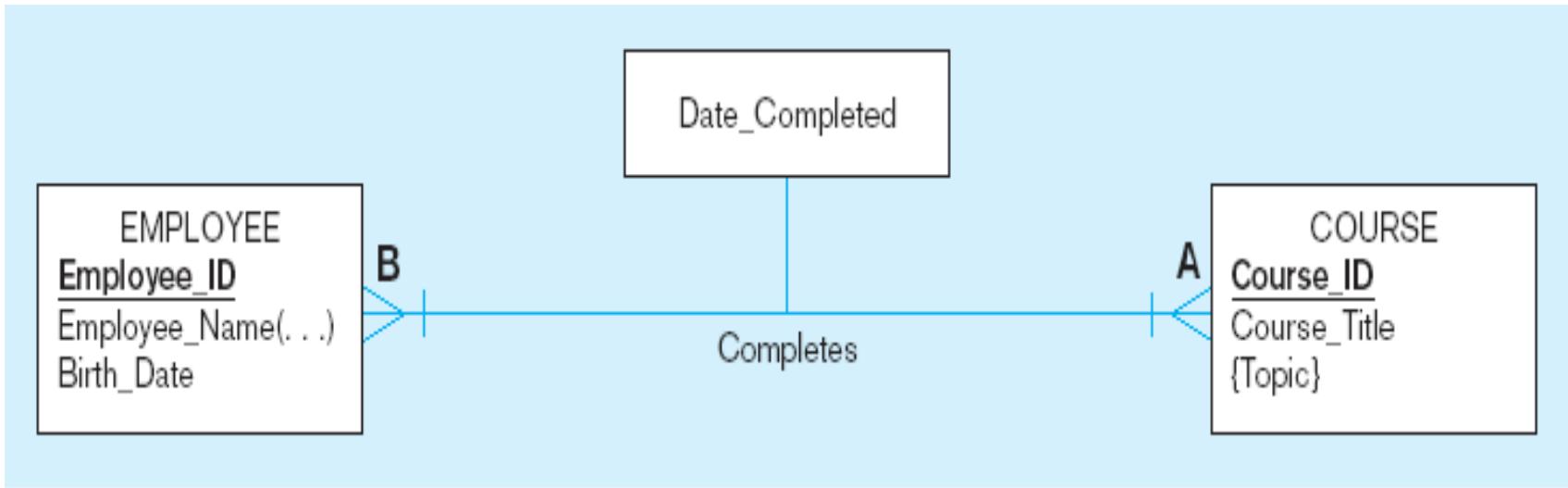
Identifying relationship



Associative Entities

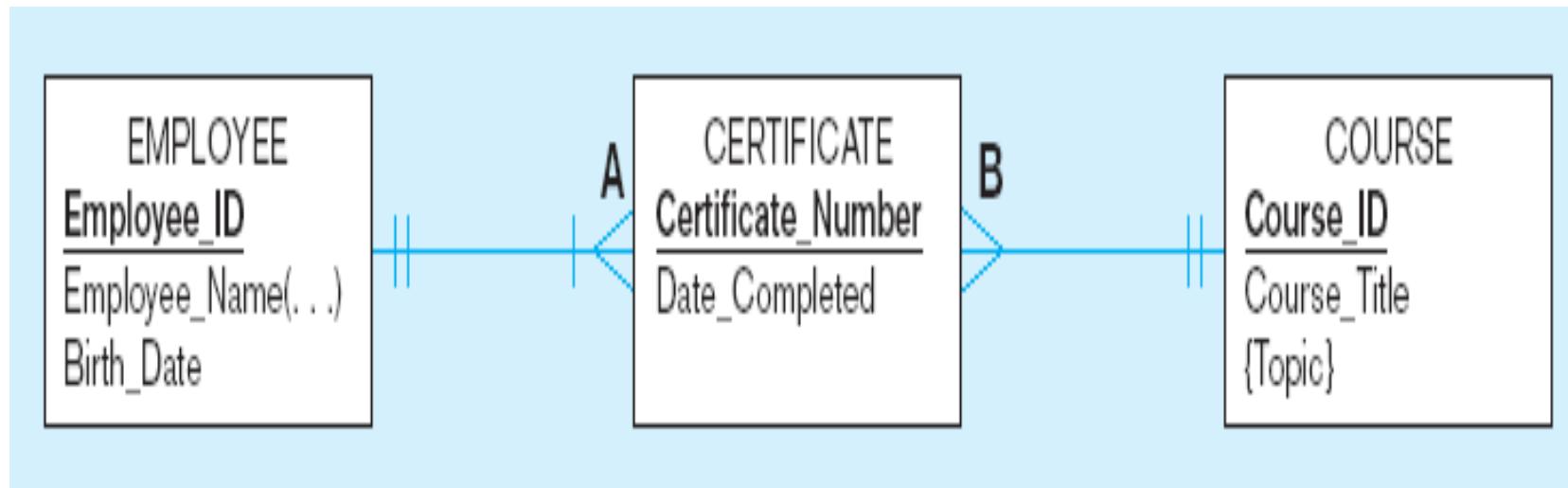
- An **entity**—has attributes
- A **relationship**—links entities together
- When should a *relationship with attributes* instead be an *associative entity*?
 - All relationships for the associative entity **should be many**
 - **Ternary relationships** should be converted to associative entities
 - The associative entity preferably has a **unique identifier**, and should also have other attributes
 - The associative entity **may participate** in other relationships other than the entities of the associated relationship

A binary relationship with an attribute



Here, the date completed attribute pertains specifically to the employee's completion of a course...it is an attribute of the *relationship*

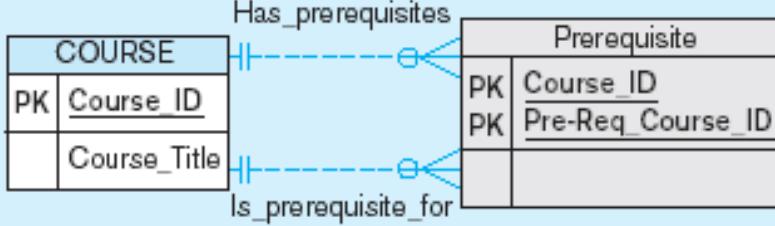
An associative entity (CERTIFICATE)

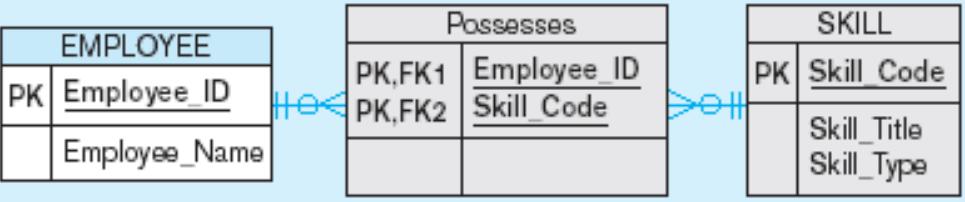


Associative entity is like a relationship with an attribute, but it is also considered to be an entity in its own right.

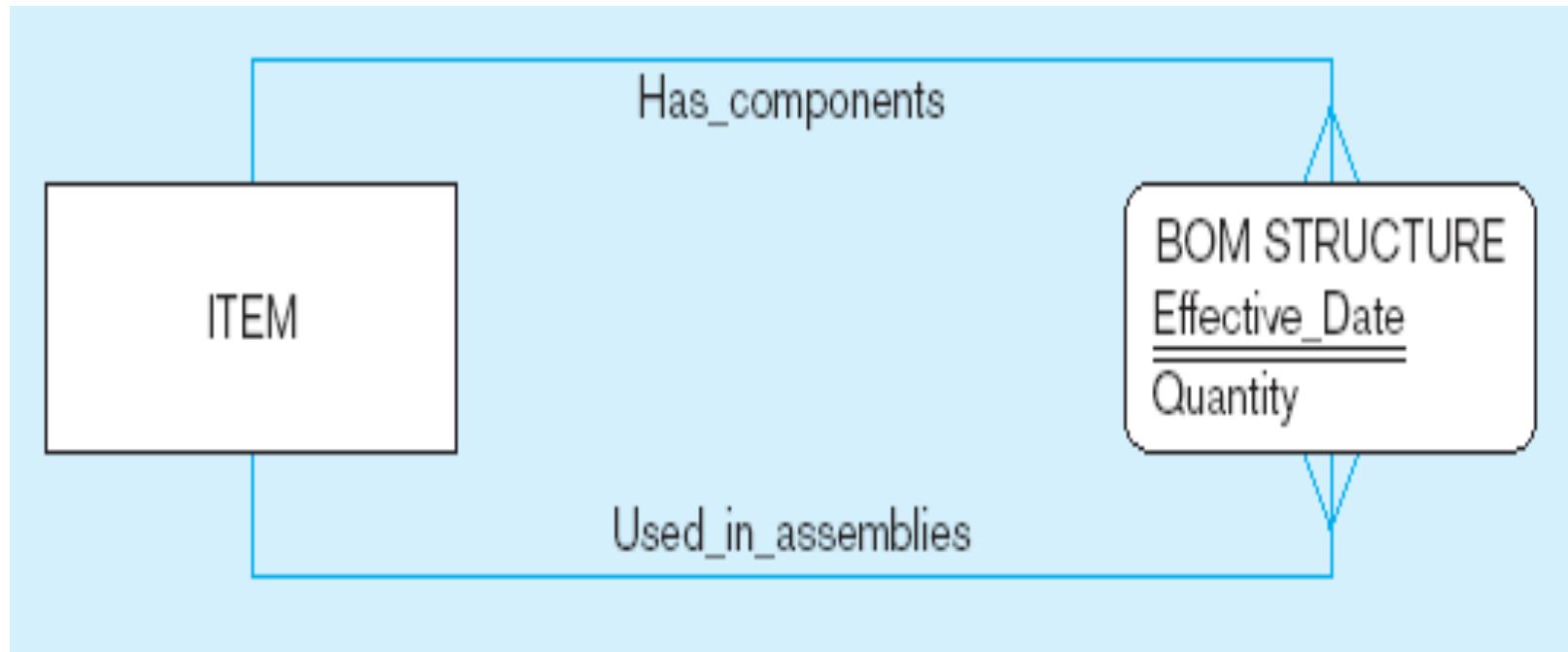
Note that the many-to-many cardinality between entities in Figure 3-11a has been replaced by two one-to-many relationships with the associative entity.

