Entity-Relationship Modeling

MSBA 7024 / MACC 7020

Database Design and Management

Objectives

- Discuss the importance of data modeling
- Write good names and definitions for entities, relationships, and attributes
- Distinguish unary, binary, and ternary relationships
- Model different types of attributes, entities, relationships, and cardinalities
- Draw E-R diagrams for common business situations
- Convert many-to-many relationships to associative entities
- Model time-dependent data using time stamps

Objectives

- Use of supertype/subtype relationships
- Use of generalization and specialization techniques
- Specification of completeness and disjointness constraints
- Develop supertype/subtype hierarchies for realistic business situations
- Develop entity clusters
- Name categories of business rules
- Define operational constraints graphically and in English

Business Rules

- Statements that define or constrain some aspect of the business
- Are derived from policies, procedures, events, functions
- Assert business structure
- Control/influence business behavior
- Expressed in terms familiar to end users
- Automated through DBMS software

Business Rules

- Statements that define or constrain some aspect of the business
- A Good Business Rule is:
  - Declarative—what, not how
  - Precise—clear, agreed-upon meaning
  - Atomic—one statement
  - Consistent—internally and externally
  - Expressible—structured, natural language
  - Distinct—non-redundant
  - Business-oriented—understood by business people

Data Definitions

- Explanation of a term or fact
  - Term—word or phrase with specific meaning
  - Fact—association between two or more terms
- Guidelines for good data definition
  - Gathered in conjunction with systems requirements
  - Accompanied by diagrams
  - Iteratively created and refined
  - Achieved by consensus

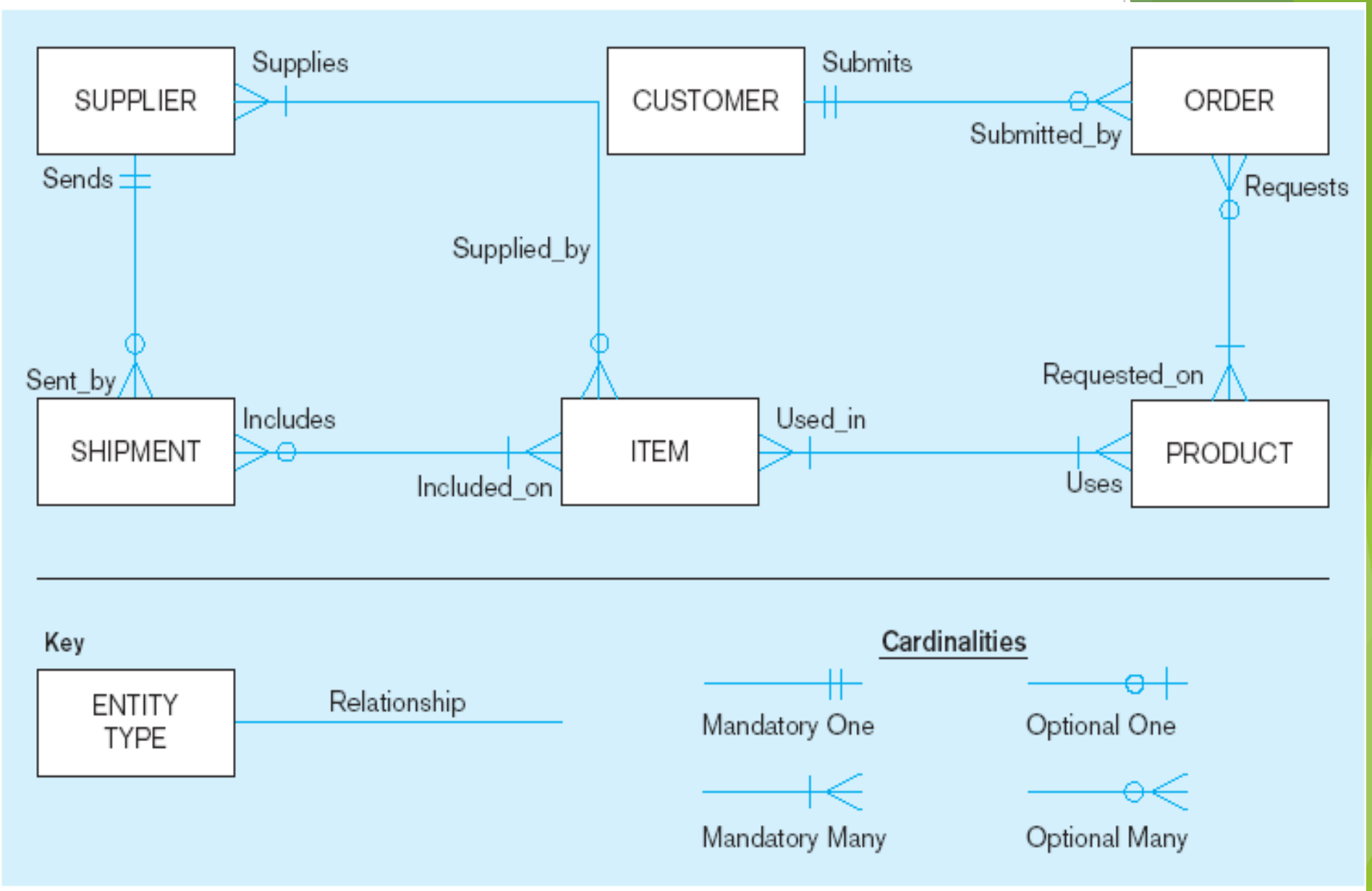
A Good Data Name is:

- Related to business, not technical, characteristics
- Meaningful and self-documenting
- Unique
- Readable
- Composed of words from an approved list
- Repeatable

E-R Model Constructs

- Entities:
  - Entity instance—person, place, object, event, concept (often corresponds to a row in a table)
  - Entity Type—collection of entities (often corresponds to a table)
- Relationships:
  - Relationship instance—link between entities (corresponds to primary key-foreign key equivalencies in related tables)
  - Relationship type—category of relationship…link between entity types
- Attribute—property or characteristic of an entity or relationship type (often corresponds to a field in a table)

Sample E-R Diagram

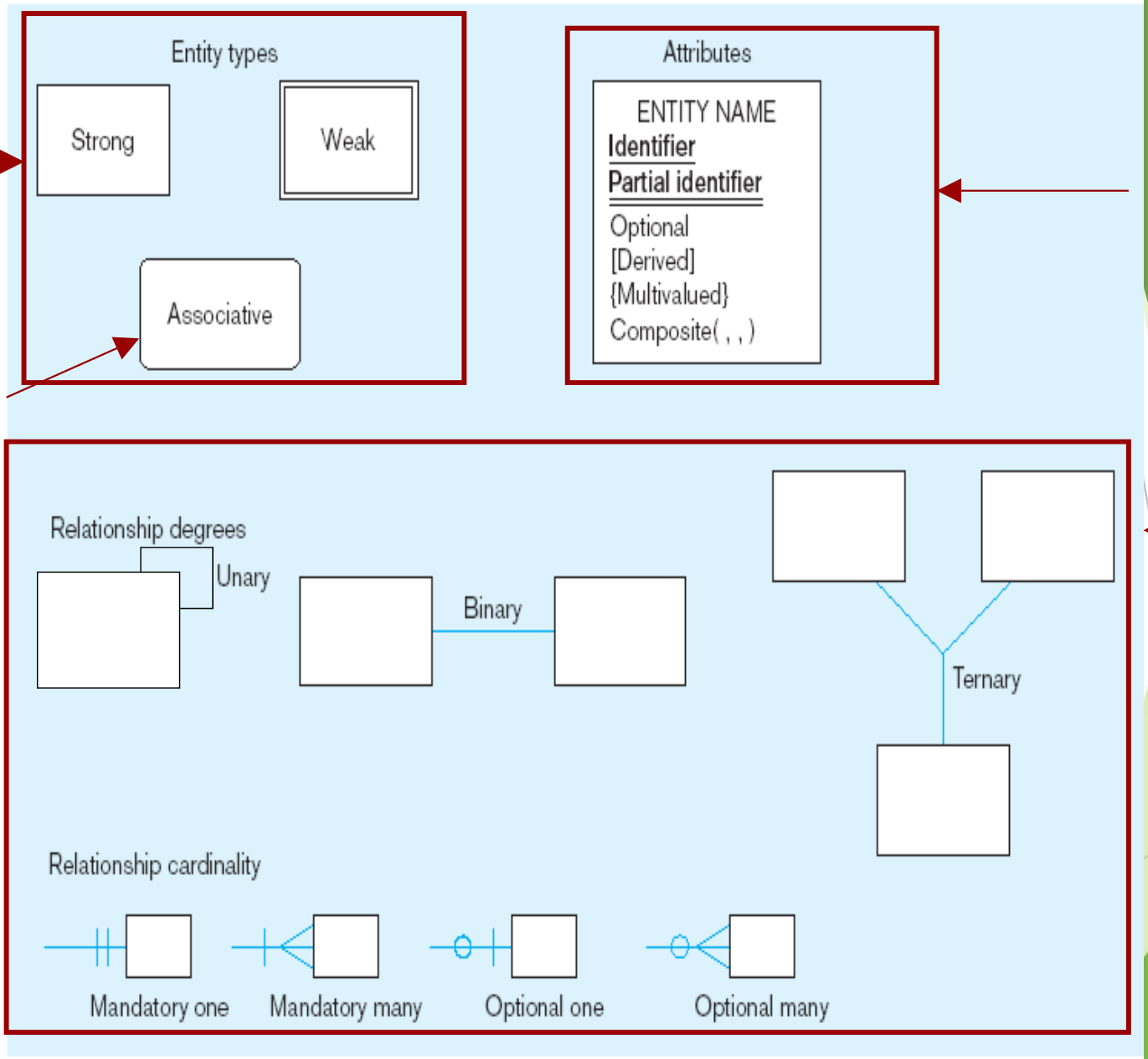


Basic E-R notation

Entity symbols

A special entity that is also a relationship

Relationship degrees specify number of entity types involved



Attribute symbols

Relationship symbols

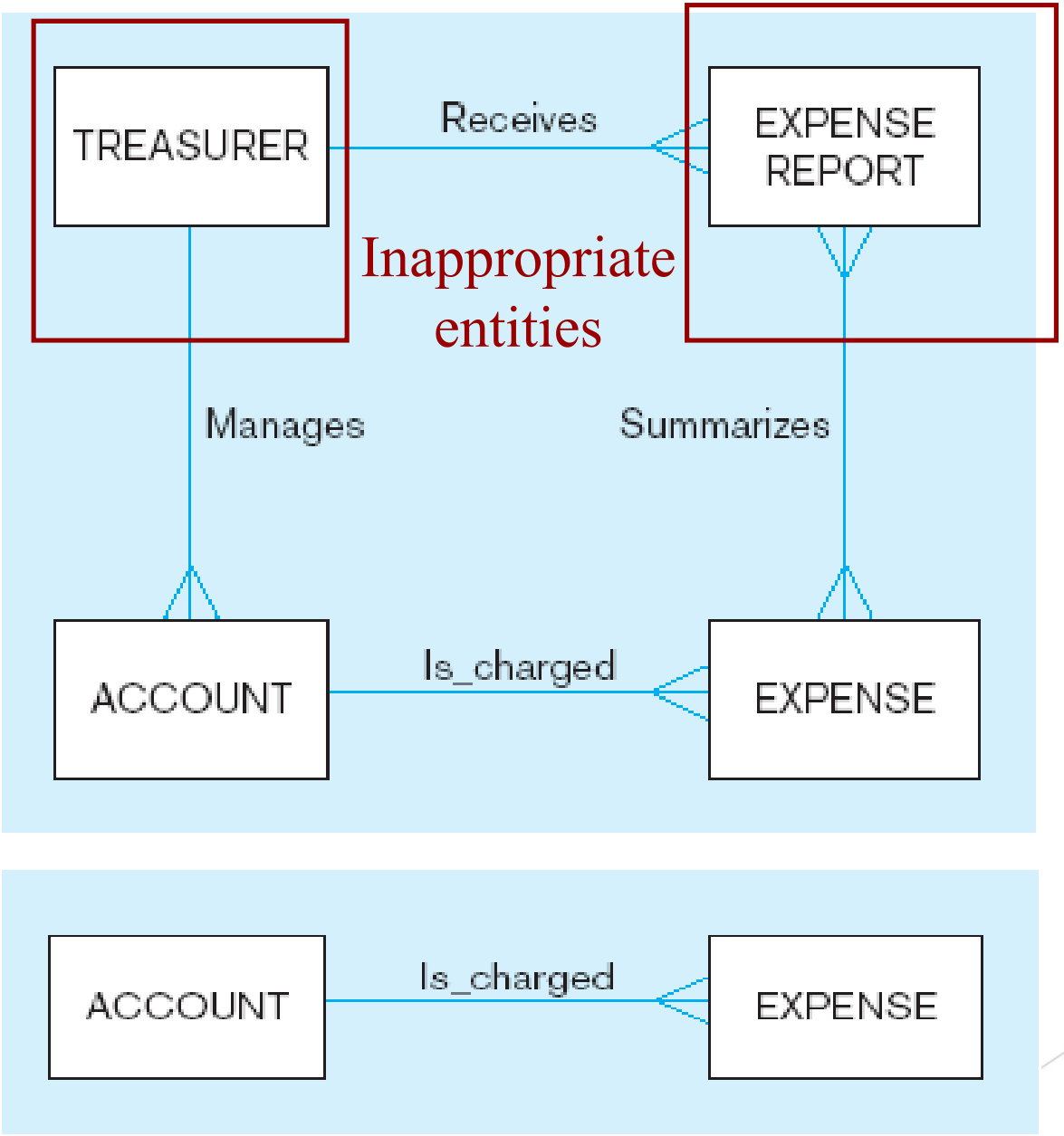
Relationship cardinalities specify how many of each entity type is allowed

What Should an Entity Be?

- SHOULD BE:
  - An object that will have many instances in the database
  - An object that will be composed of multiple attributes
  - An object that we are trying to model
- SHOULD NOT BE:
  - A user of the database system
  - An output of the database system (e.g., a report)

Example of inappropriate entities

System user



System output

Appropriate entities

Attributes

- Attribute—property or characteristic of an entity or relationship type
- Classifications of attributes:
  - Required versus Optional Attributes
  - Simple versus Composite Attribute
  - Single-valued versus Multivalued Attribute
  - Stored versus Derived Attributes
  - Identifier Attributes

Identifiers (Keys)

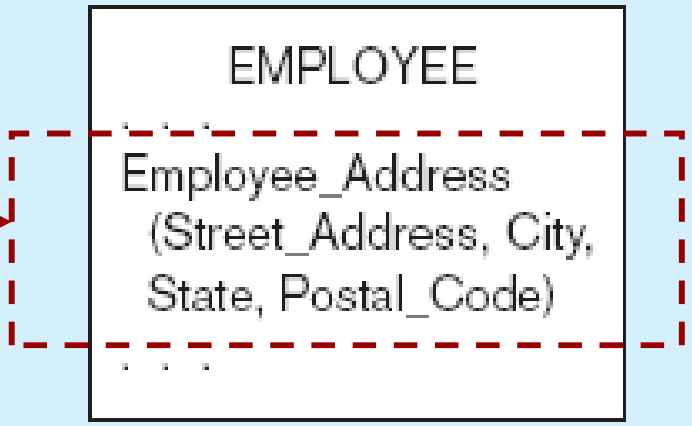
- Identifier (Key)—An attribute (or combination of attributes) that uniquely identifies individual instances of an entity type
- Simple versus Composite Identifier
- Candidate Identifier—an attribute that could be a key… satisfies the requirements for being an identifier

Characteristics of Identifiers

- Will not change in value
- Will not be null
- No intelligent identifiers (e.g., containing locations or people that might change)
- Substitute new, simple keys for long, composite keys

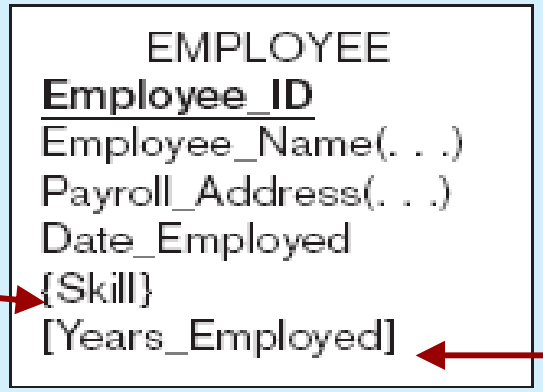
A **composite** attribute

An attribute broken into component parts



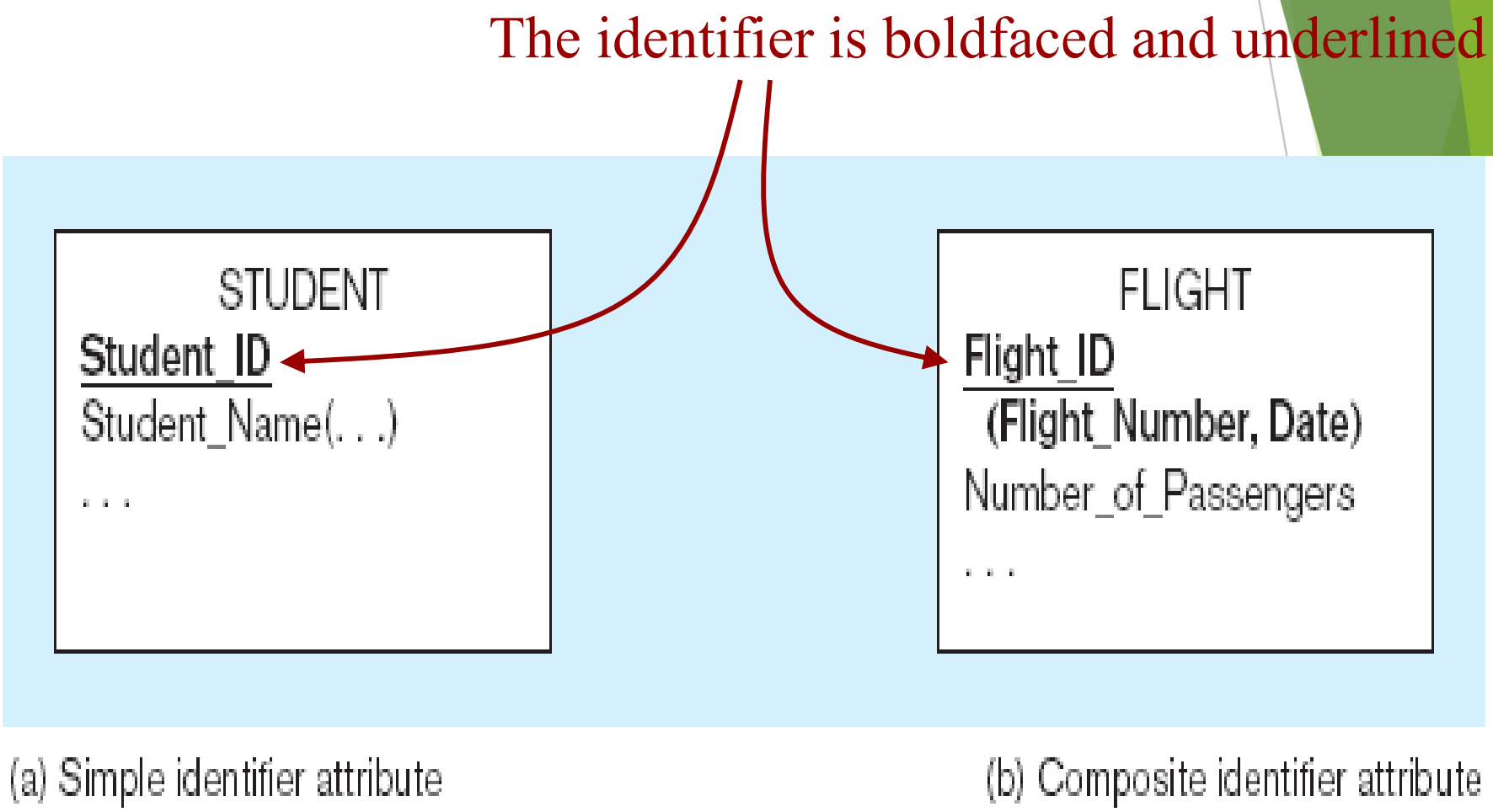
Entity with **multivalued** attribute (Skill) and **derived** attribute (Years_Employed)

Multivalued
an employee can have more than one skill



Derived
from date employed and current date

Simple and composite identifier attributes



Simple example of time-stamping

This attribute is both multivalued *and* composite

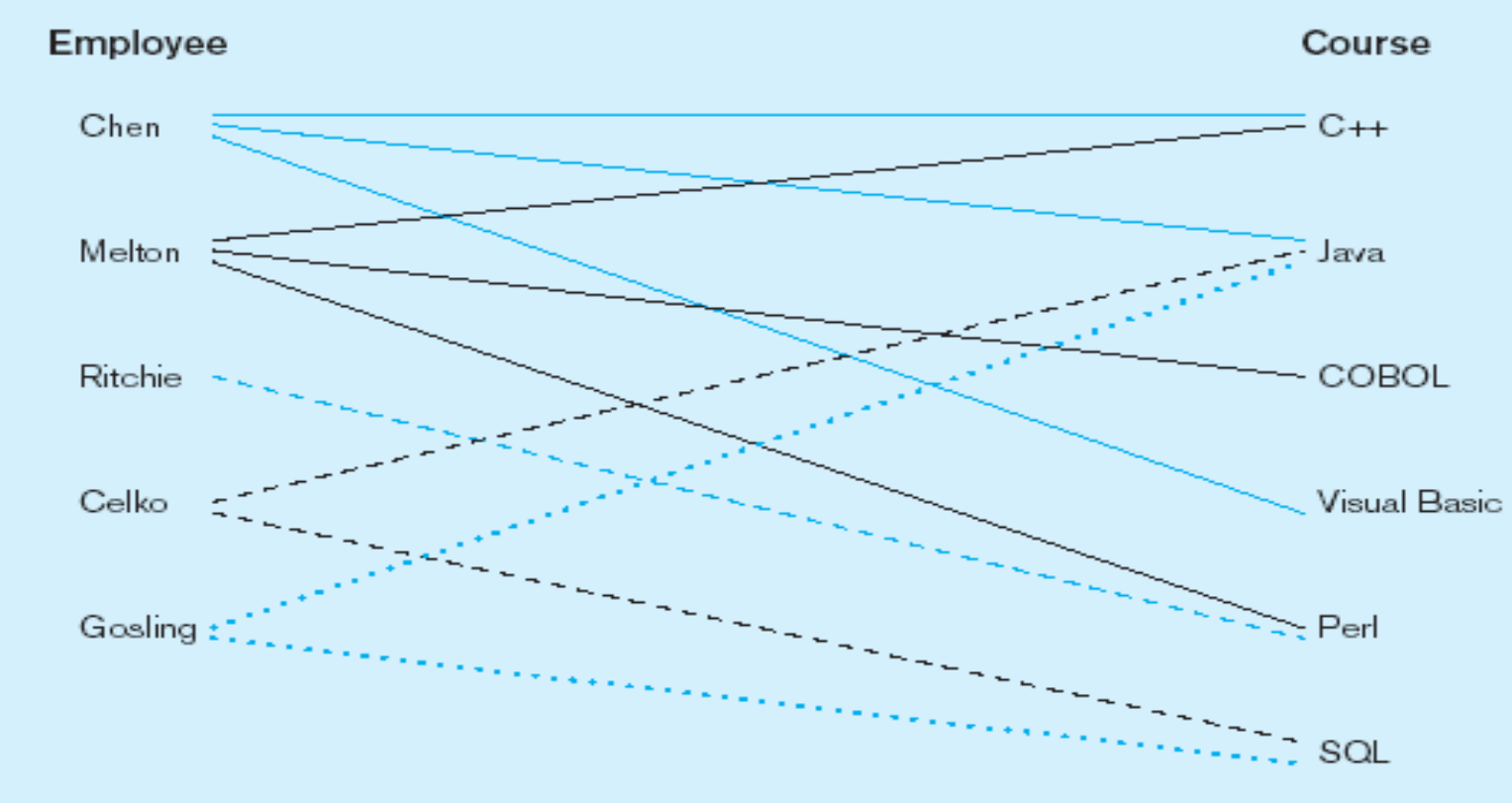
More on Relationships

- Relationship Types vs. Relationship Instances
  - The relationship type is modeled as lines between entity types…the instance is between specific entity instances
- Relationships can have attributes
  - These describe features pertaining to the association between the entities in the relationship
- Two entities can have more than one type of relationship between them (multiple relationships)
- Associative Entity—combination of relationship and entity

Relationship types and instances

a) Relationship type

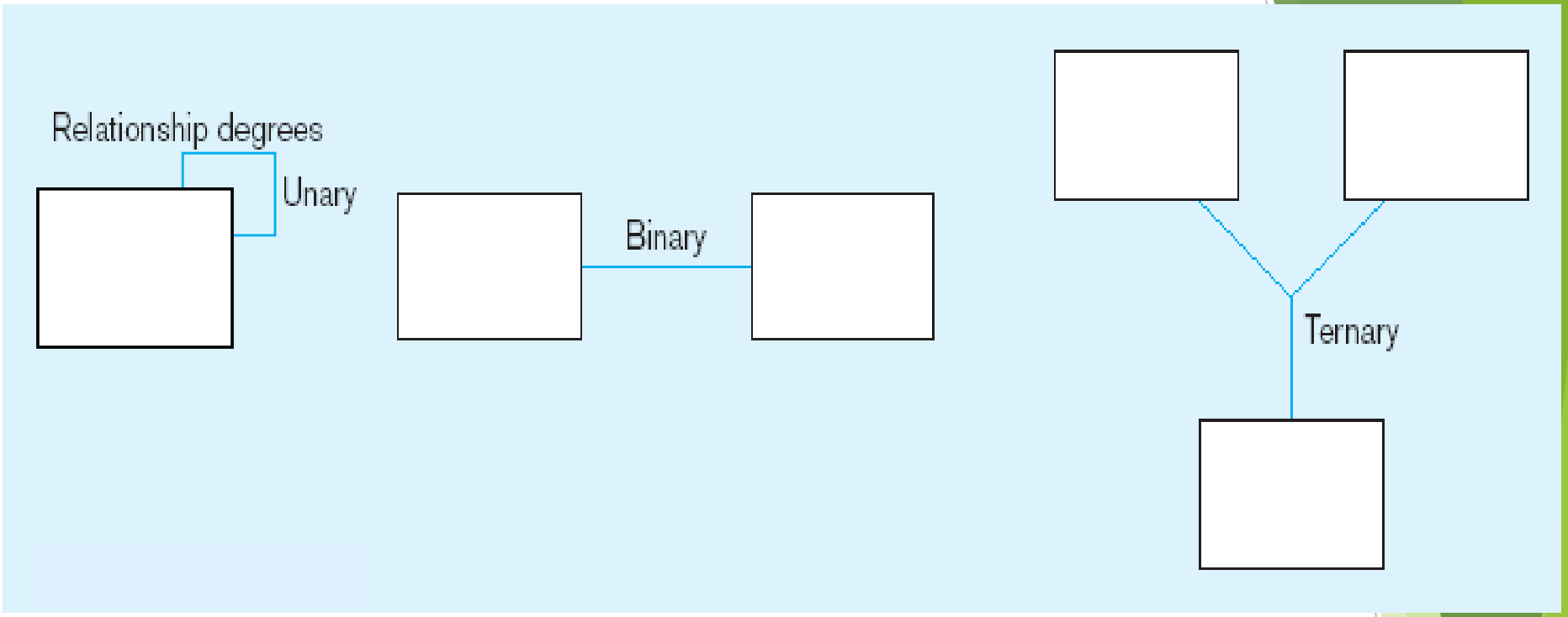
b) Relationship instances



Degree of Relationships

- Degree of a relationship is the number of entity types that participate in it
  - Unary Relationship
  - Binary Relationship
  - Ternary Relationship

Degree of relationships



One entity related to another of the same entity type

Entities of two different types related to each other

Entities of three different types related to each other

Cardinality of Relationships

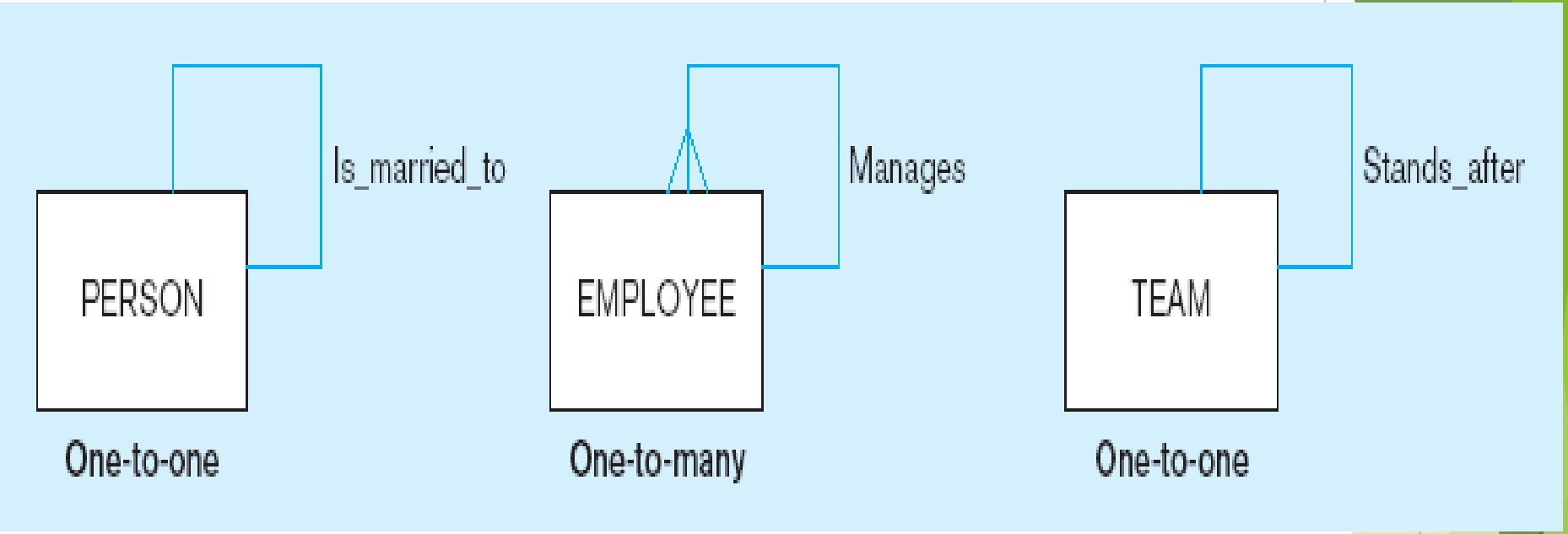
- One-to-One
  - Each entity in the relationship will have exactly one related entity
- One-to-Many
  - An entity on one side of the relationship can have many related entities, but an entity on the other side will have a maximum of one related entity
- Many-to-Many
  - Entities on both sides of the relationship can have many related entities on the other side