Multivalued attributes can be represented as relationships

ATTRIBUTE	RELATIONSHIP & ENTITY				
<p style="color: red;">simple</p> <table border="1" data-bbox="616 351 931 534"> <tr> <td>COURSE</td> </tr> <tr> <td>Course_ID</td> </tr> <tr> <td>Course_Title</td> </tr> <tr> <td>{Prerequisite}</td> </tr> </table>	COURSE	Course_ID	Course_Title	{Prerequisite}	 <p>The diagram illustrates a simple multivalued attribute representation. On the left, the COURSE entity is shown with attributes: Course_ID, Course_Title, and {Prerequisite}. On the right, the Prerequisite relationship is represented by a relationship line connecting the COURSE entity to itself. This relationship is named Has_prerequisites. The relationship line has two ends, each with a diamond symbol, indicating a many-to-many relationship. The Prerequisite entity contains attributes: Course_ID (PK) and Pre-Req_Course_ID (PK).</p>
COURSE					
Course_ID					
Course_Title					
{Prerequisite}					

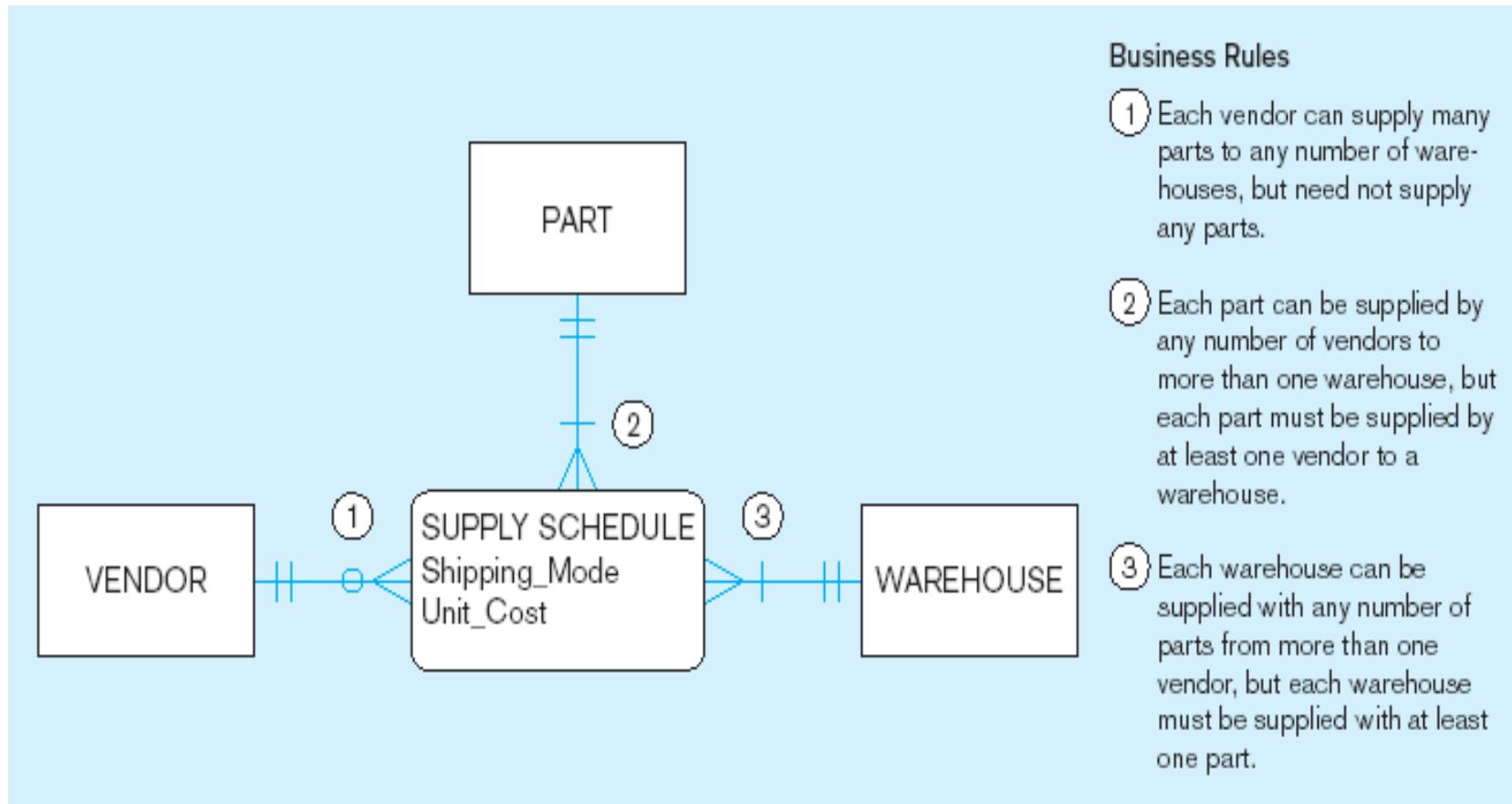
<p style="color: red;">composite</p> <table border="1" data-bbox="616 908 931 1110"> <tr> <td>EMPLOYEE</td> </tr> <tr> <td>Employee_ID</td> </tr> <tr> <td>Employee_Name</td> </tr> <tr> <td>{Skill (Skill_Code, Skill_Title, Skill_Type)}</td> </tr> </table>	EMPLOYEE	Employee_ID	Employee_Name	{Skill (Skill_Code, Skill_Title, Skill_Type)}	 <p>The diagram illustrates a composite multivalued attribute representation. On the left, the EMPLOYEE entity is shown with attributes: Employee_ID, Employee_Name, and {Skill (Skill_Code, Skill_Title, Skill_Type)}. On the right, the Possesses relationship is represented by a relationship line connecting the EMPLOYEE entity to the SKILL entity. This relationship is named Possesses. The relationship line has two ends, each with a diamond symbol. The Possesses entity contains attributes: PK,FK1 (Employee_ID), PK,FK2 (Skill_Code). The SKILL entity contains attributes: Skill_Code (PK), Skill_Title, and Skill_Type.</p>
EMPLOYEE					
Employee_ID					
Employee_Name					
{Skill (Skill_Code, Skill_Title, Skill_Type)}					