Cardinality Constraints

- Cardinality Constraints - the number of instances of one entity that can or must be associated with each instance of another entity
- Minimum Cardinality
  - If zero, then optional
  - If one or more, then mandatory
- Maximum Cardinality
  - The maximum number

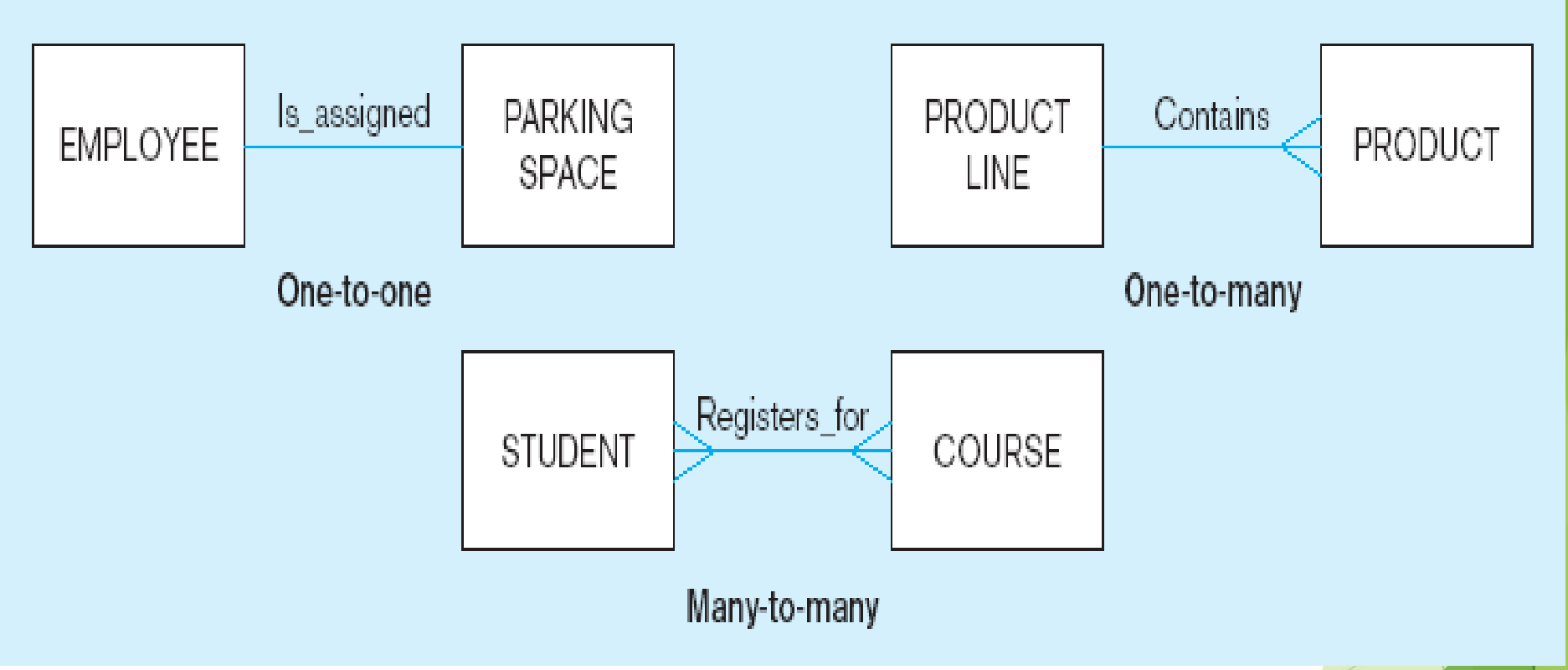
Examples of relationships of different degrees

a) Unary relationships



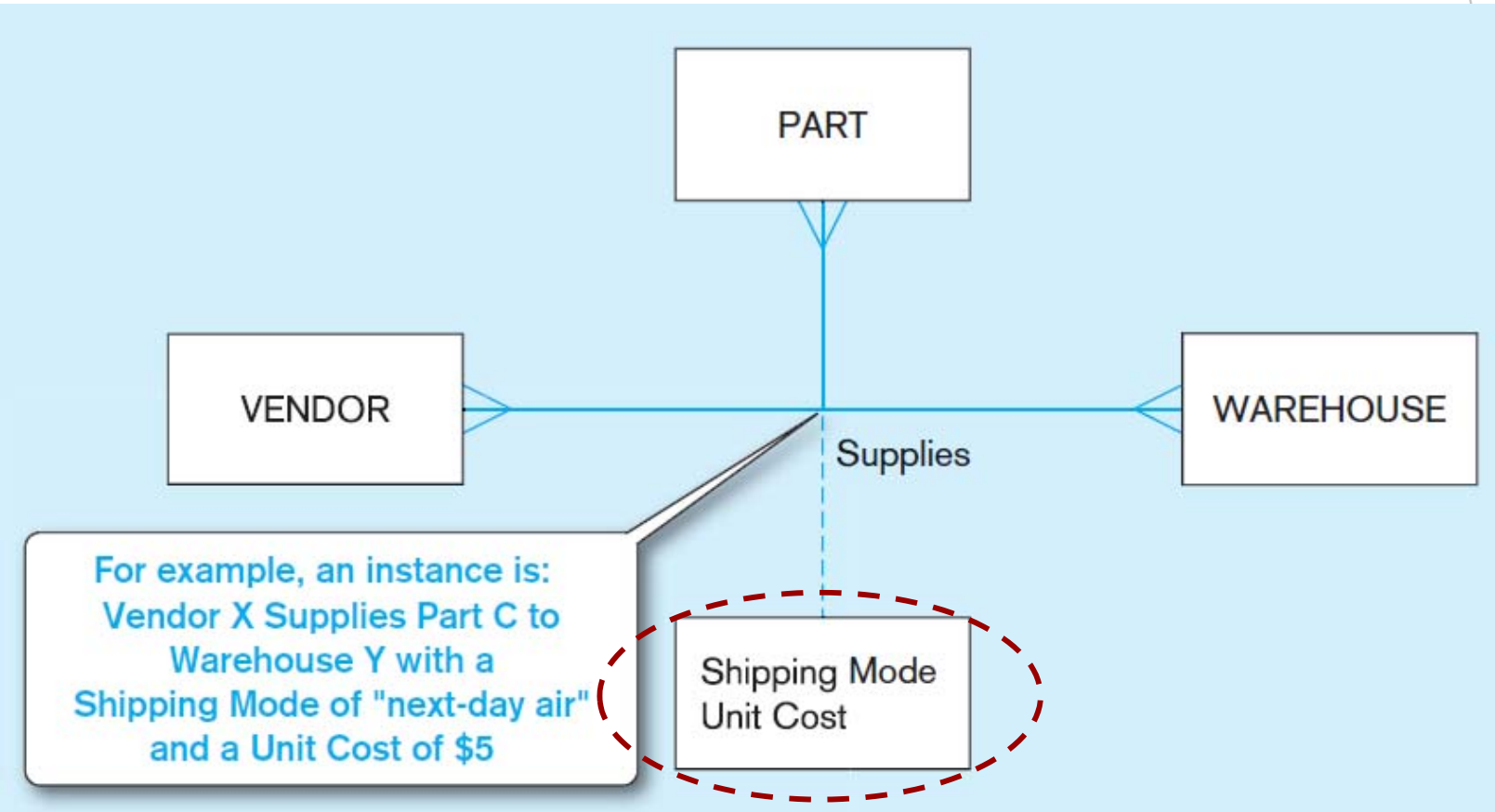
Examples of relationships of different degrees (cont.)

b) Binary relationships



Examples of relationships of different degrees (cont.)

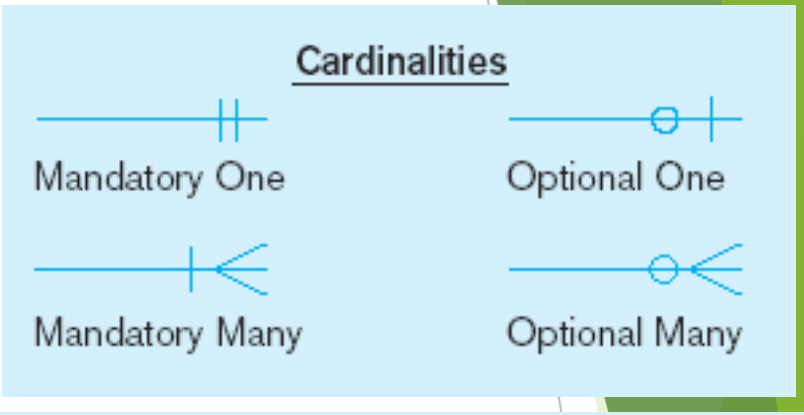
c) Ternary relationship

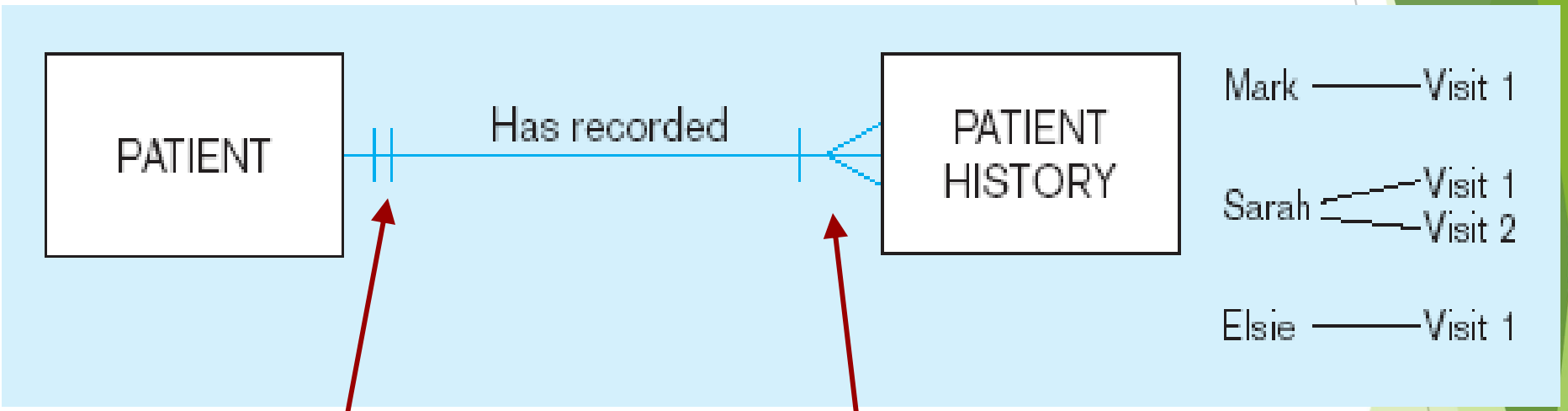


Note: a relationship can have attributes of its own

Examples of cardinality constraints

a) Mandatory cardinalities

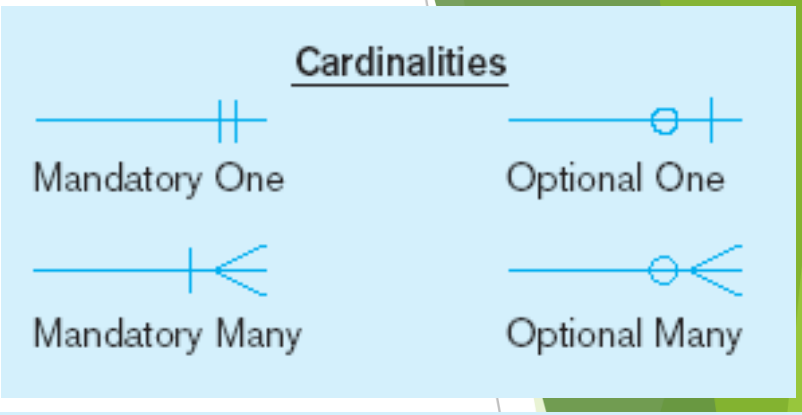




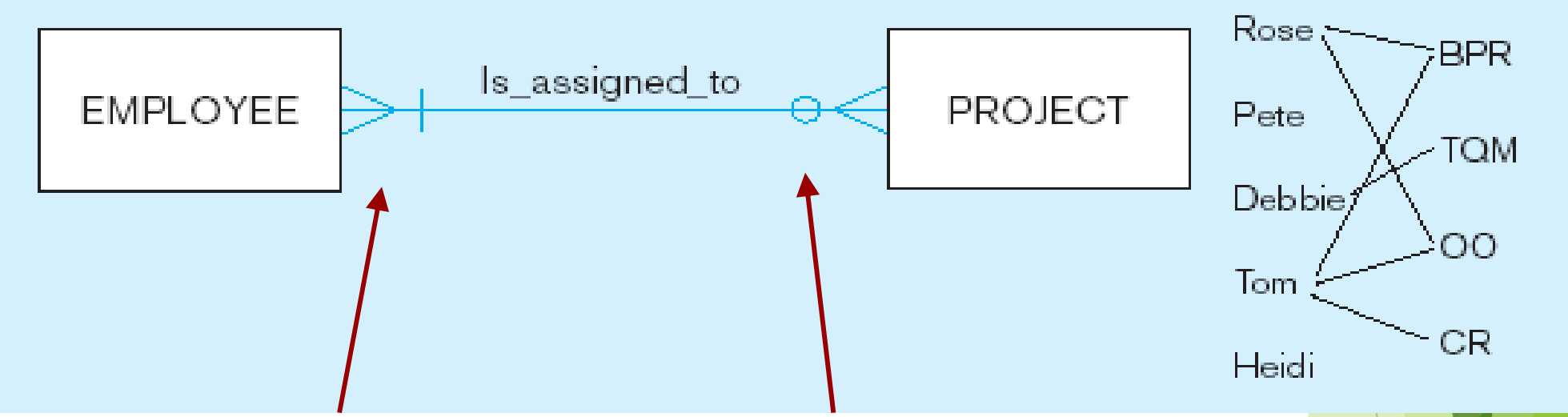
A patient history is recorded for one and only one patient

A patient must have recorded at least one history, and can have many

Examples of cardinality constraints (cont.)



b) One optional, one mandatory

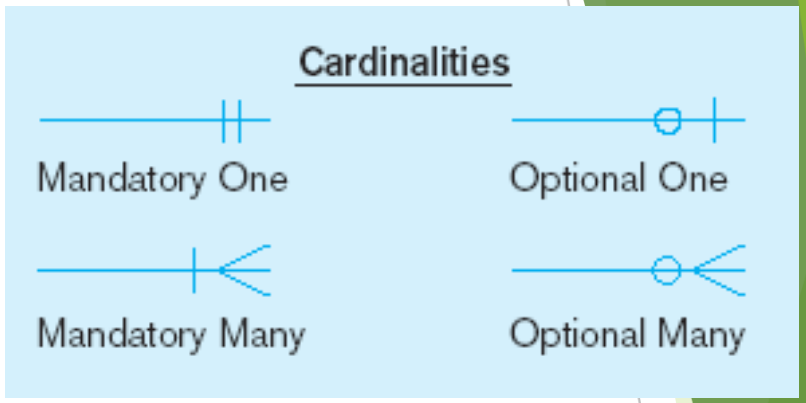


A project must be assigned to at least one employee, and may be assigned to many

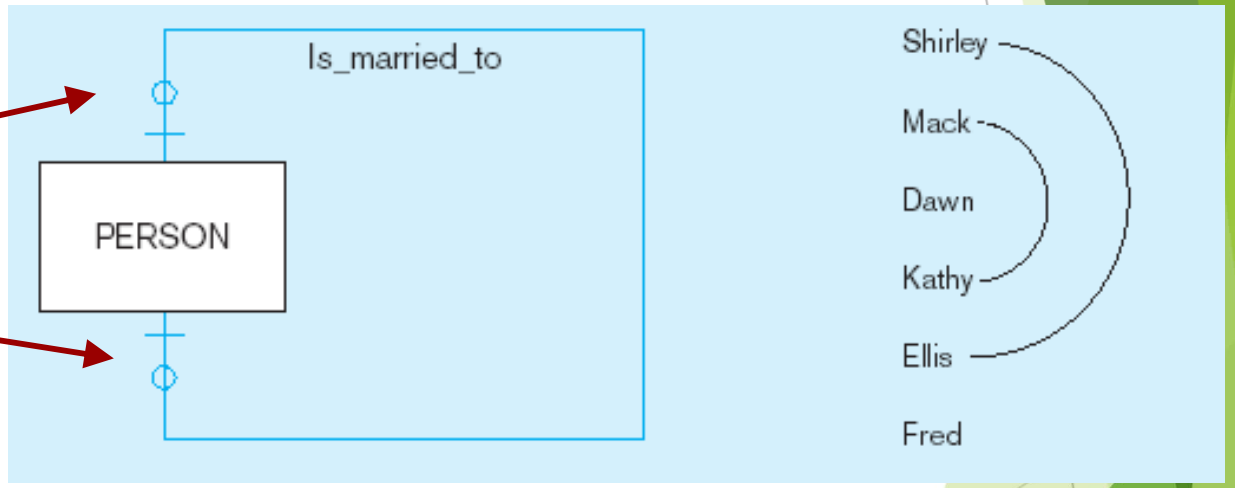
An employee can be assigned to any number of projects, or may not be assigned to any at all

Examples of cardinality constraints (cont.)

a) Optional cardinalities

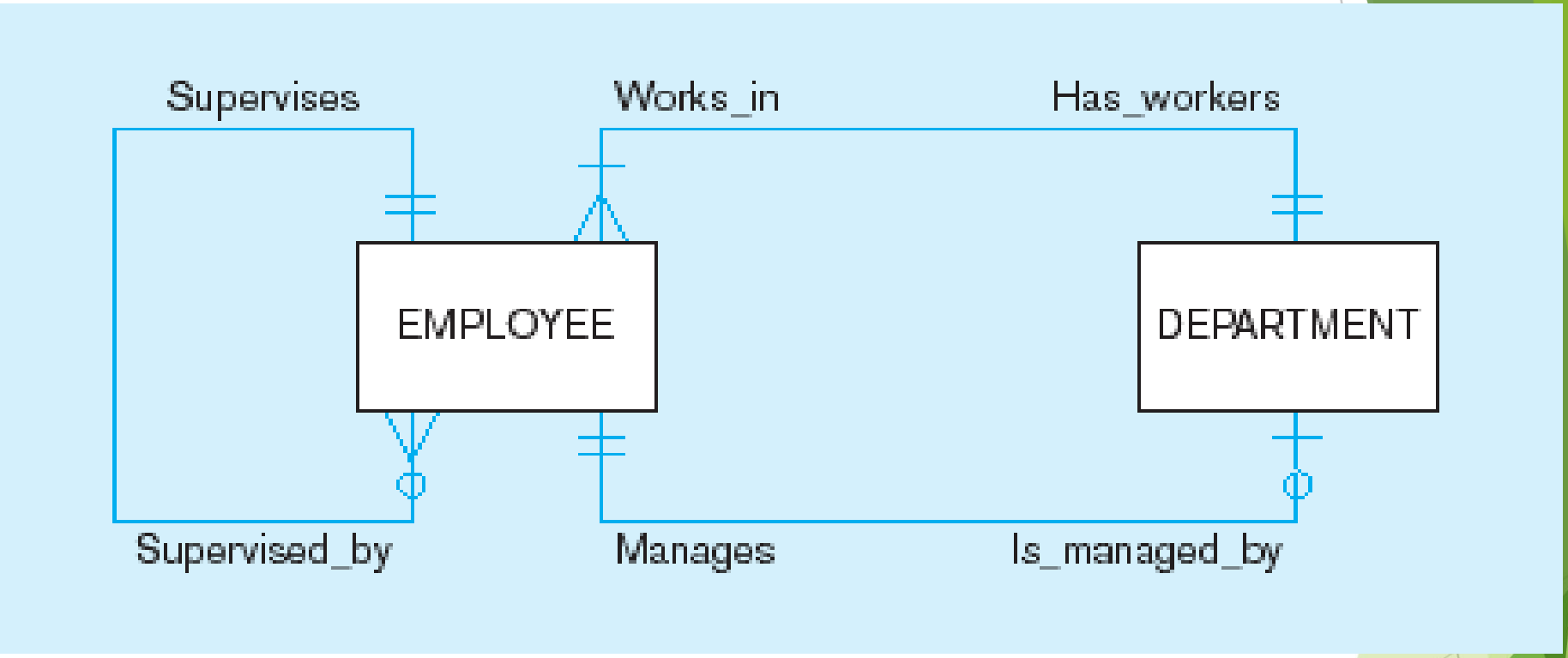


A person is married to at most one other person, or may not be married at all



Examples of multiple relationships

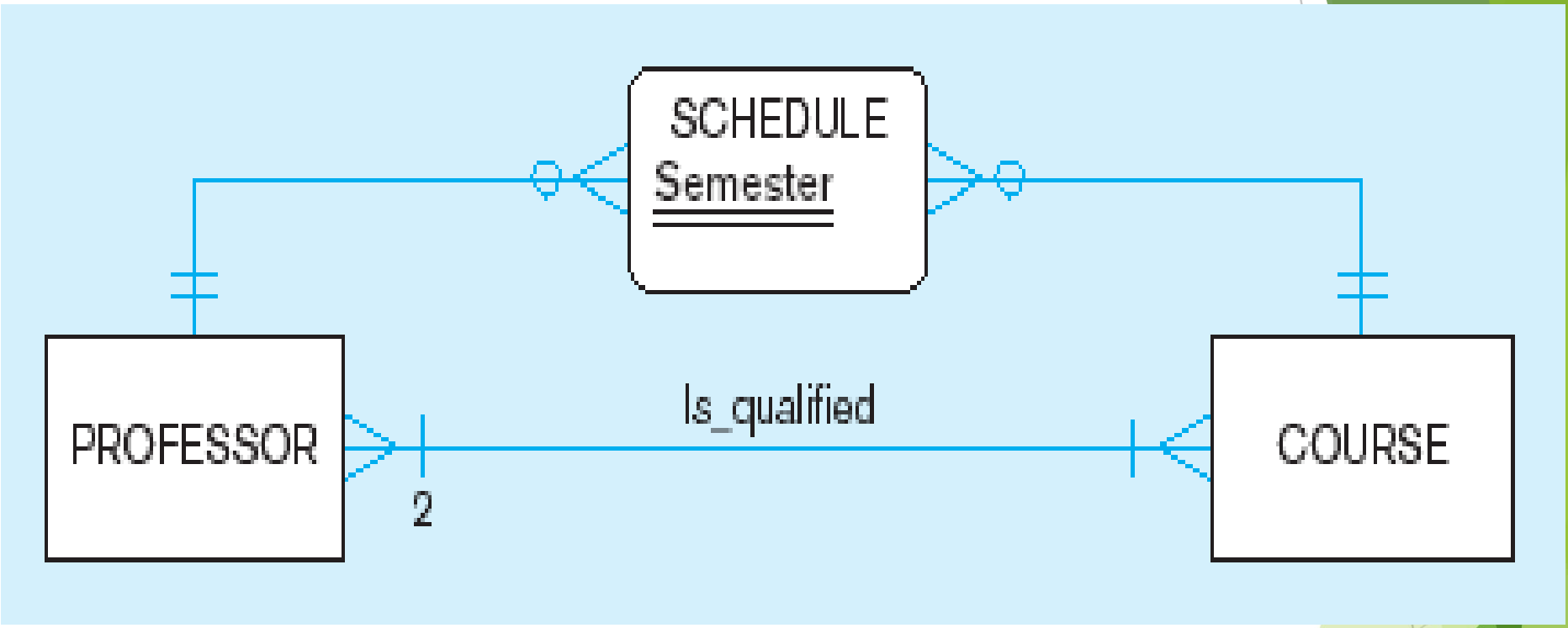
a) Employees and departments



Entities can be related to one another in more than one way

Examples of multiple relationships (cont.)

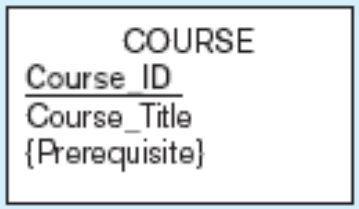
b) Professors and courses (fixed lower limit constraint)