An associative entity – bill of materials structure

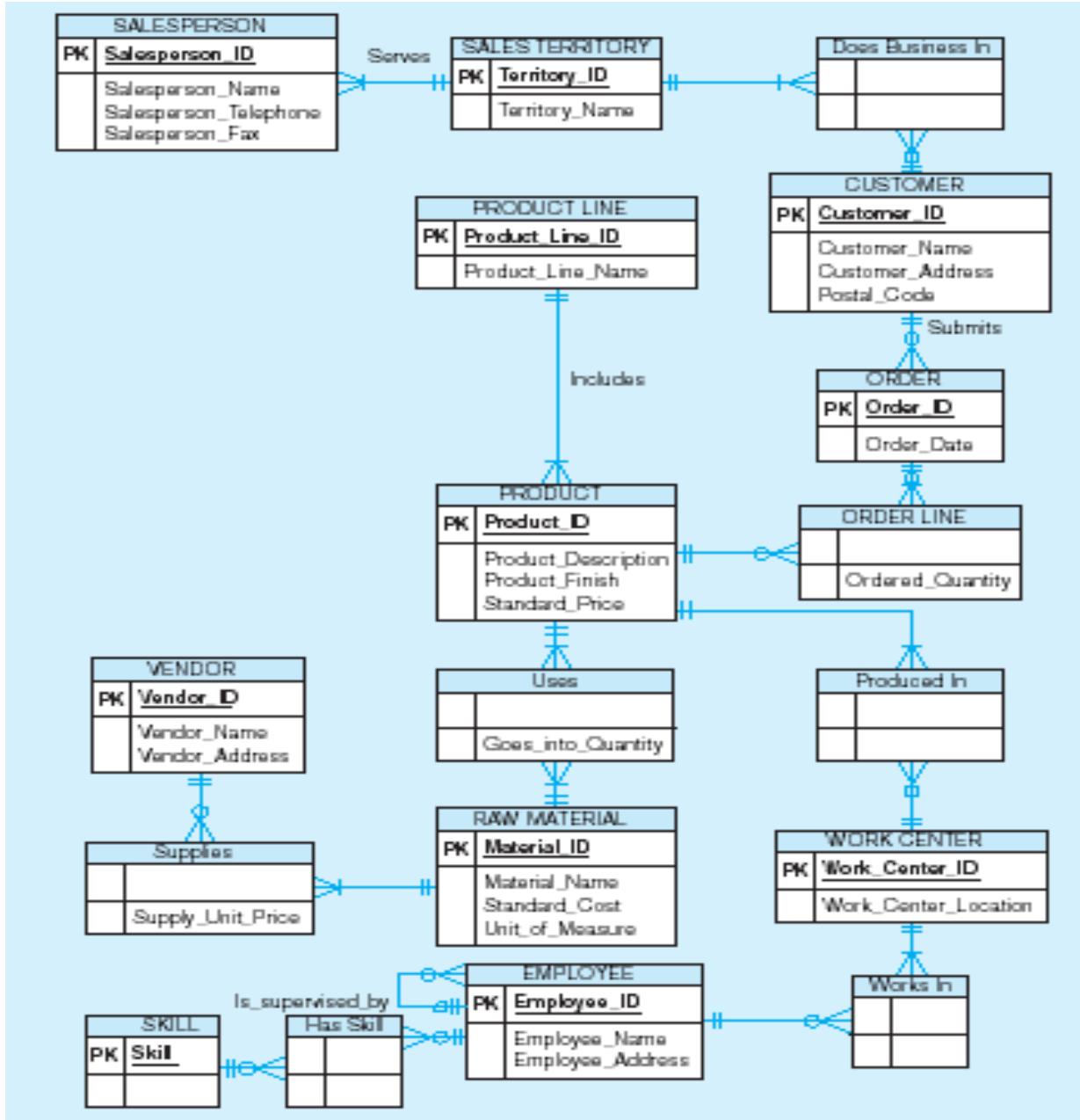


This could just be a relationship with attributes...it's a judgment call

Ternary relationship as an associative entity

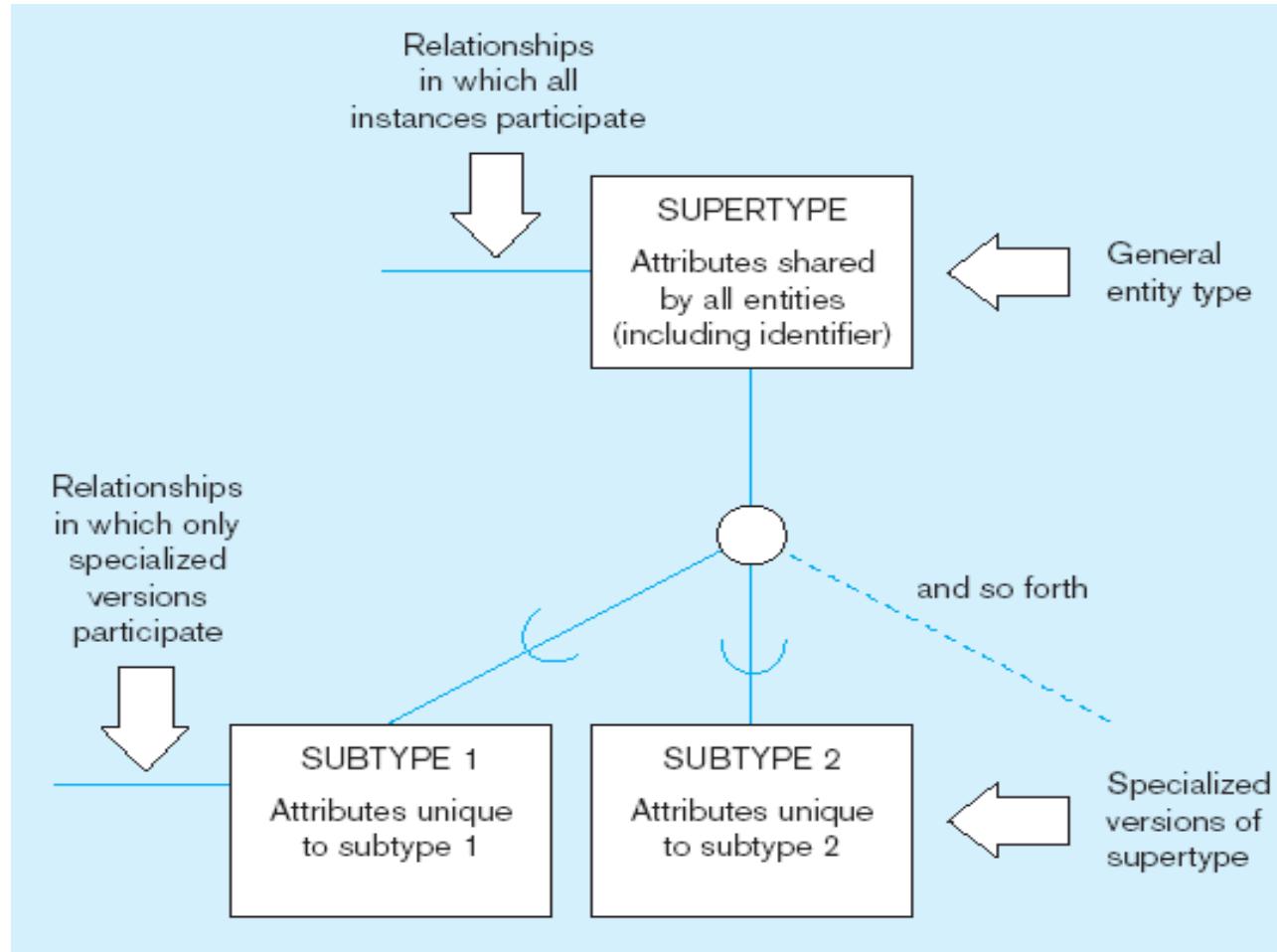


Microsoft Visio Notation for Pine Valley Furniture E-R diagram



Different modeling software tools may have different notation for the same constructs

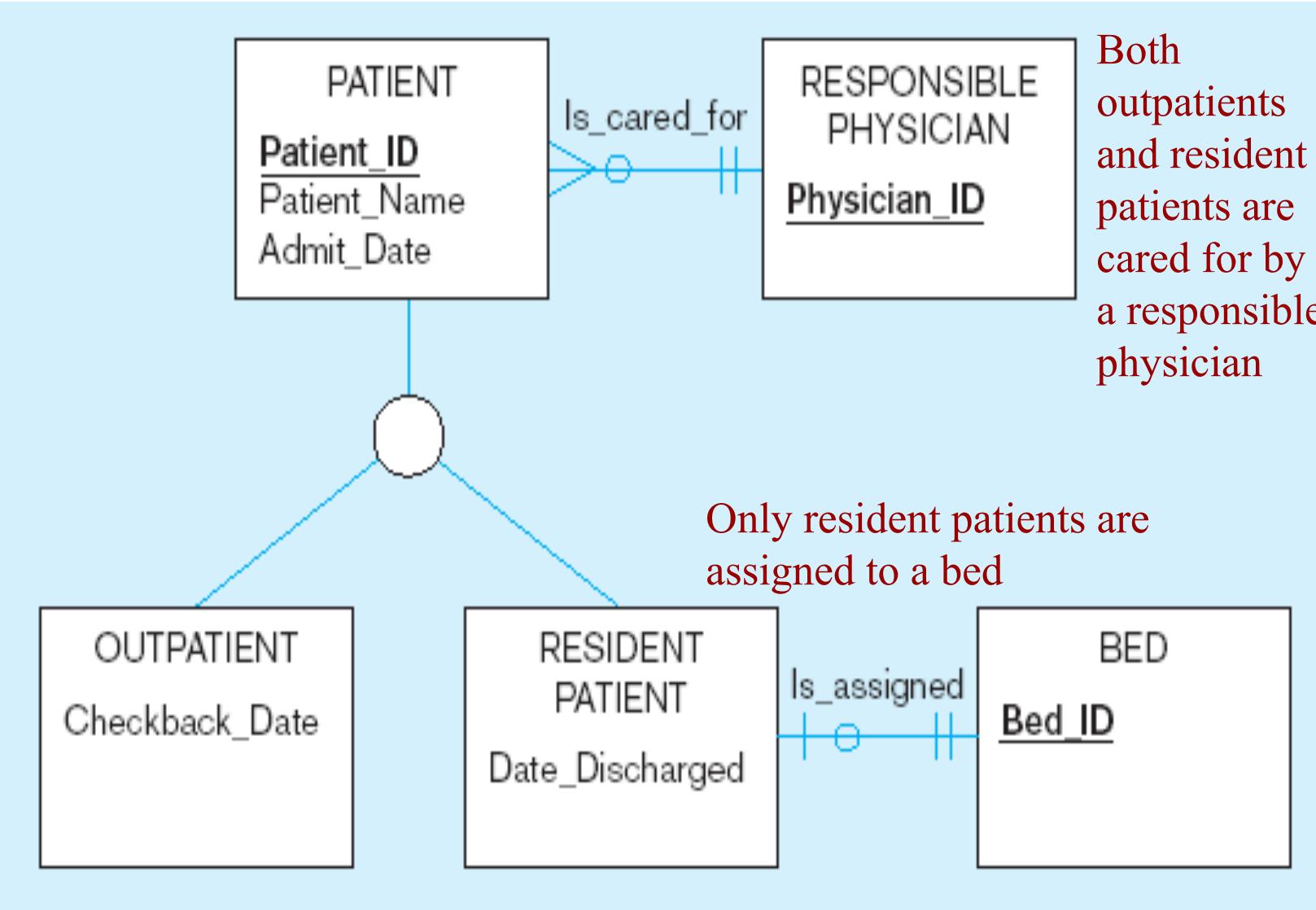
Basic notation for supertype/subtype notation



Supertypes and Subtypes

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

Supertype/subtype relationships in a hospital

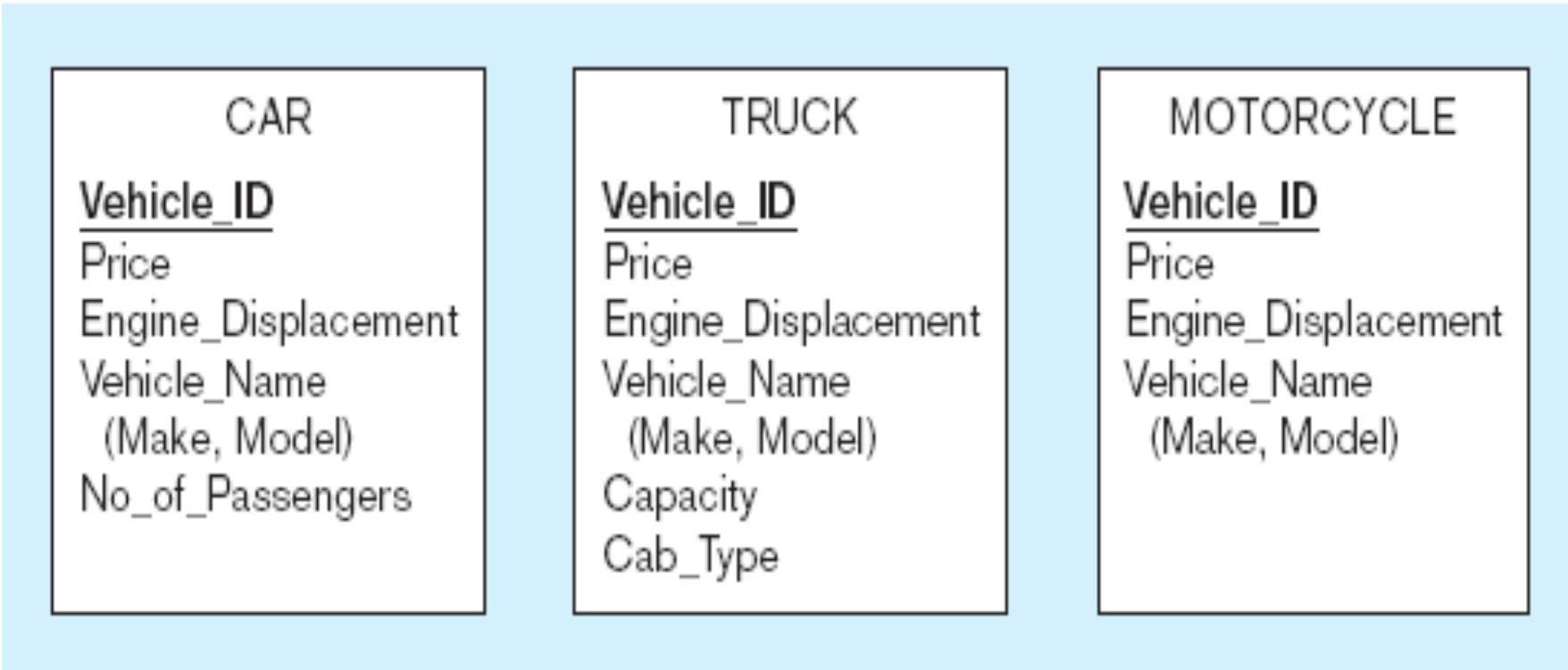


Generalization and Specialization

- ***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

Example of generalization

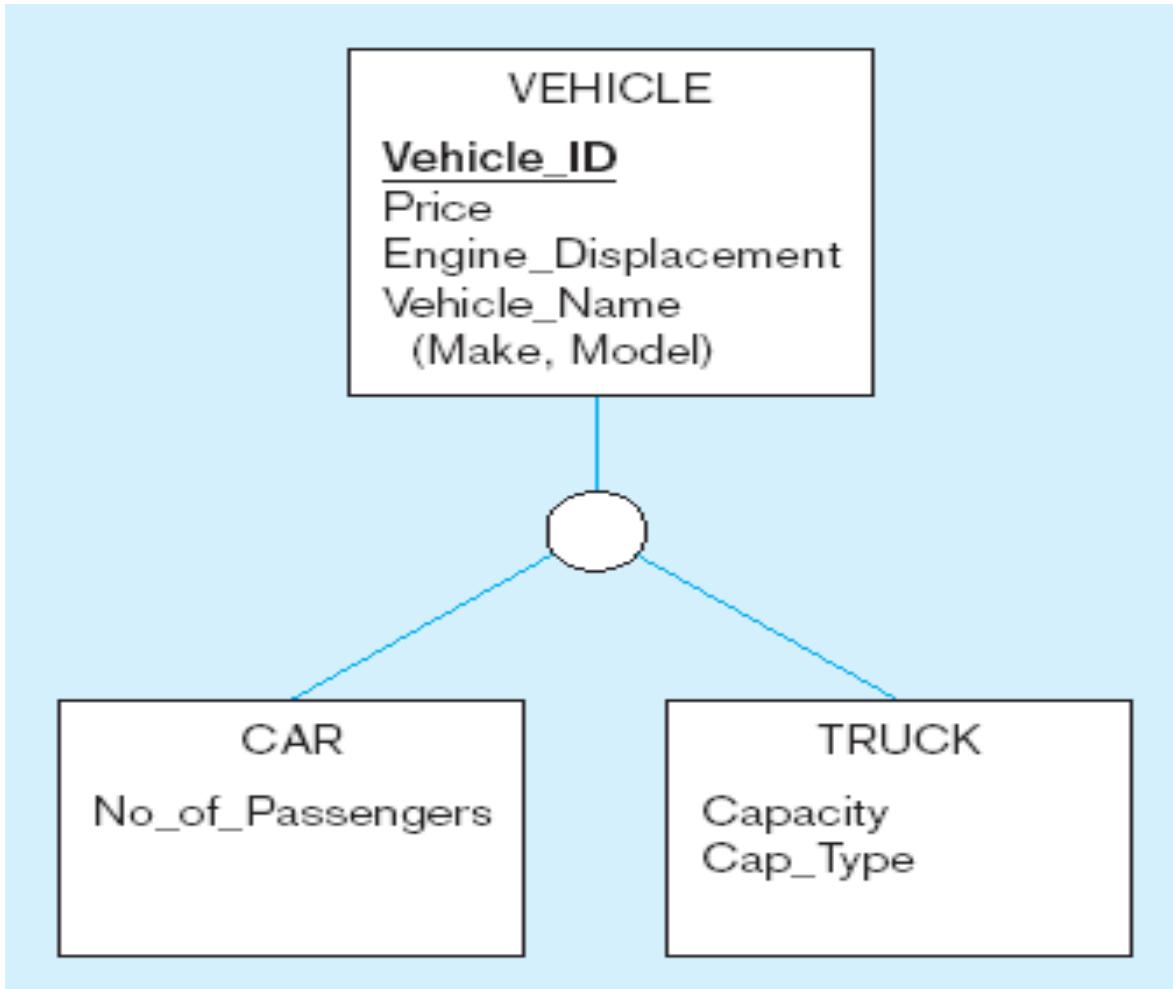
a) Three entity types: CAR, TRUCK, and MOTORCYCLE



All these types of vehicles have common attributes

Example of generalization (cont.)

b) Generalization to VEHICLE supertype

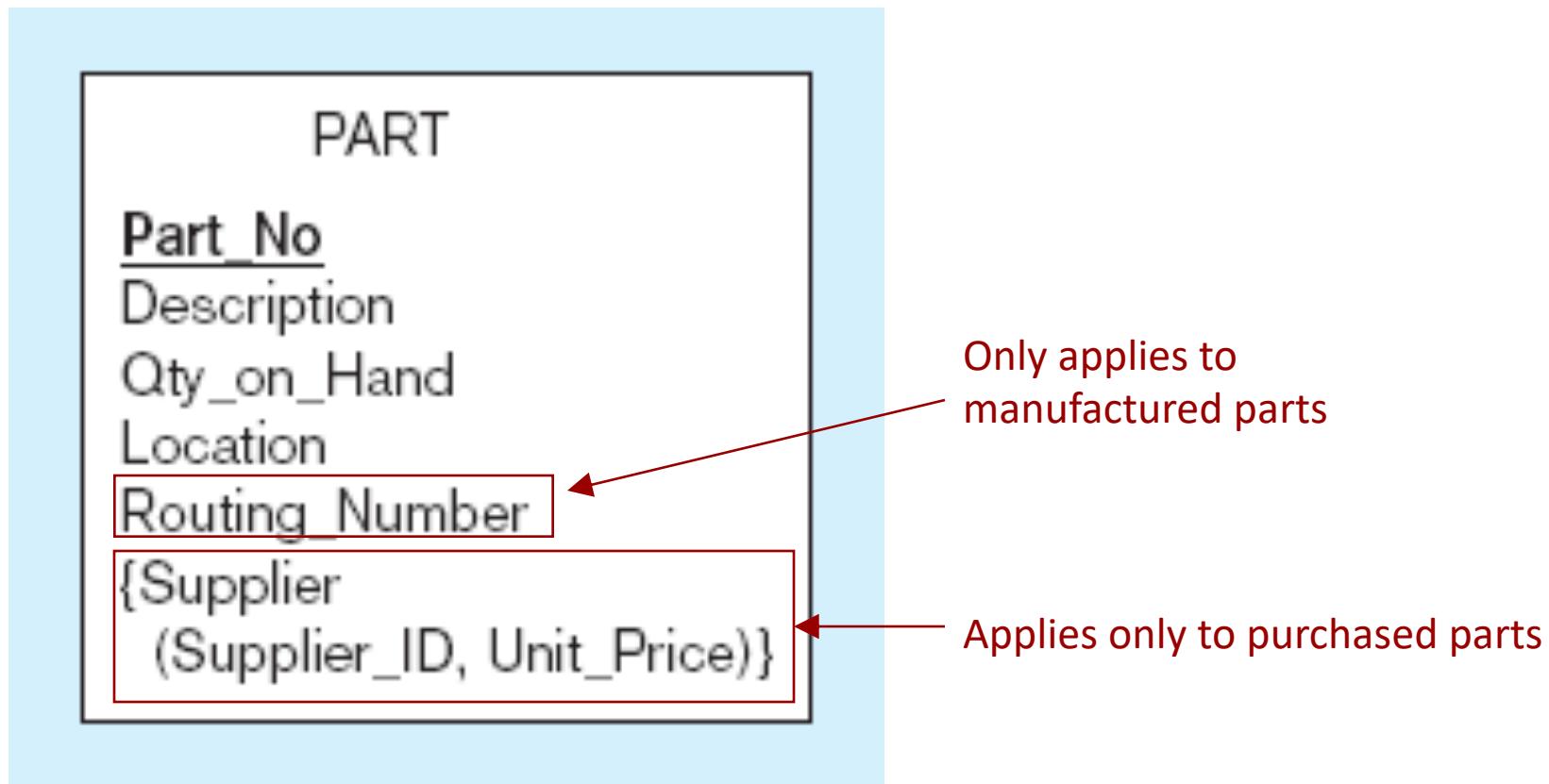


So we put
the shared
attributes in
a supertype

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

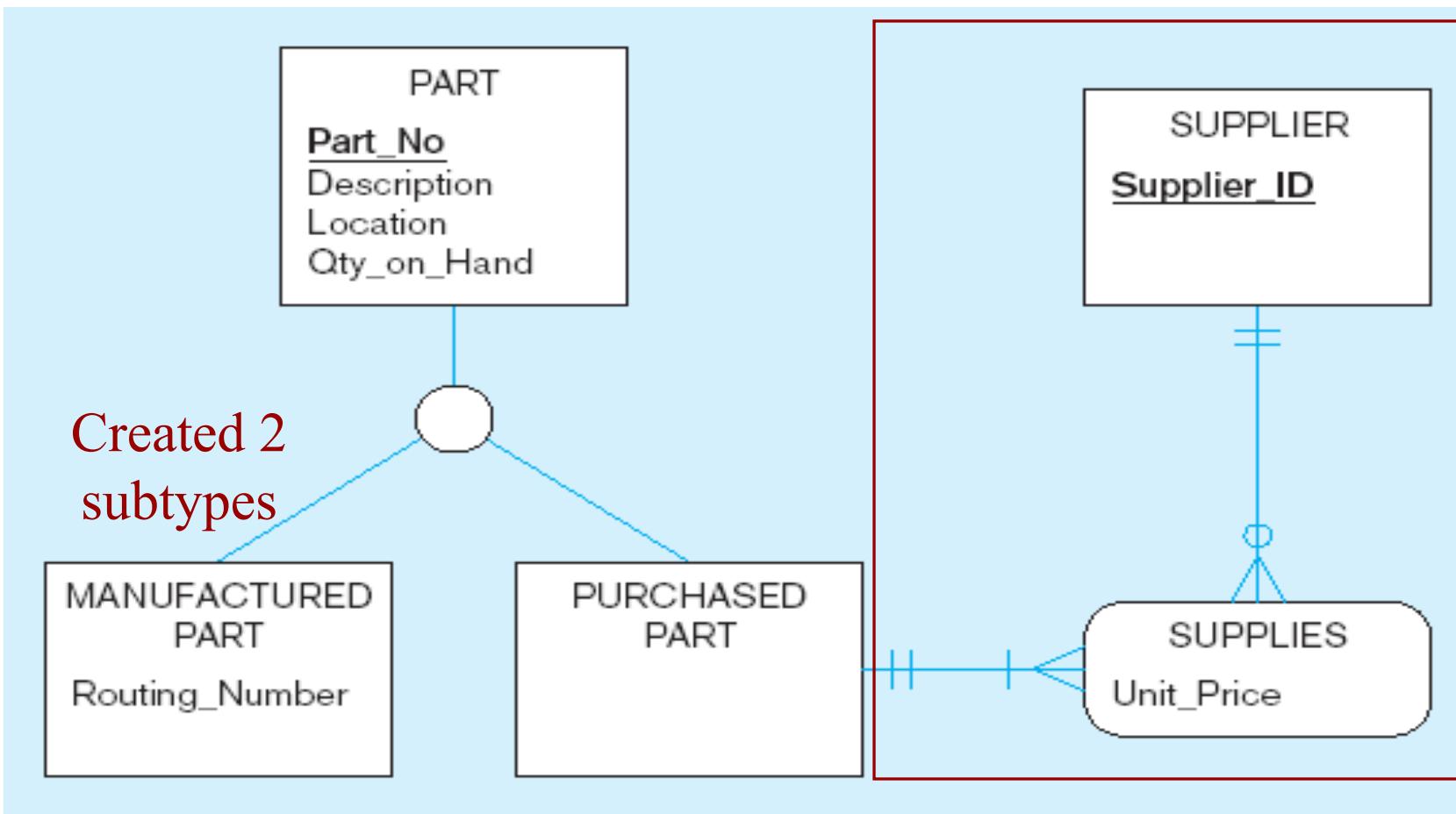
Example of specialization

a) Entity type PART



Example of specialization (cont.)