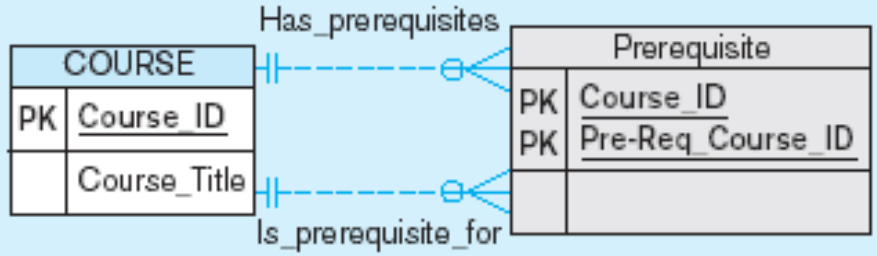


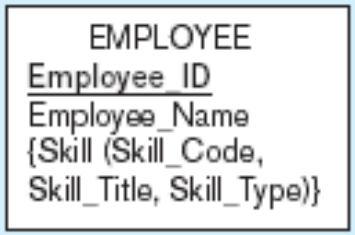
Here, min cardinality constraint is 2

Multivalued attributes can be represented as relationships

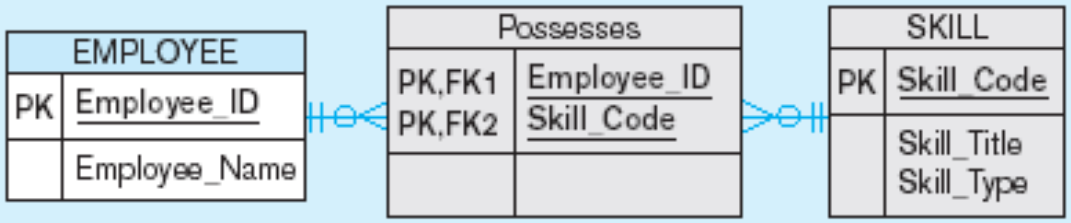
ATTRIBUTE	RELATIONSHIP & ENTITY
simple	







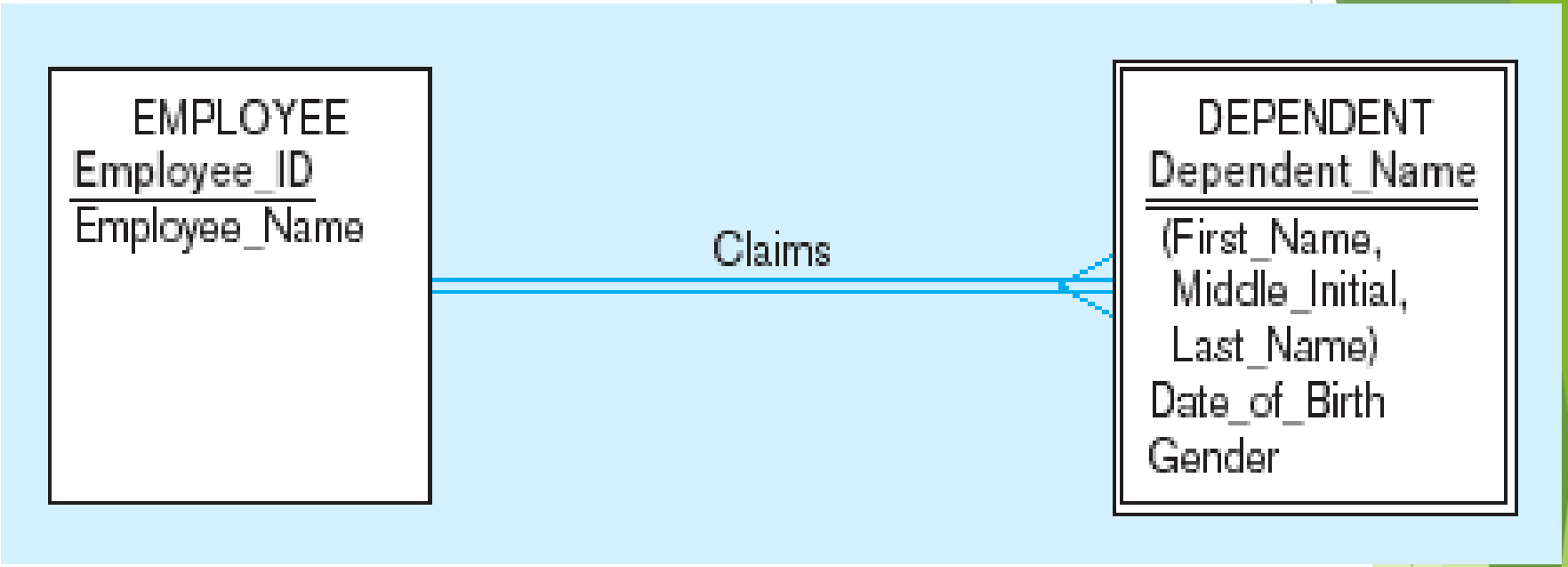
composite



Strong vs. Weak Entities, and Identifying Relationships

- Strong entity
  - exists 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



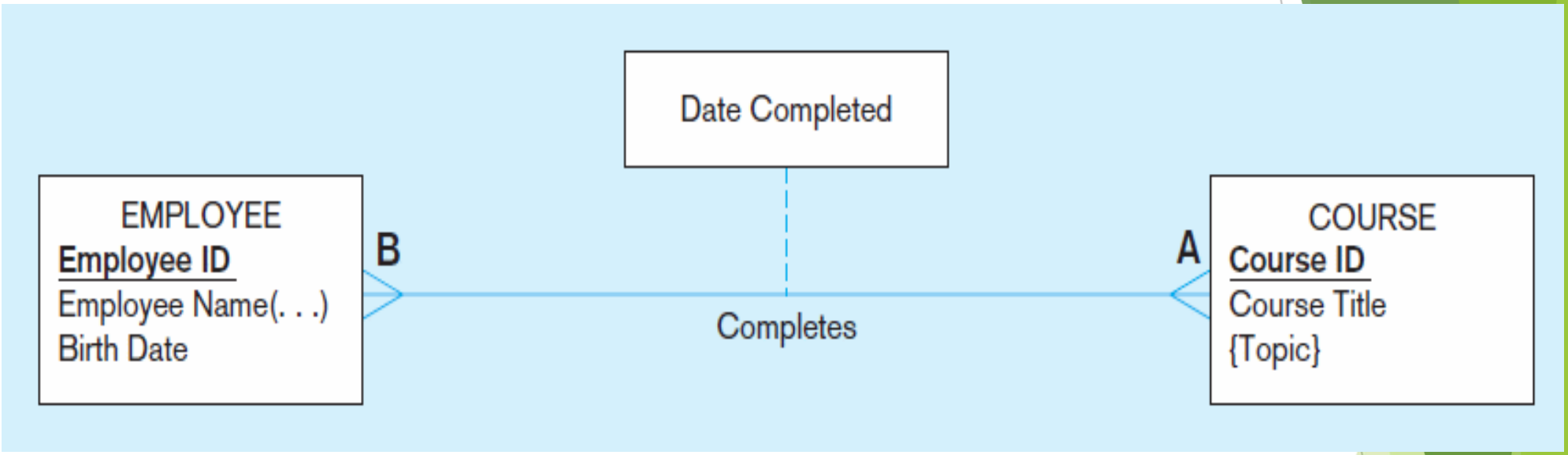
Strong entity

Weak entity

Associative Entities

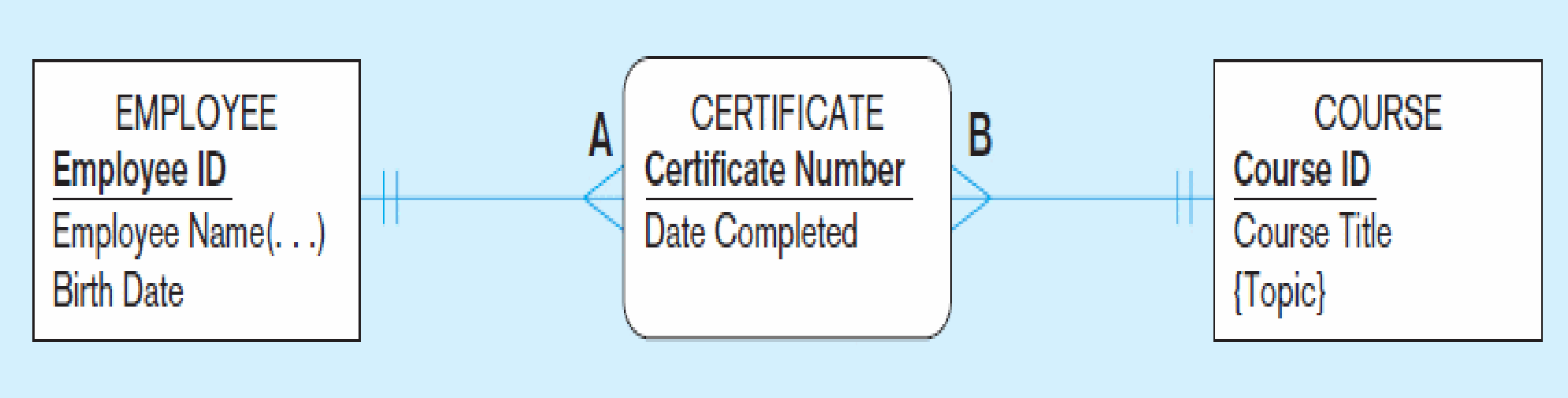
- An entity — has attributes
- A relationship — links entities together
- When should a *relationship with attributes* instead be an *associative entity*?
- Some guidelines:
  - All relationships for the associative entity should be “many”
  - The associative entity could have meaning independent of the other 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
  - Ternary relationships should be converted to associative entities

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*

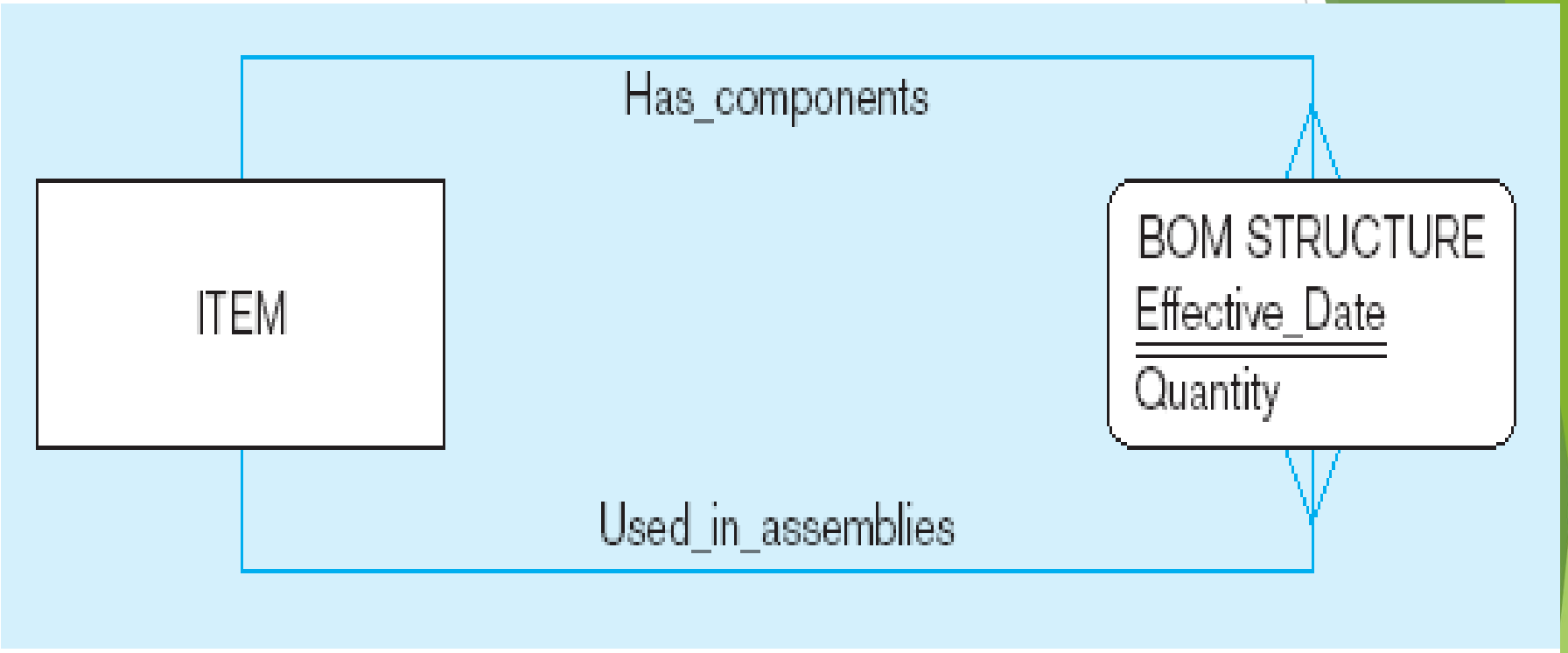
An associative entity (CERTIFICATE)



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

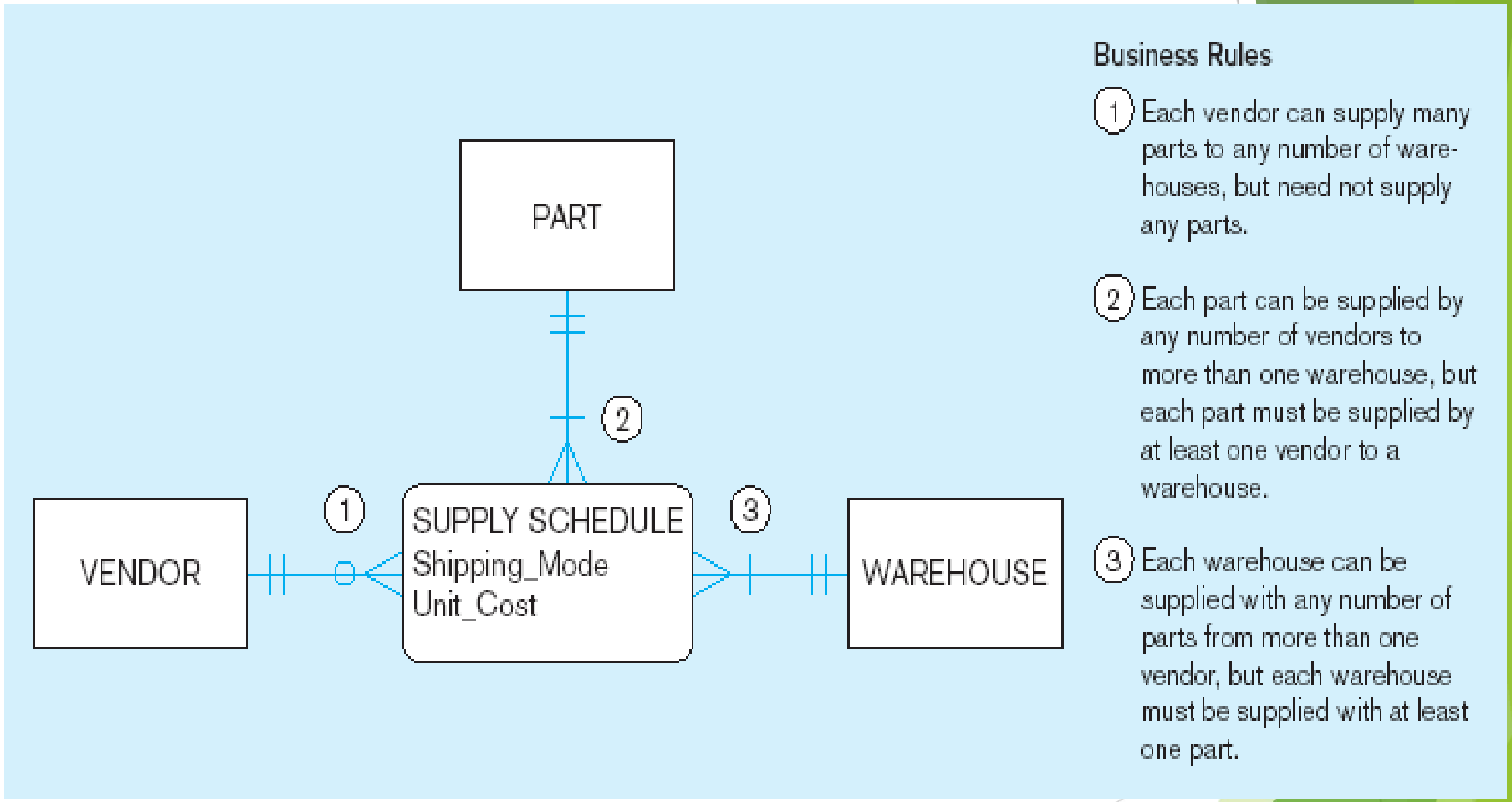
Note that the many-to-many cardinality between entities in the figure in the previous slide has been replaced by two one-to-many relationships with the associative entity.

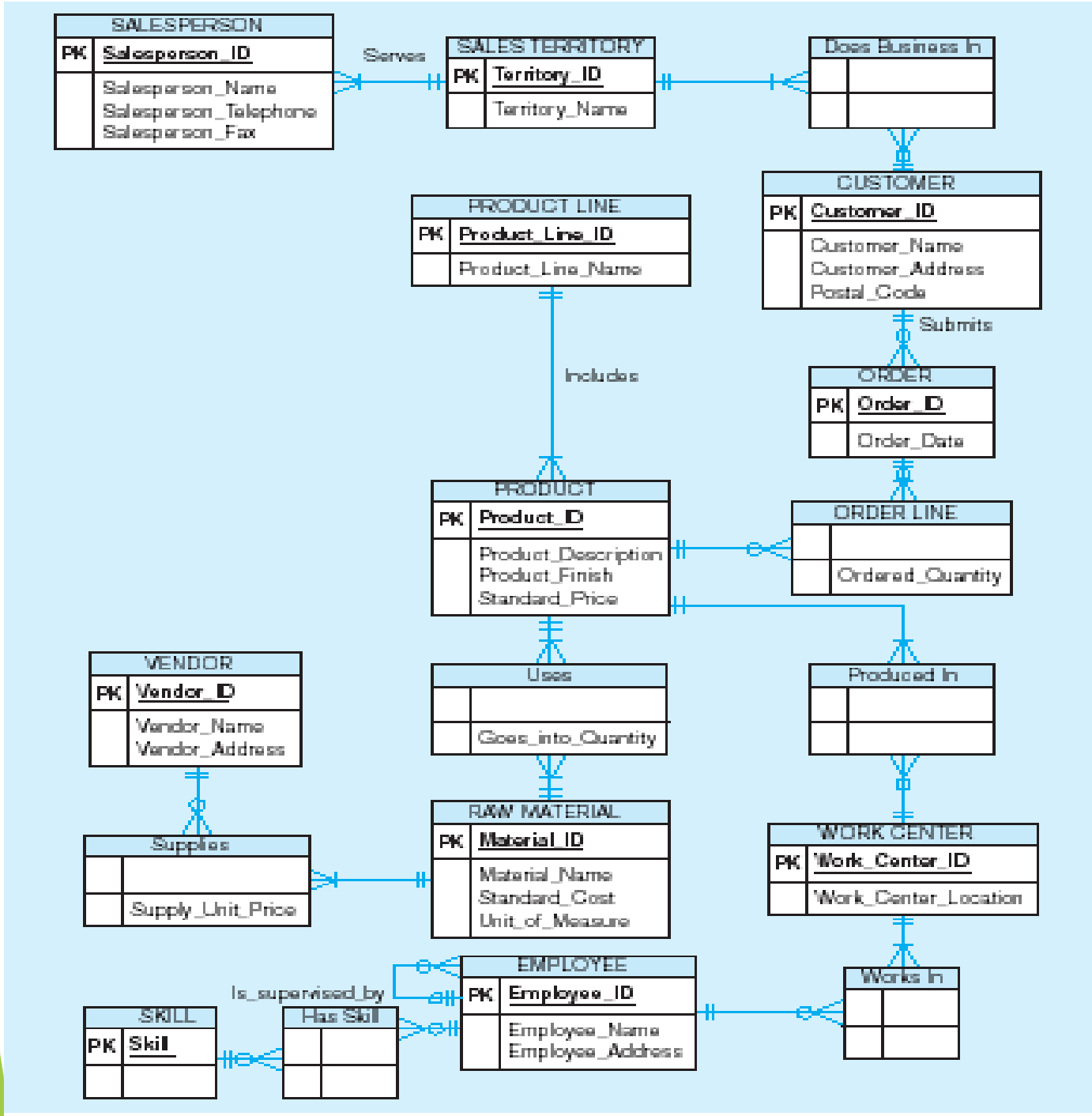
An associative entity – bill of materials structure



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

Ternary relationship as an associative entity





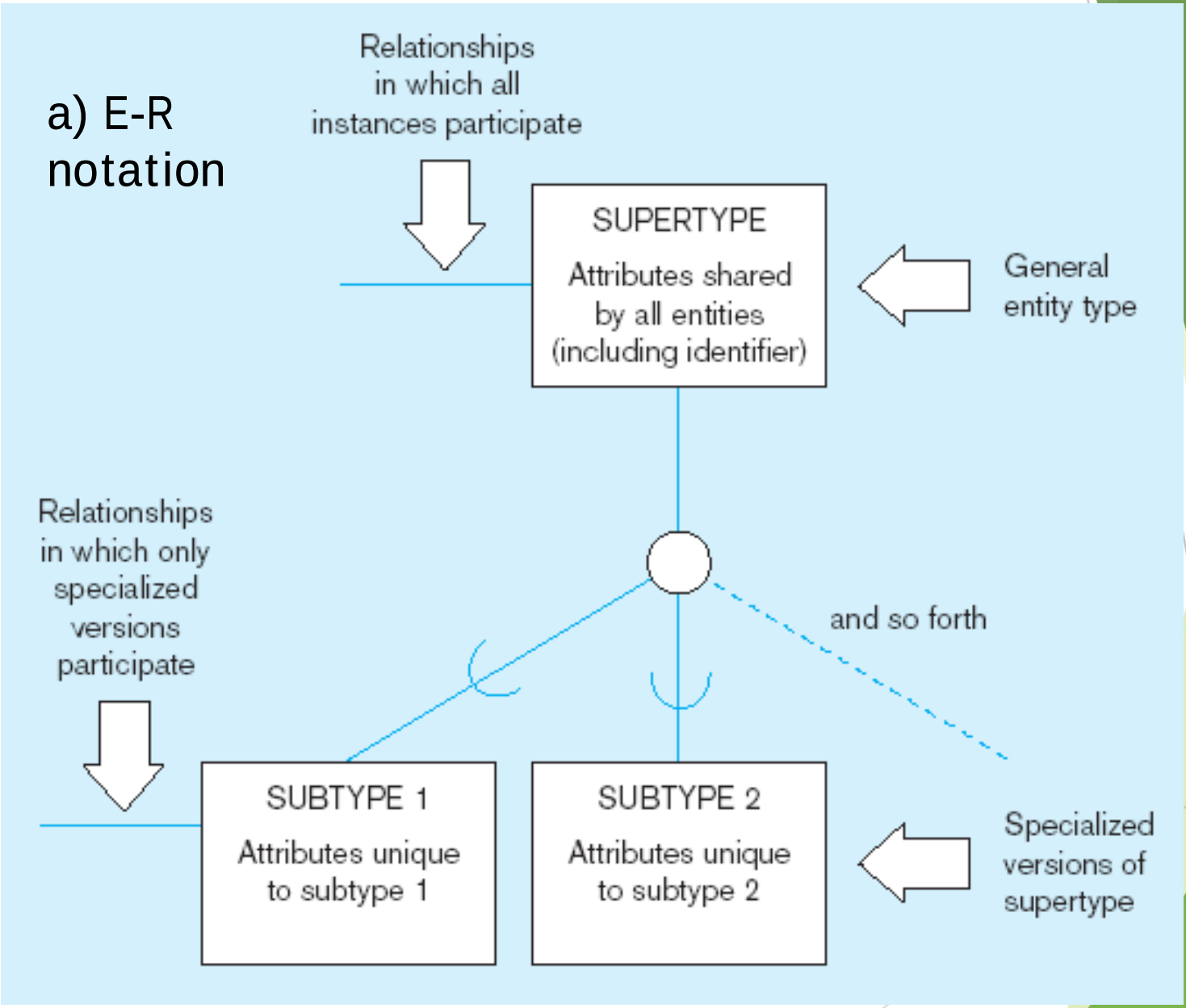
Microsoft Visio Notation for Pine Valley Furniture E-R diagram