b) Specialization to MANUFACTURED PART and PURCHASED PART



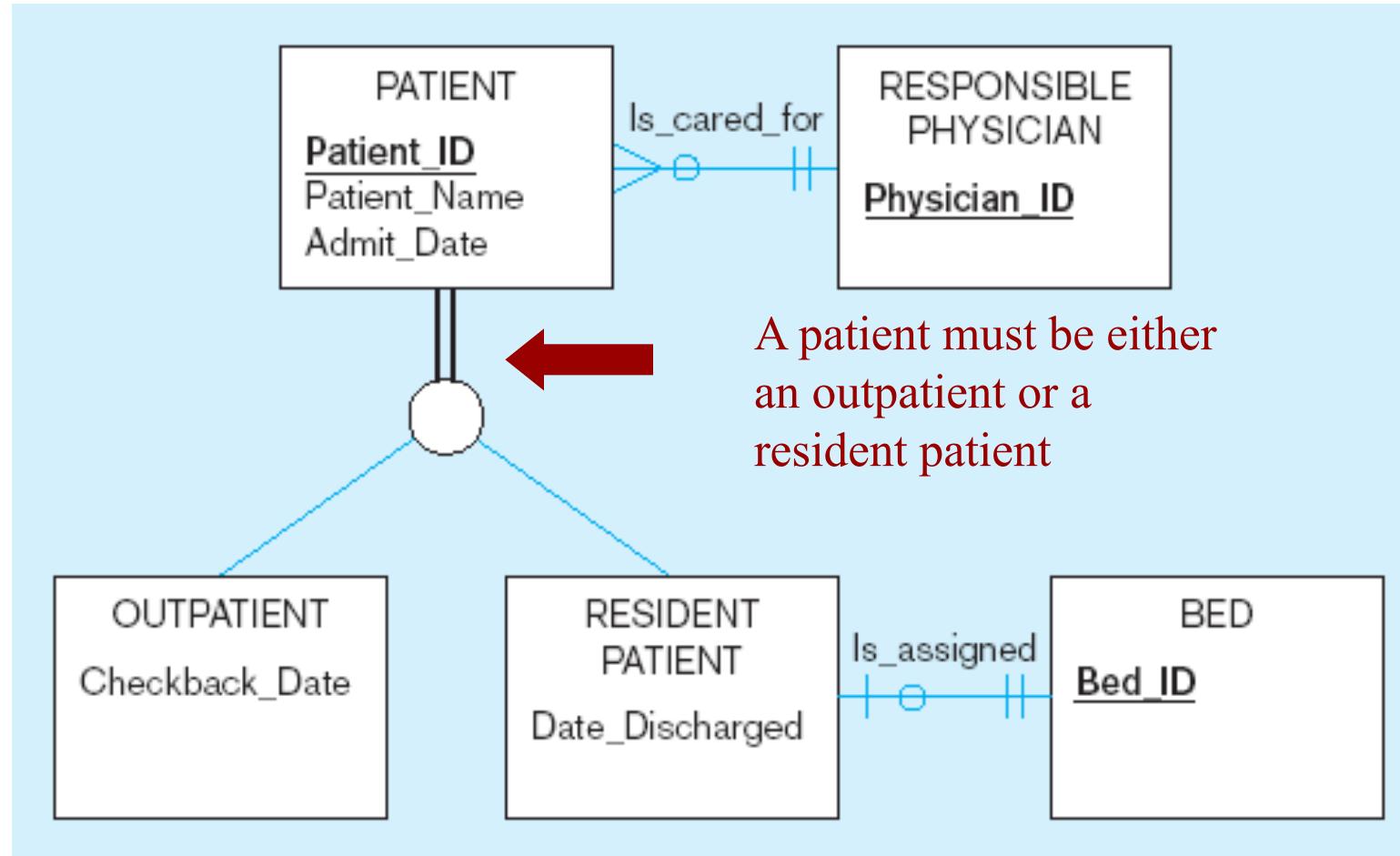
Note: multivalued attribute was replaced by an associative entity relationship to another entity

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)

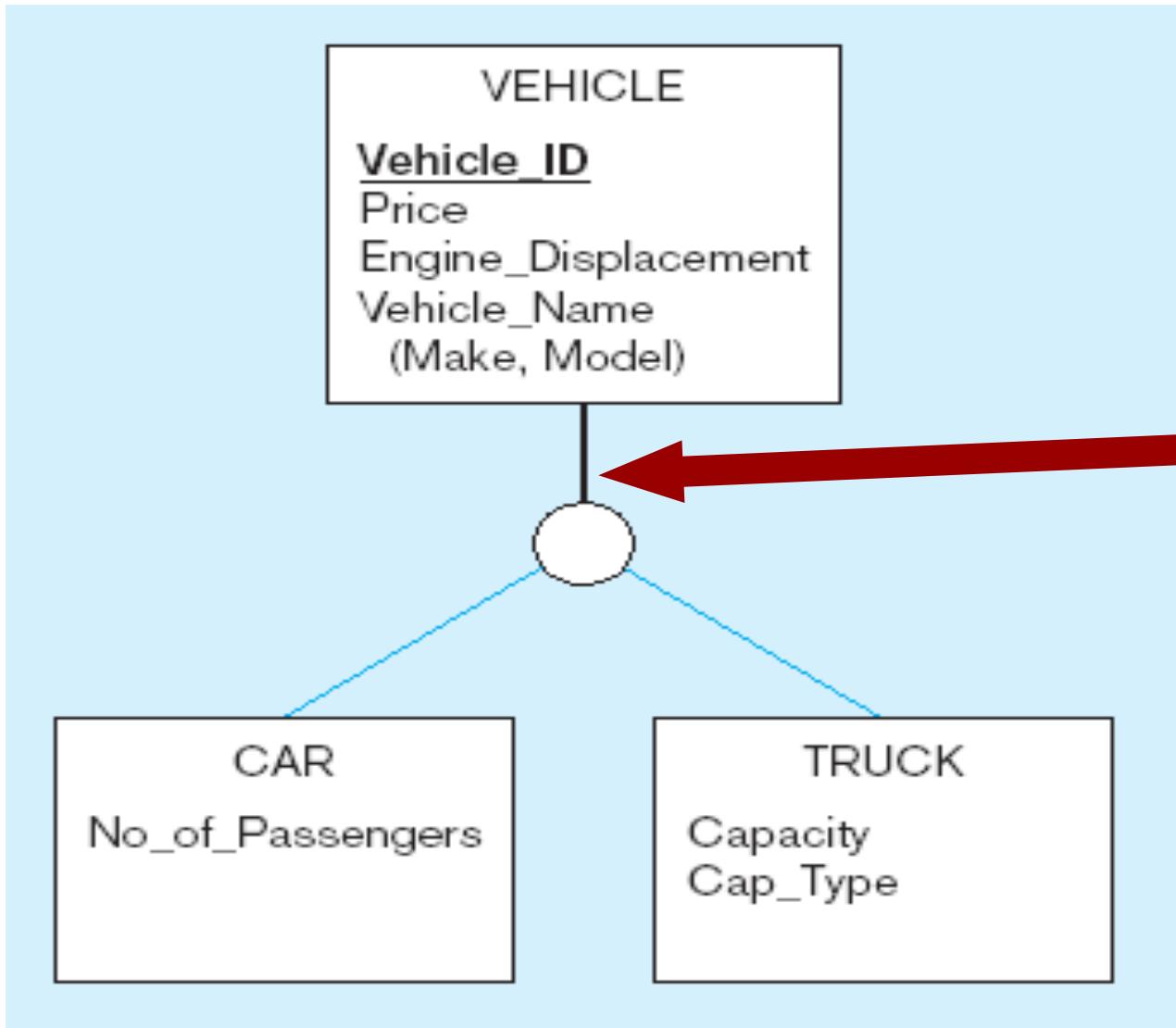
Examples of completeness constraints

a) Total specialization rule



Examples of completeness constraints (cont.)

b) Partial specialization rule



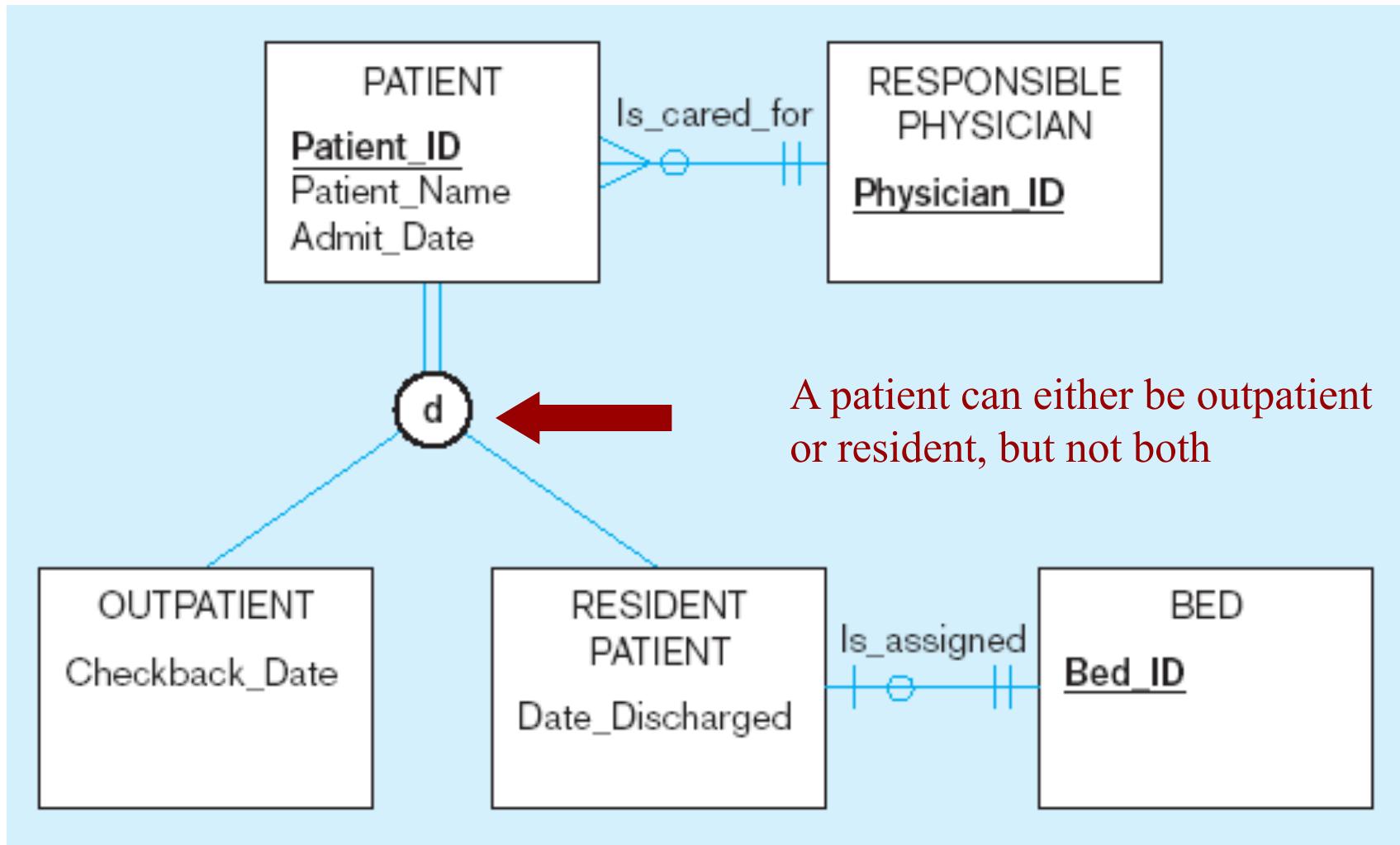
A vehicle
could be a
car, a truck,
or neither