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

Supertypes and Subtypes

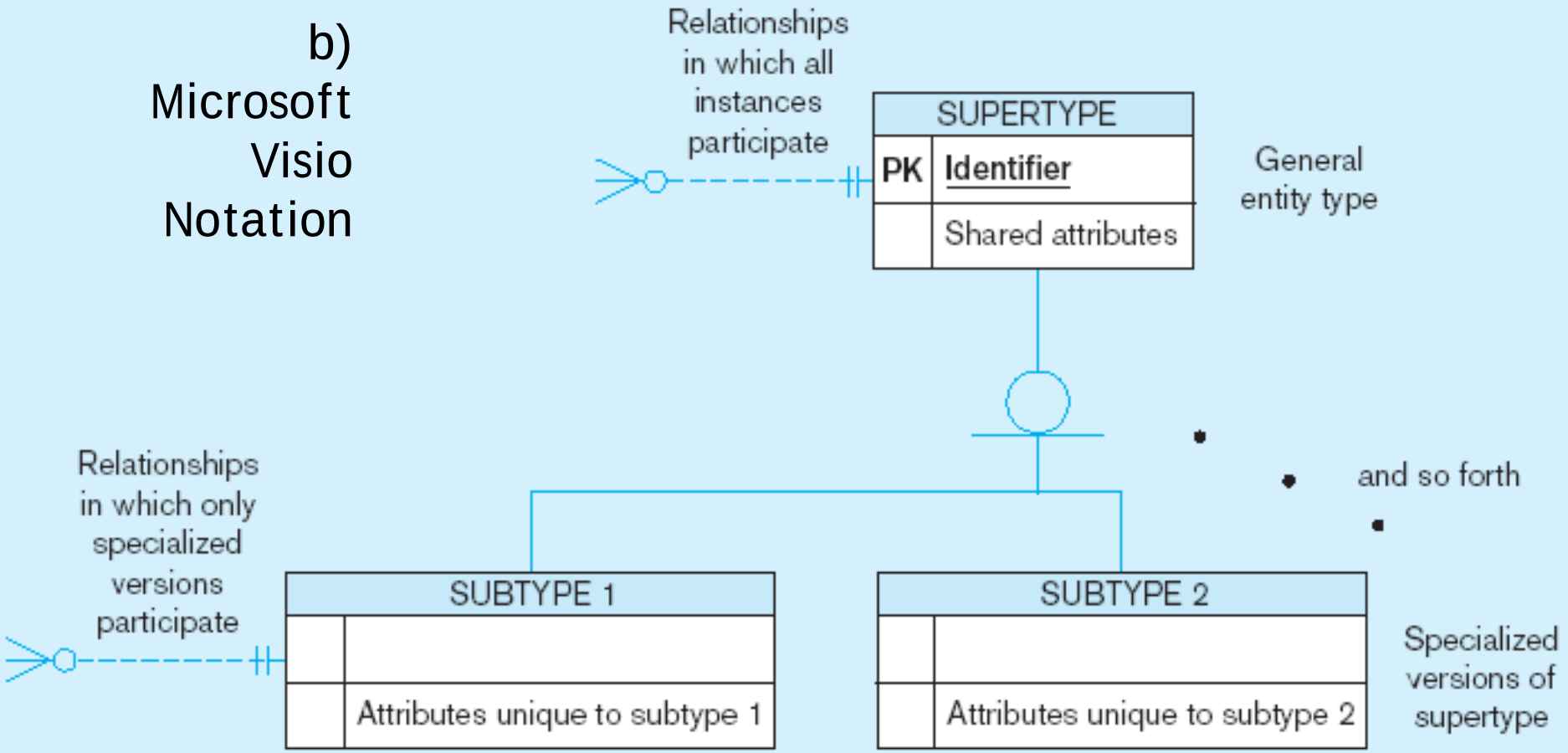
- Subtype: A subgrouping of the entities in an entity type that has attributes distinct from those in other subgroupings
- Supertype: A generic entity type that has a relationship with one or more subtypes
- Attribute Inheritance:
  - Subtype entities inherit values of all attributes of the supertype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

Basic notation for supertype/subtype notation



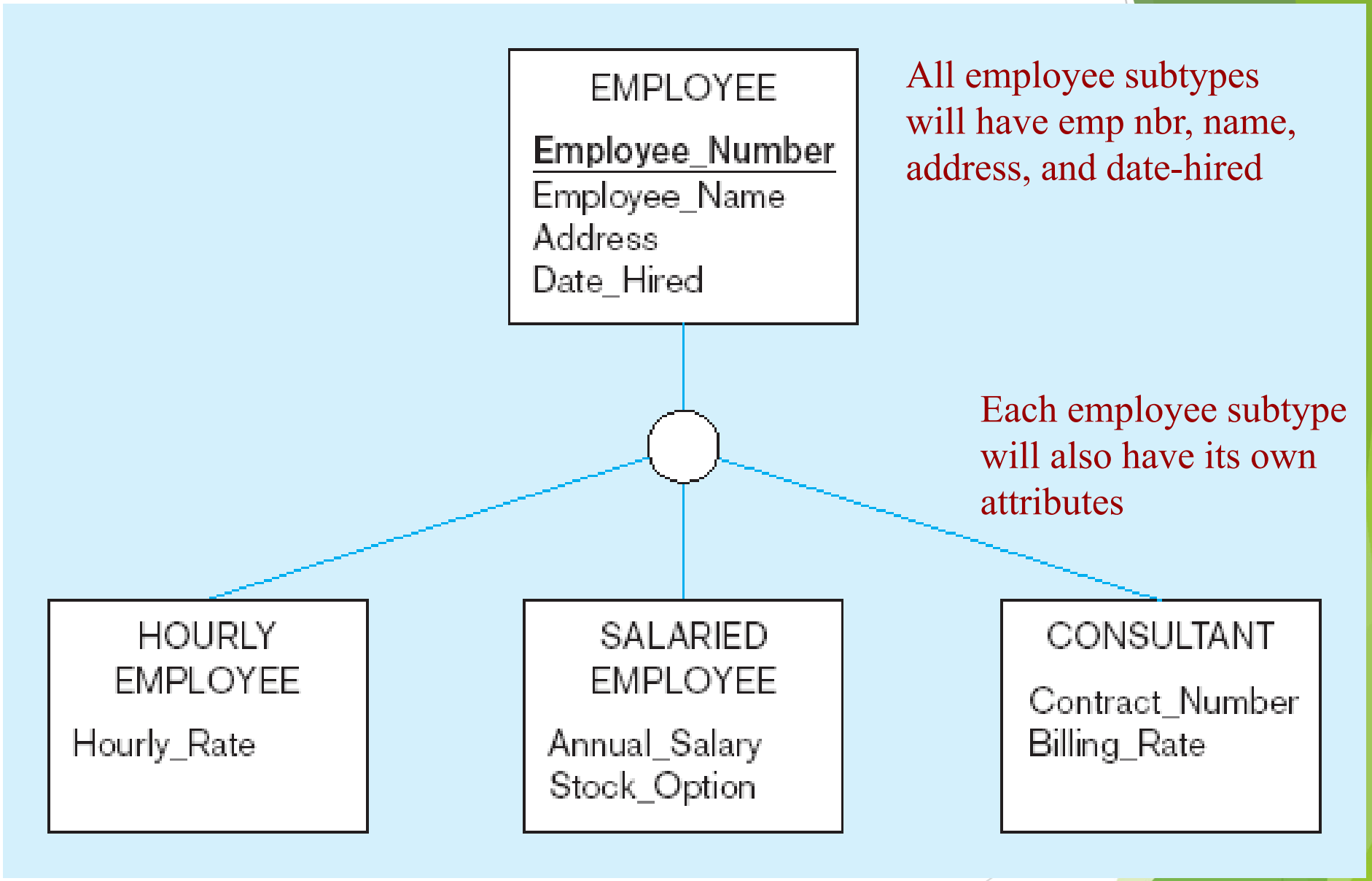
Basic notation for supertype/subtype notation (cont.)