Constraints in Supertype/ Disjointness constraint

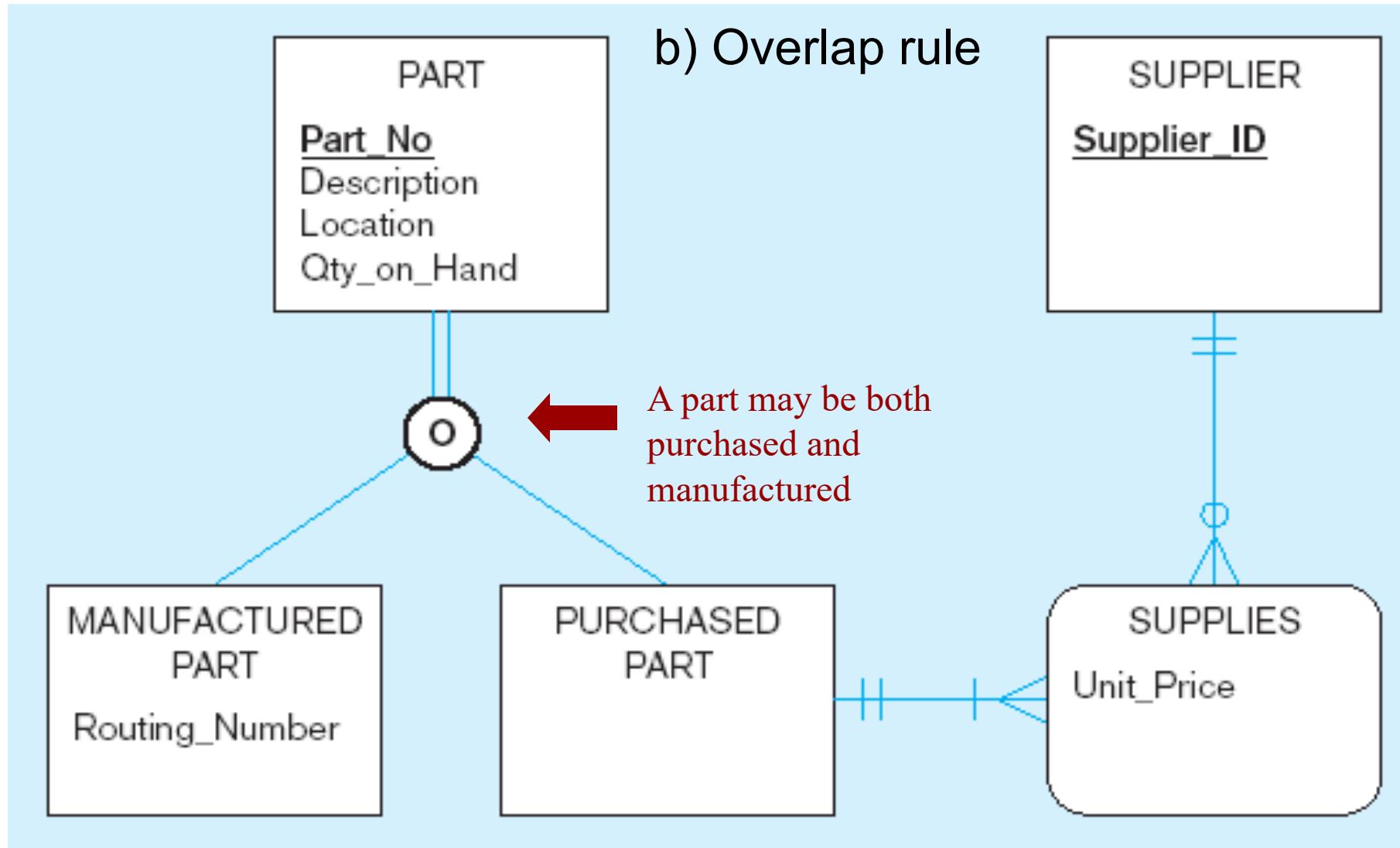
- **Disjointness Constraints**: Whether an instance of a supertype may *simultaneously* be a member of two (or more) subtypes
 - **Disjoint Rule**: An instance of the supertype can be only ONE of the subtypes
 - **Overlap Rule**: An instance of the supertype could be more than one of the subtypes

Figure 4-7 Examples of disjointness constraints

a) Disjoint rule



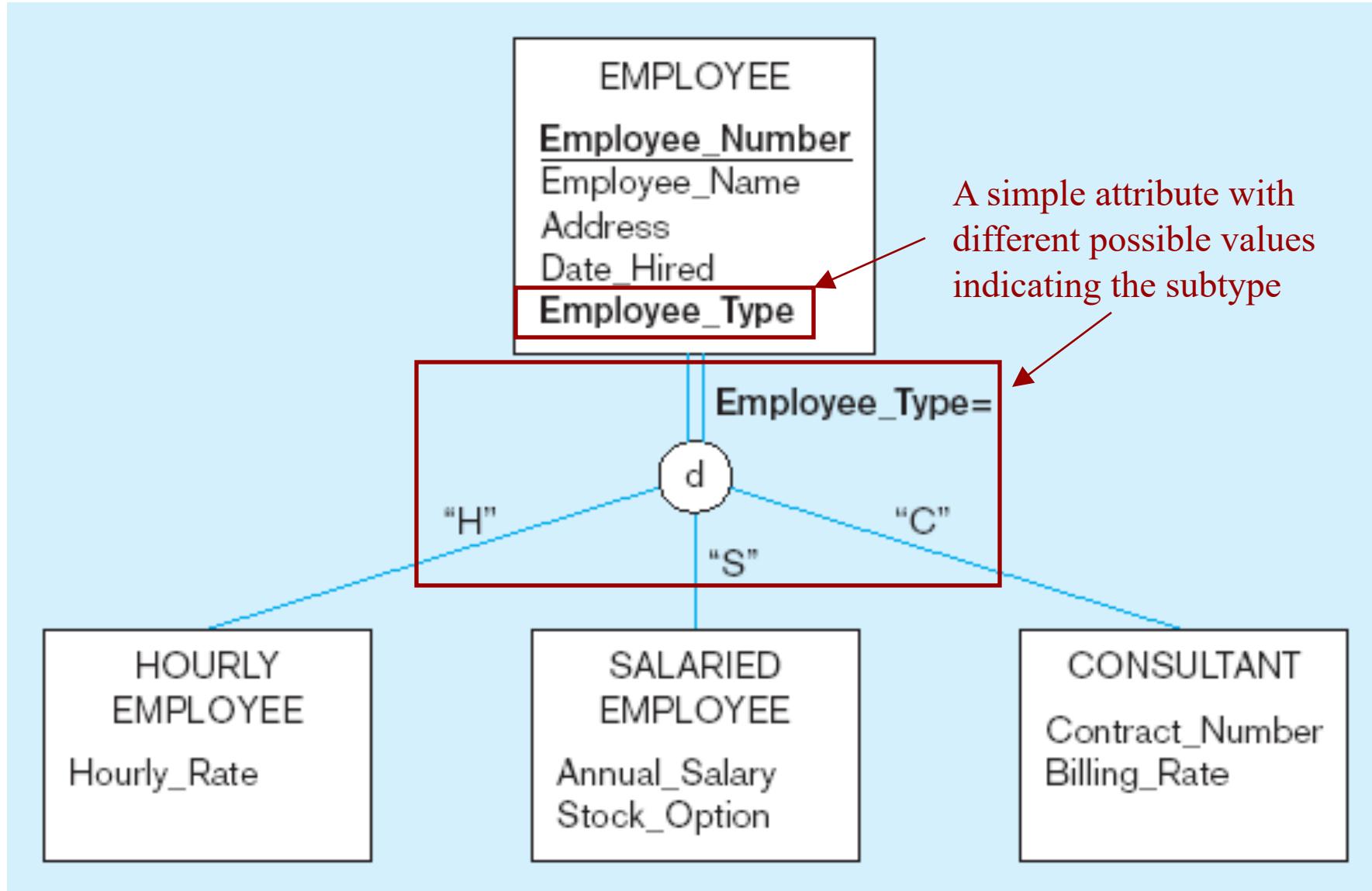
Examples of disjointness constraints (cont.)



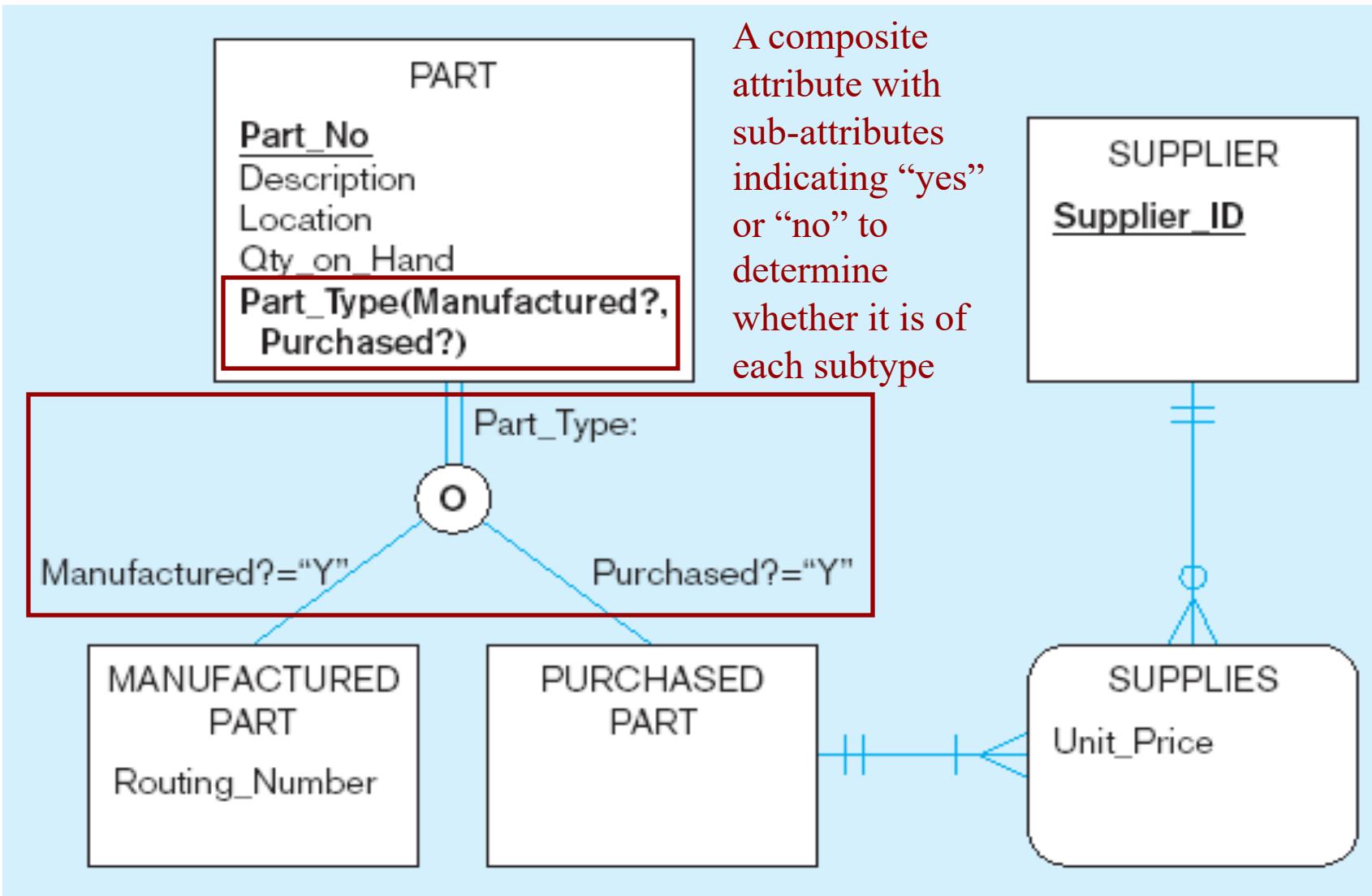
Constraints in Supertype/ Subtype Discriminators

- **Subtype Discriminator**: An attribute of the supertype whose values determine the target subtype(s)
 - **Disjoint** – a *simple* attribute with alternative values to indicate the possible subtypes
 - **Overlapping** – a *composite* attribute whose subparts pertain to different subtypes. Each subpart contains a boolean value to indicate whether or not the instance belongs to the associated subtype

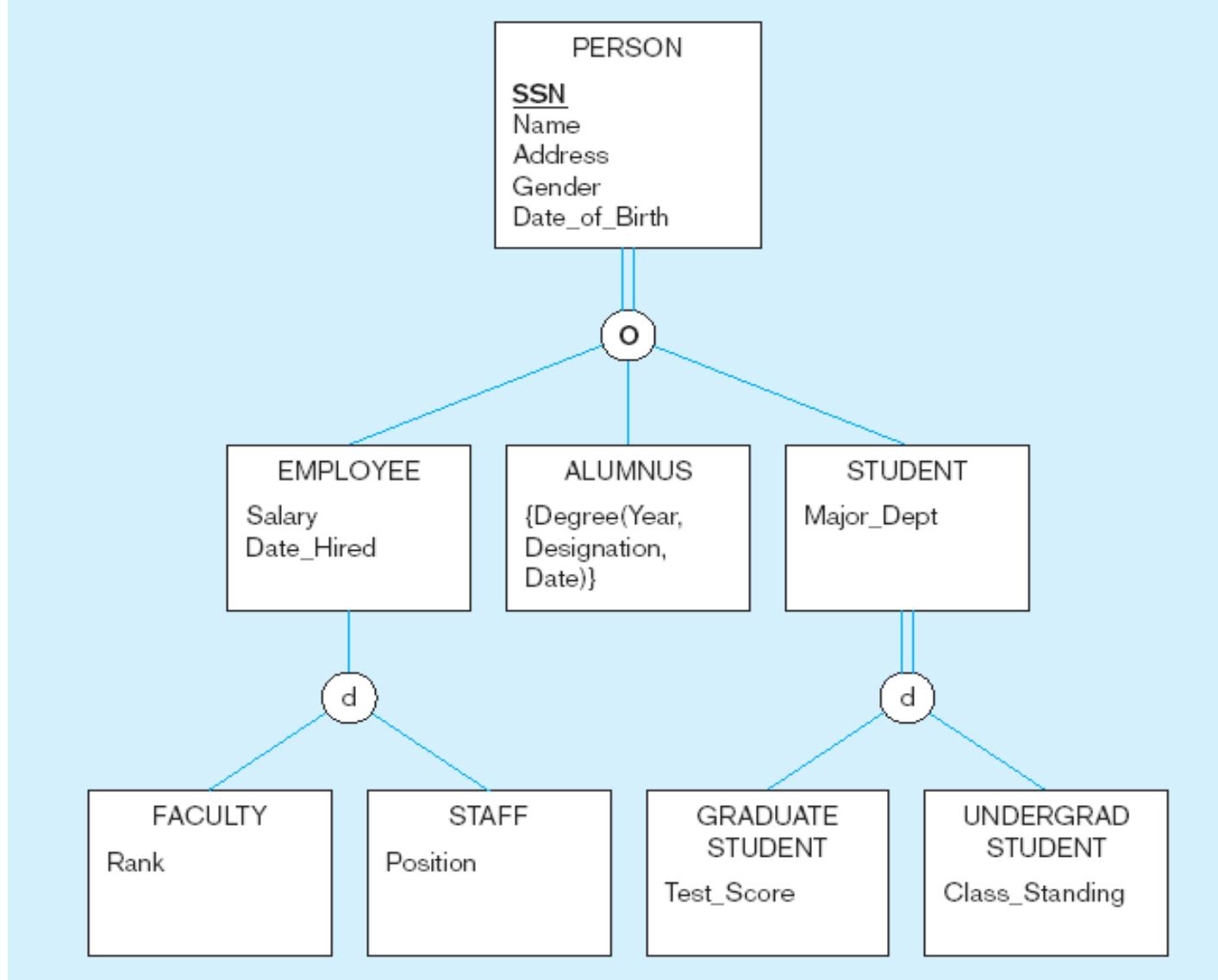
Introducing a subtype discriminator (*disjoint* rule)



Subtype discriminator (**overlap** rule)