b) Microsoft Visio Notation



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

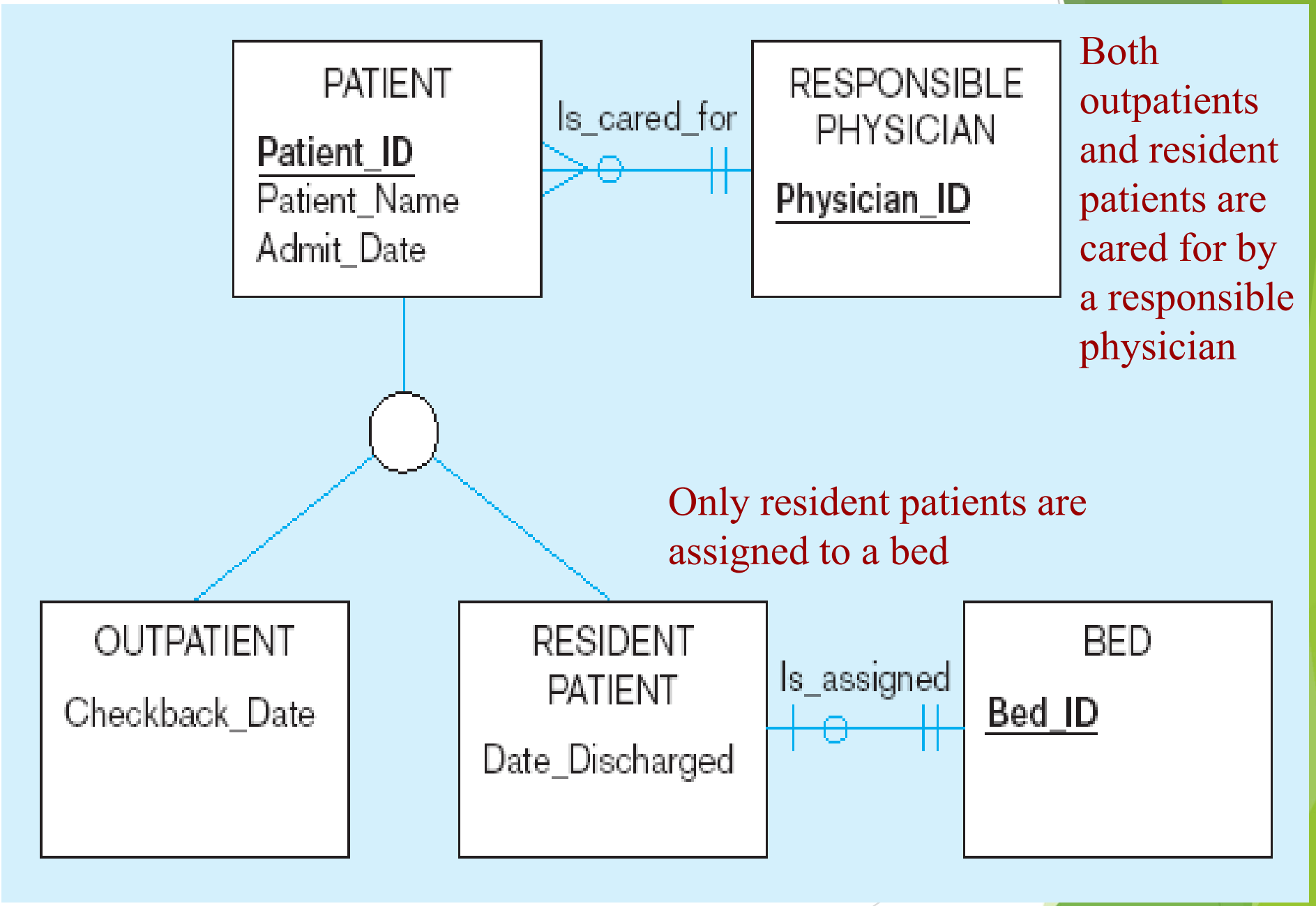
Employee supertype with three subtypes



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

Supertype/subtype relationships in a hospital

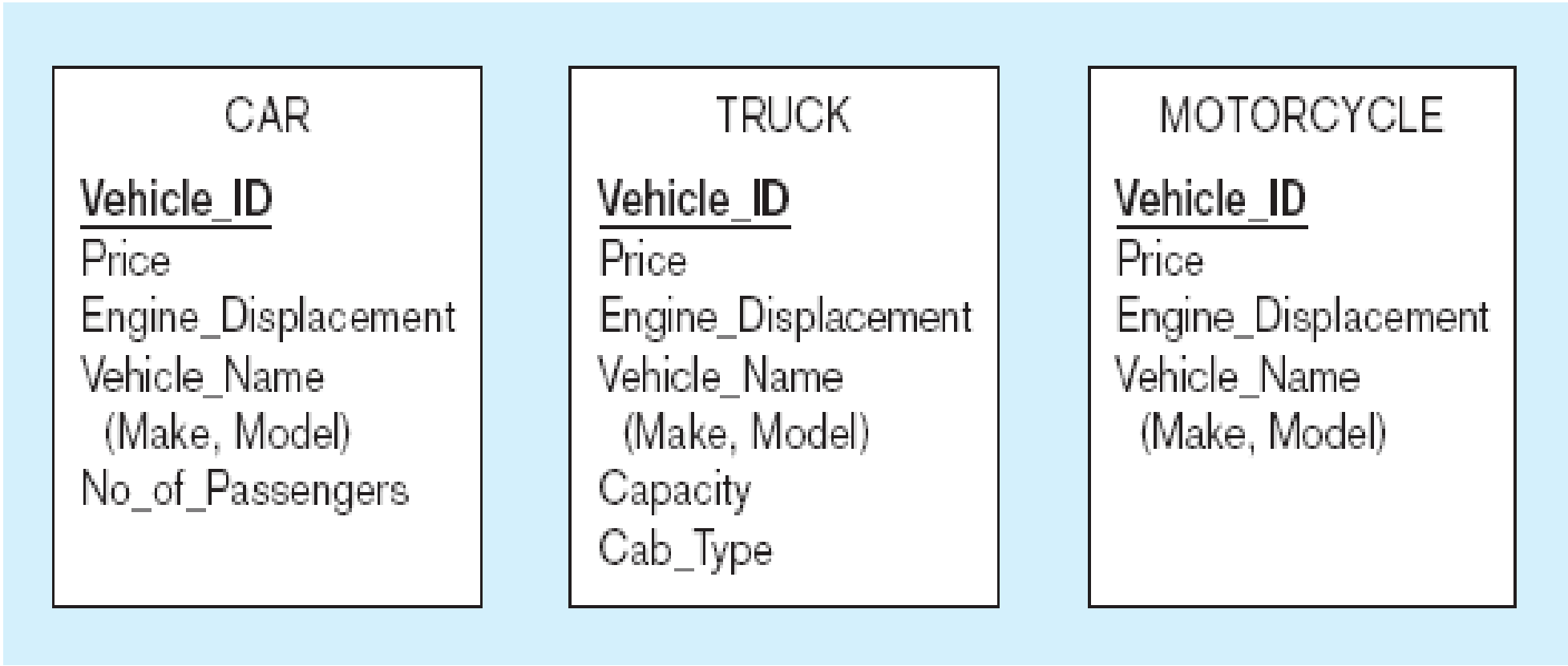


Generalization and Specialization

- *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

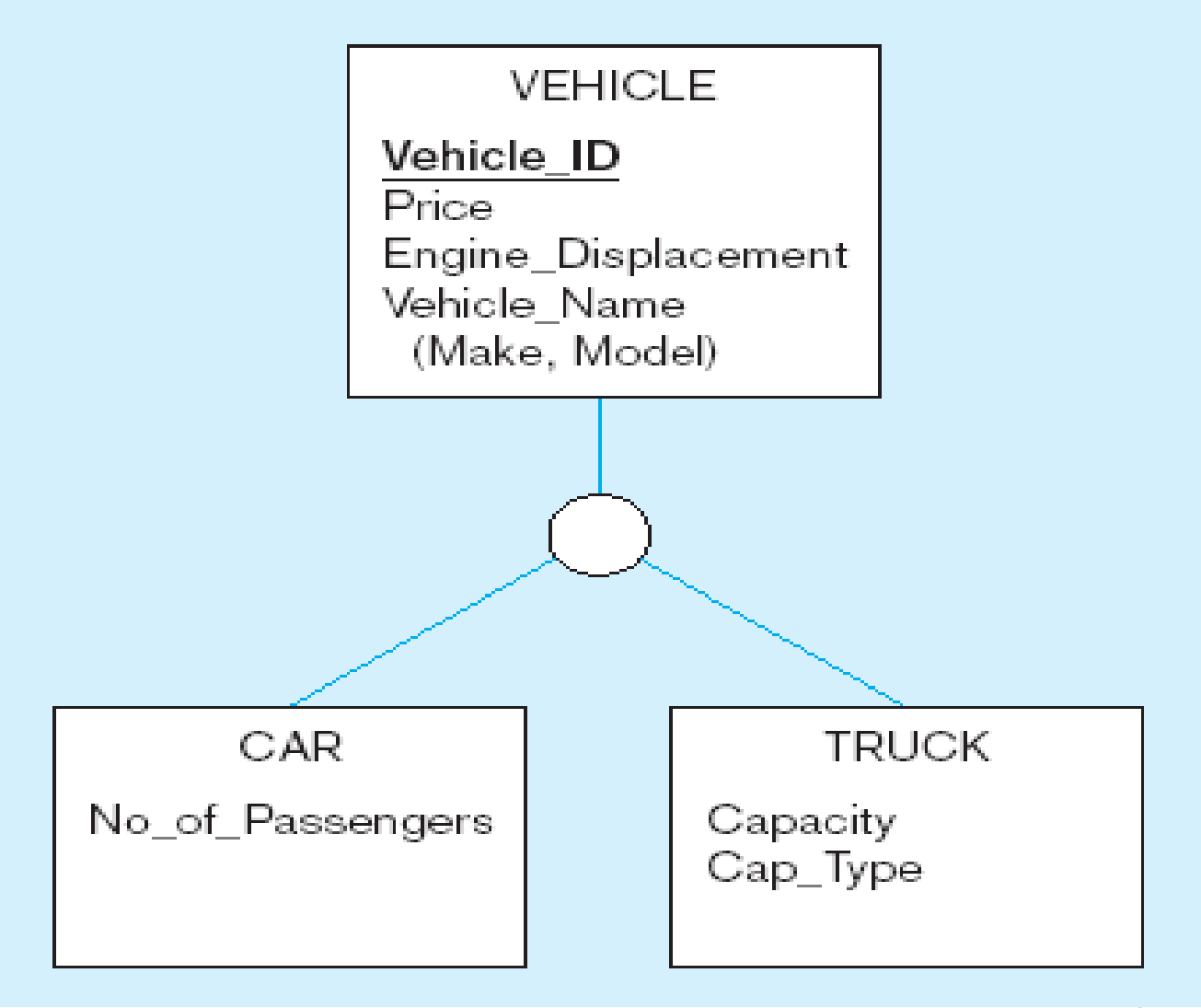
a) Three entity types: CAR, TRUCK, and MOTORCYCLE



All these types of vehicles have common attributes

Example of generalization (cont.)

b) Generalization to VEHICLE supertype

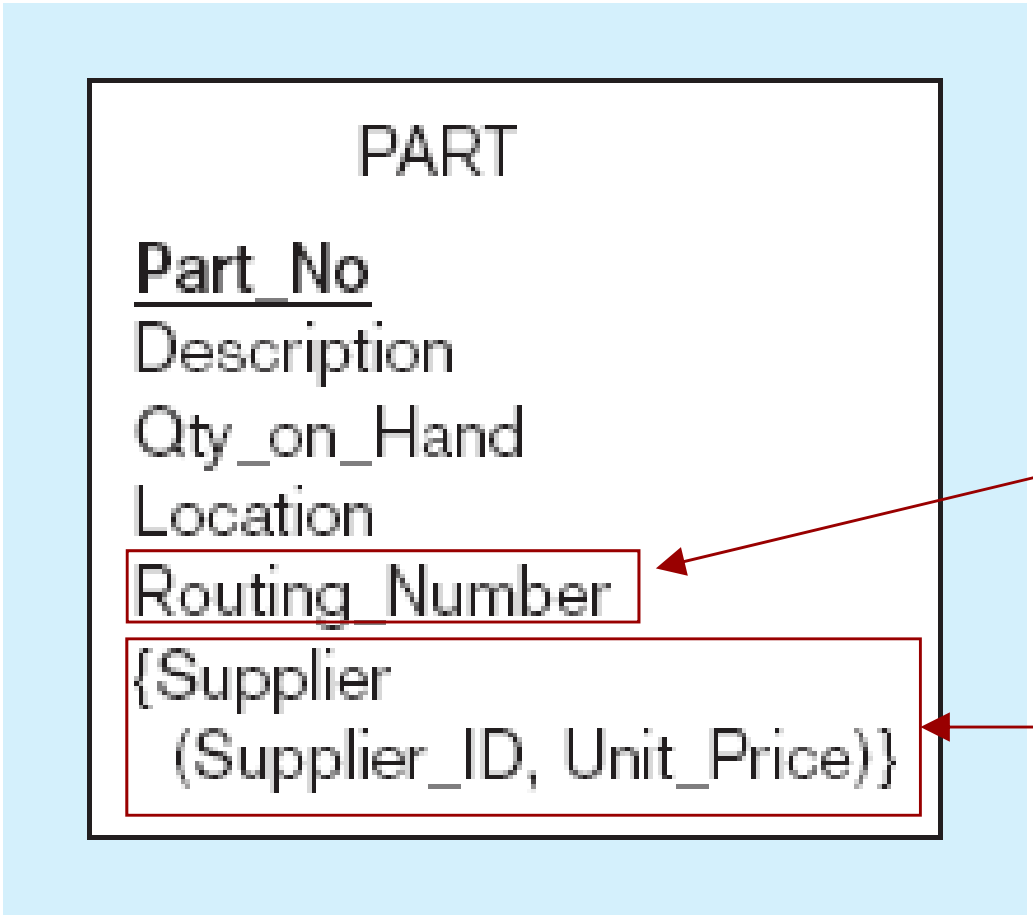


So we put the shared attributes in a supertype

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

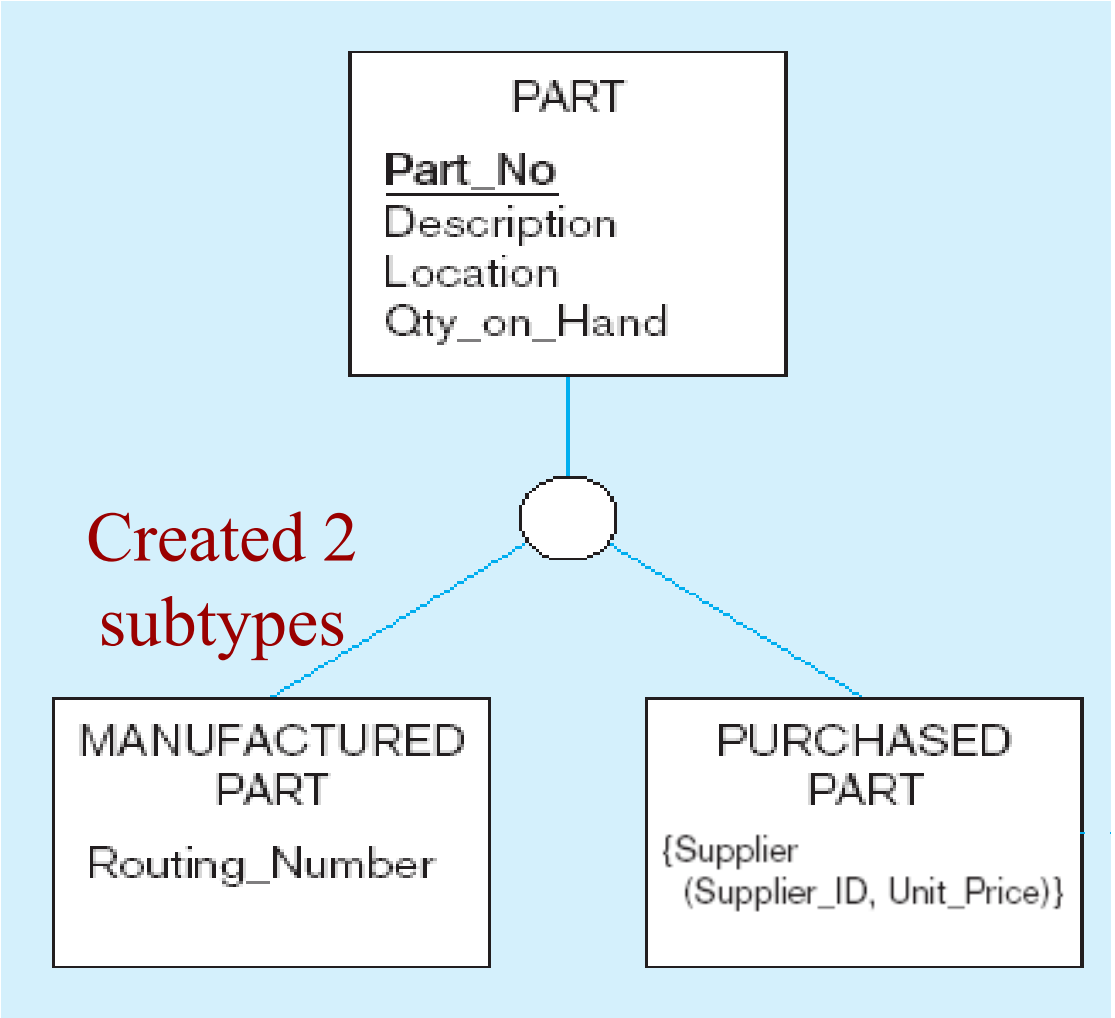
Example of specialization

a) Entity type PART



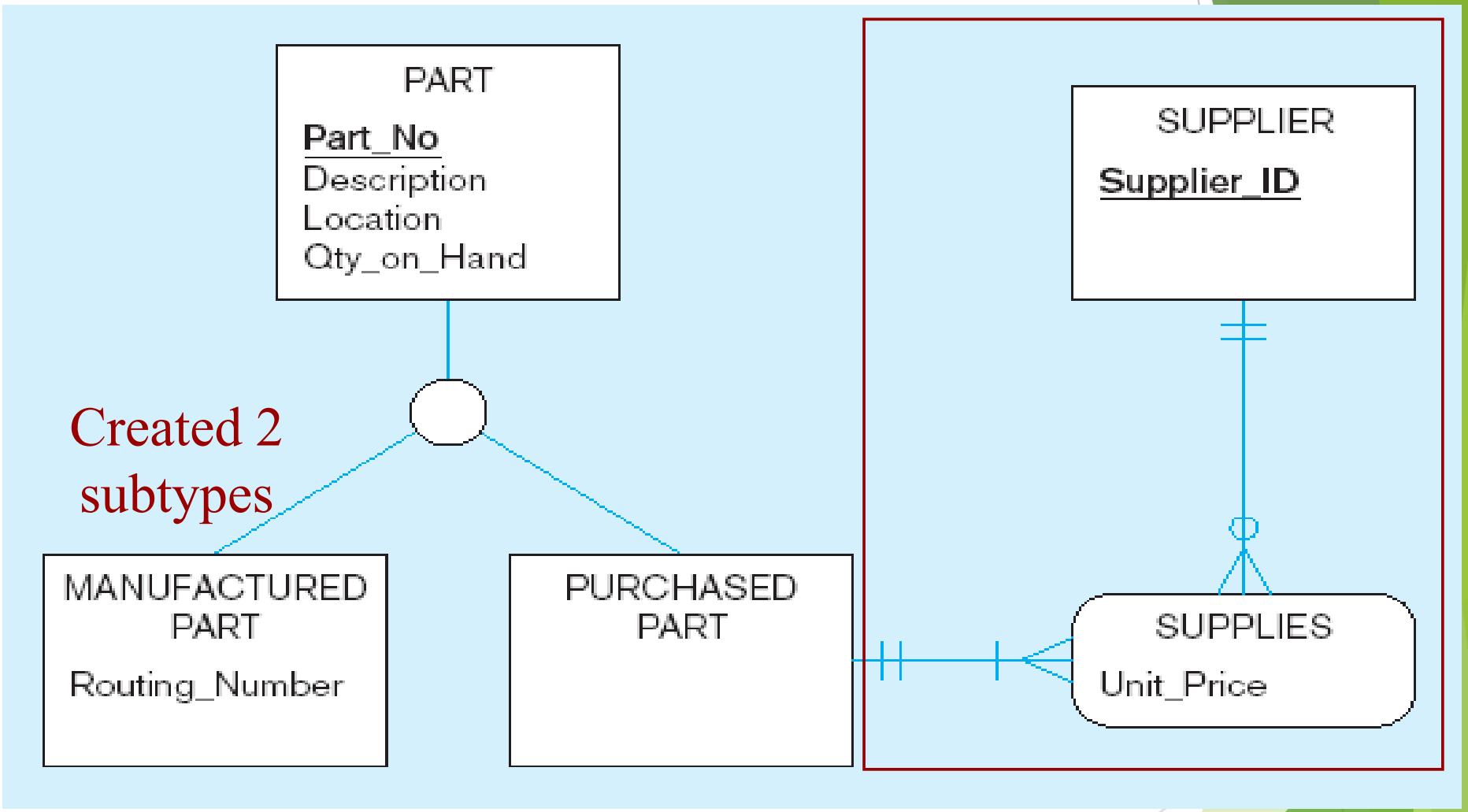
Example of specialization (cont.)