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

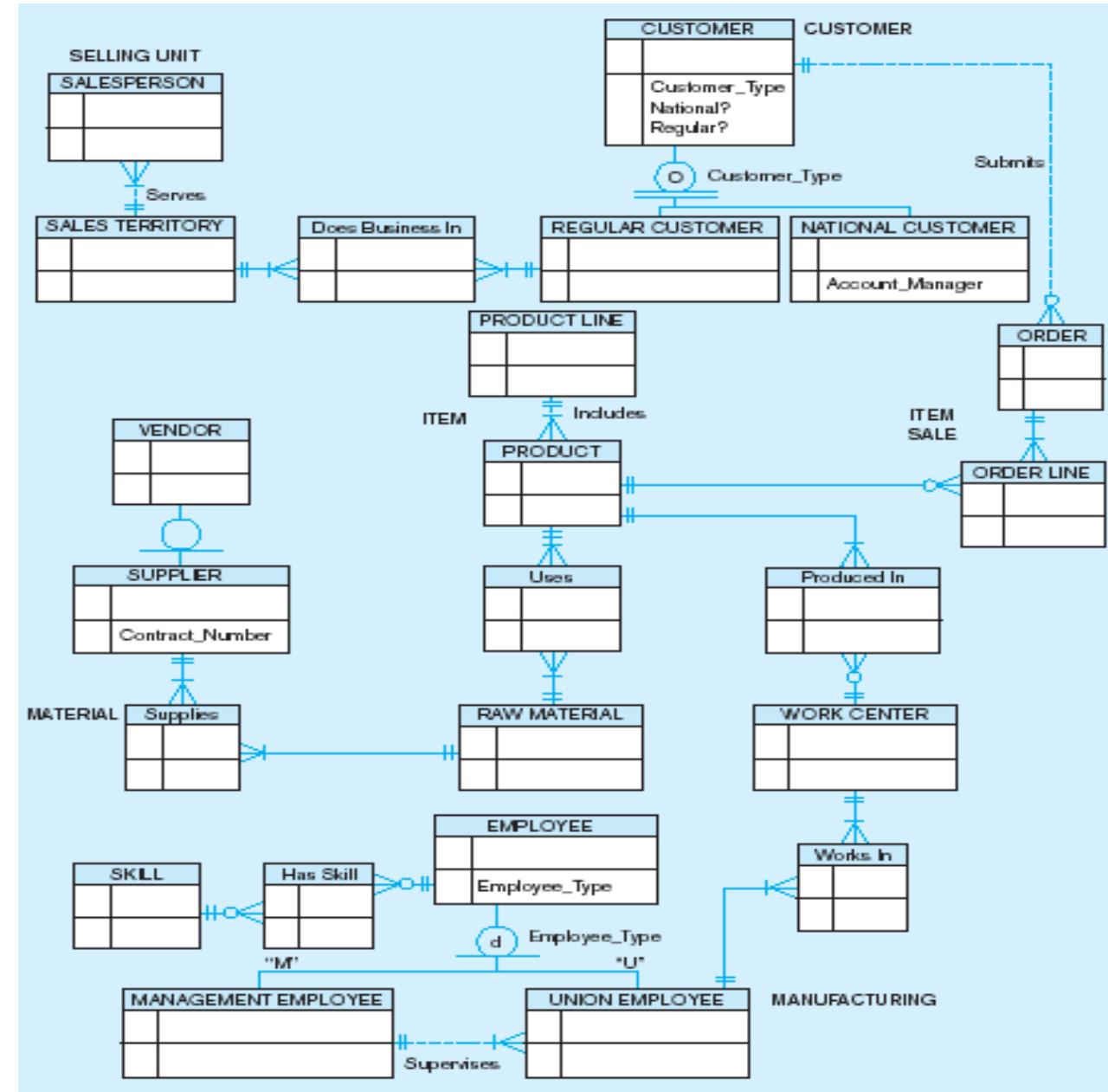


Figure 4-13b EER diagram of PVF entity clusters

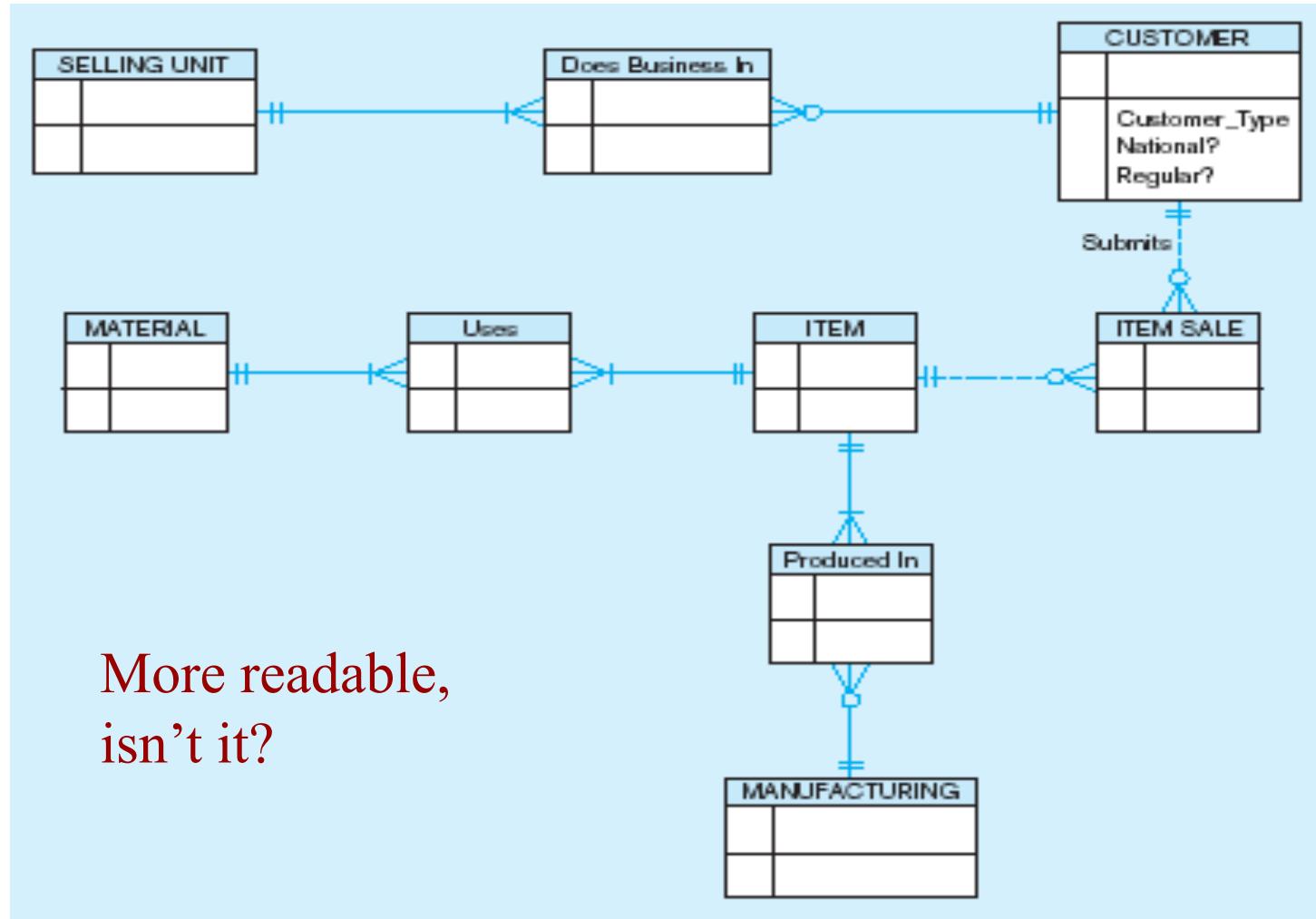
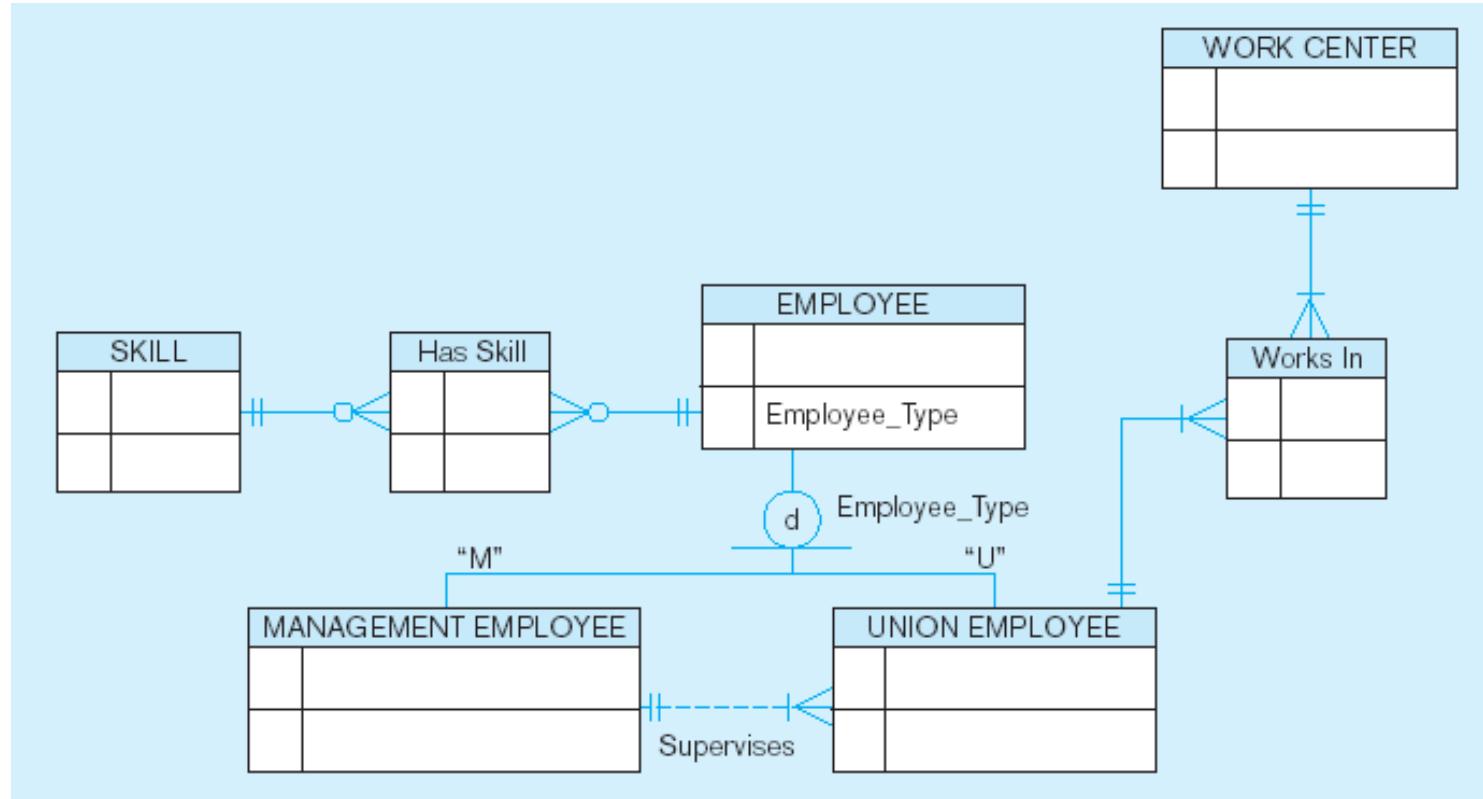
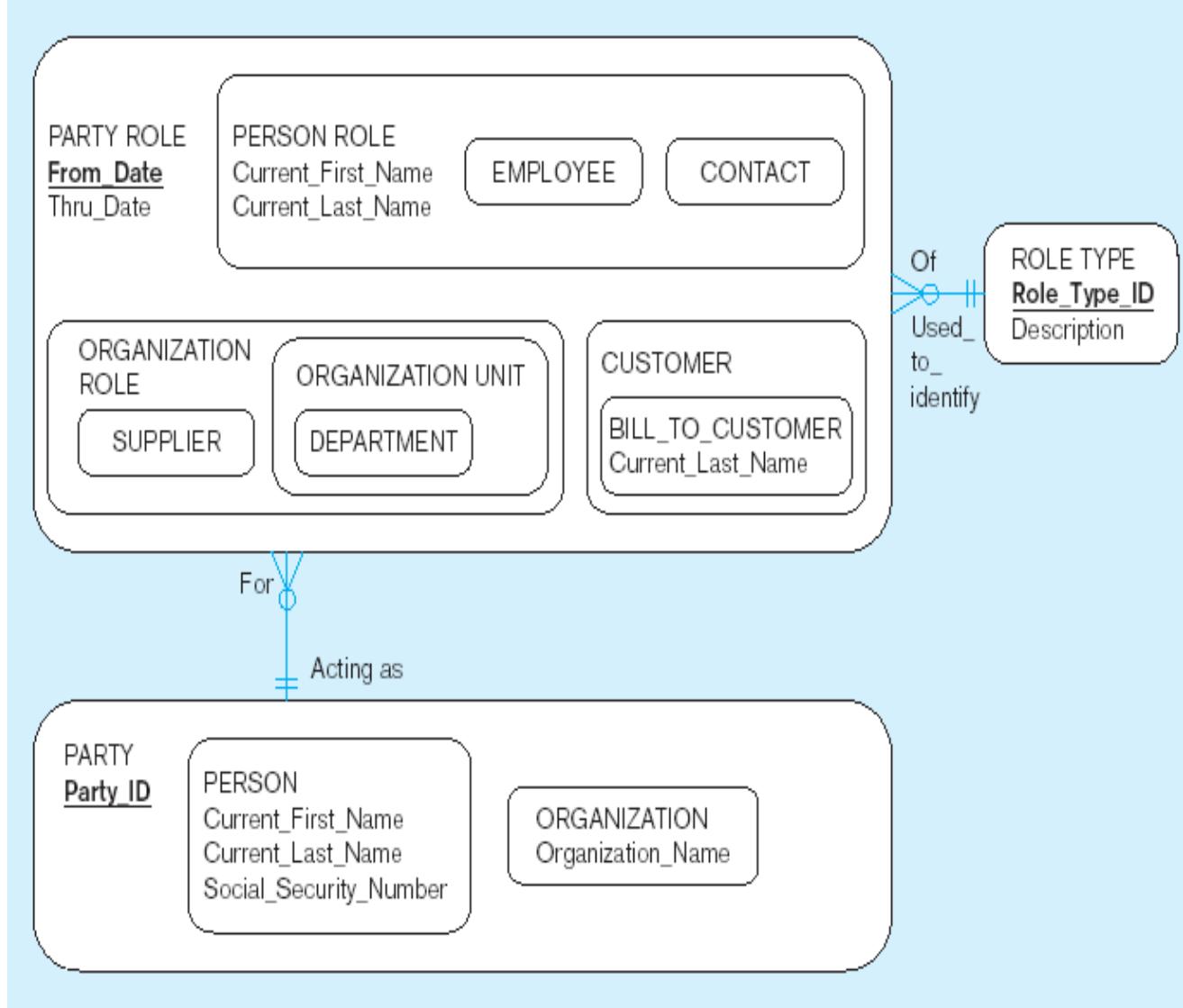


Figure 4-14 Manufacturing entity cluster



Detail for a single cluster



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

Case study

- A college **course** may have **one or more** scheduled sections, or **may not** have a scheduled section.
- Attributes of **COURSE** include Course ID, Course Name, and Units.
- Attributes of **SECTION** include Section Number and Semester ID.
- Semester ID is composed of two parts: Semester and Year. Section Number is an integer (such as 1 or 2) that distinguishes one section from another for the same course but **does not uniquely** identify a section.