b) Specialization to MANUFACTURED PART and PURCHASED PART



Example of specialization (cont.)

b) Specialization to MANUFACTURED PART and PURCHASED PART



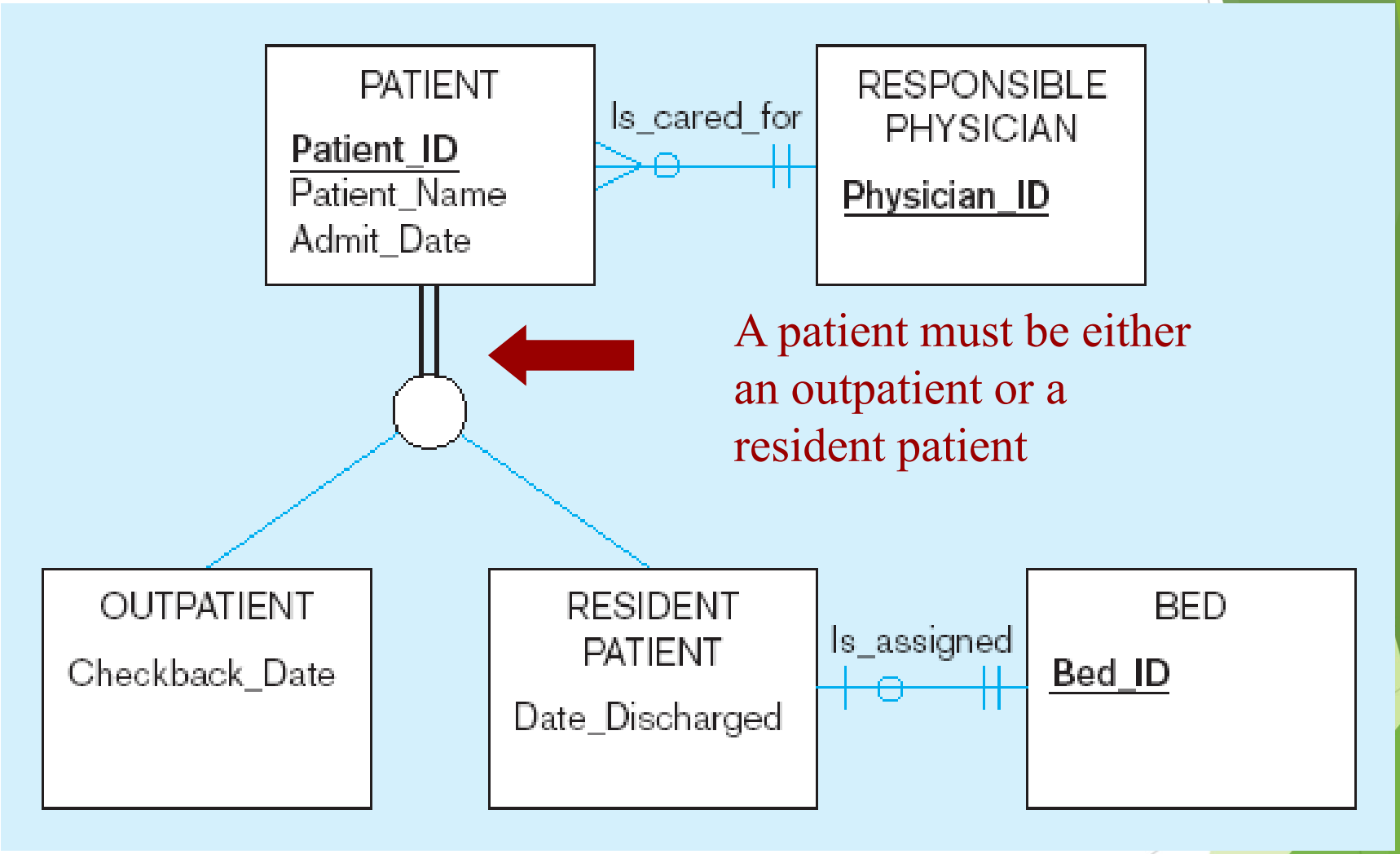
Note: multivalued attribute can be replaced by an associative entity relationship to another entity

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)

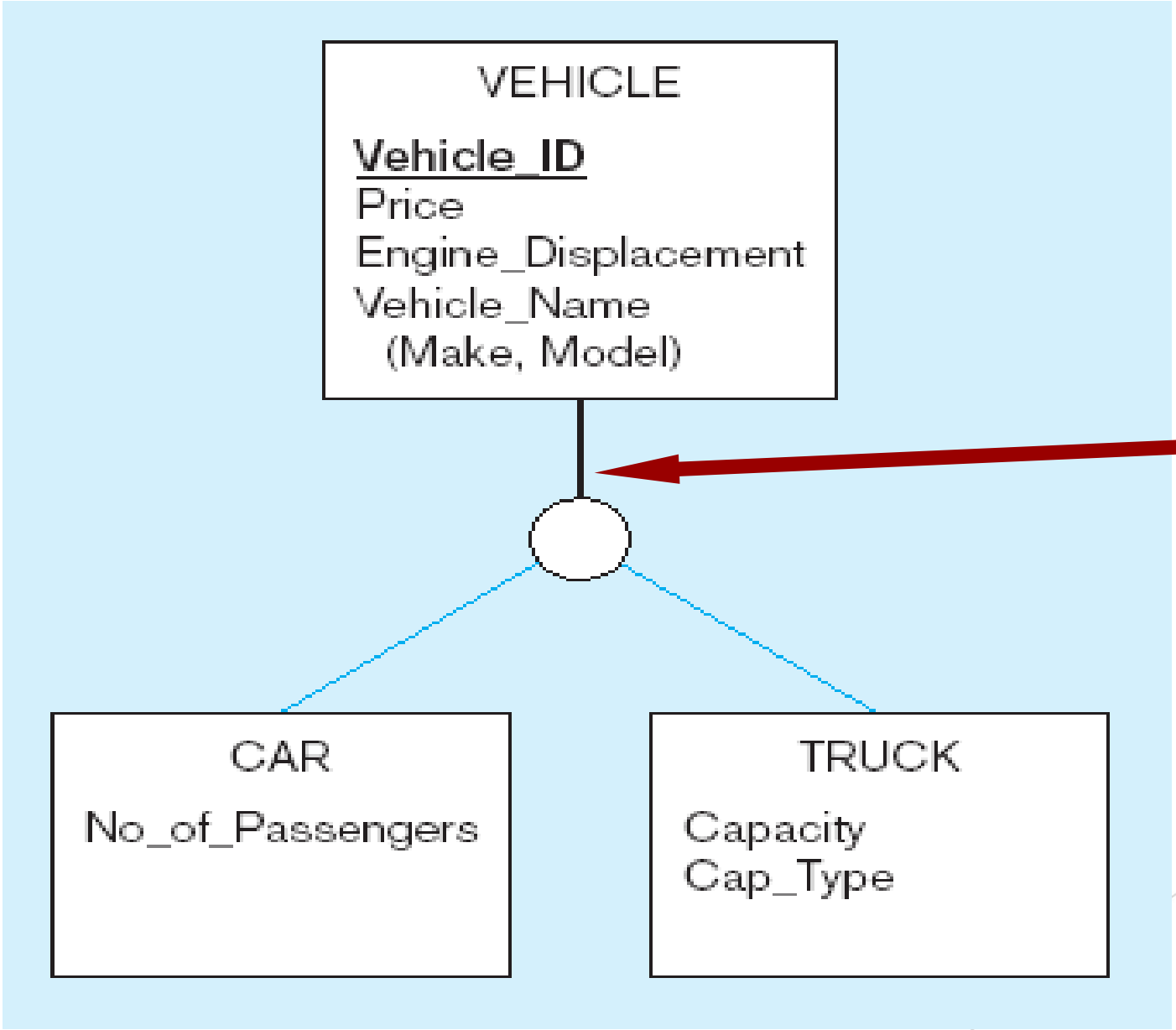
Examples of completeness constraints

a) Total specialization rule



Examples of completeness constraints (cont.)

b) Partial specialization rule



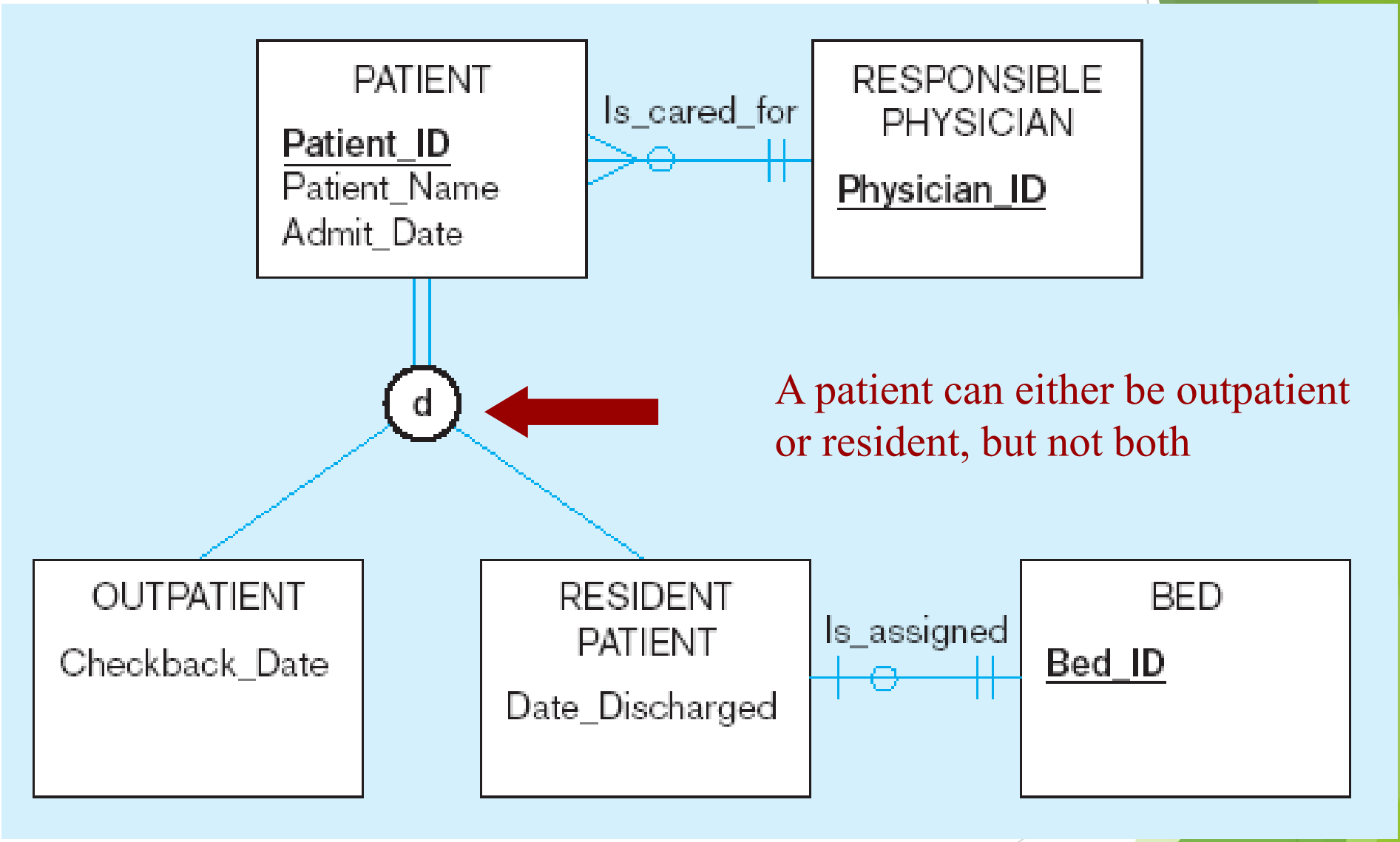
A vehicle could be a car, a truck, or neither

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

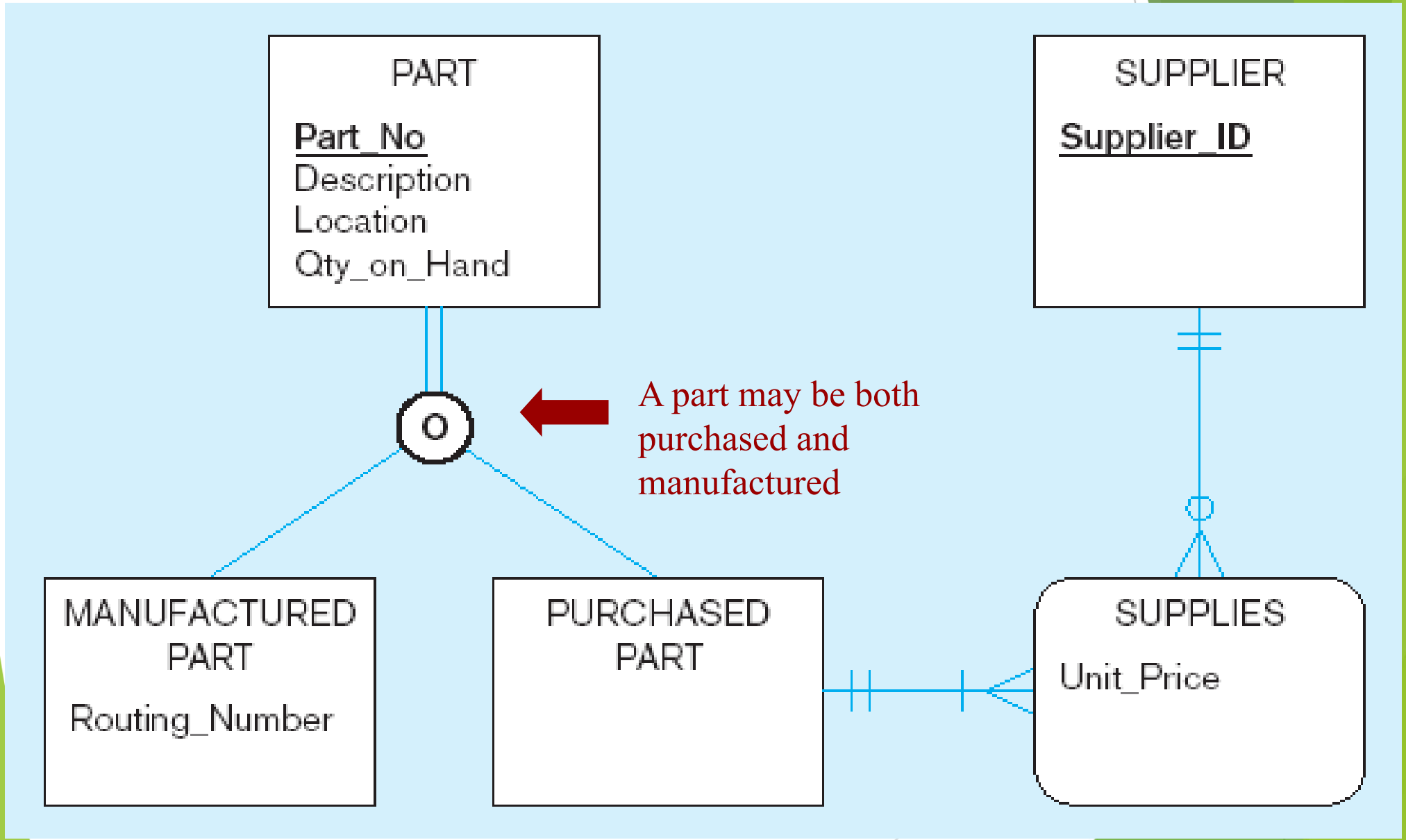
Examples of disjointness constraints

a) Disjoint rule



Examples of disjointness constraints (cont.)

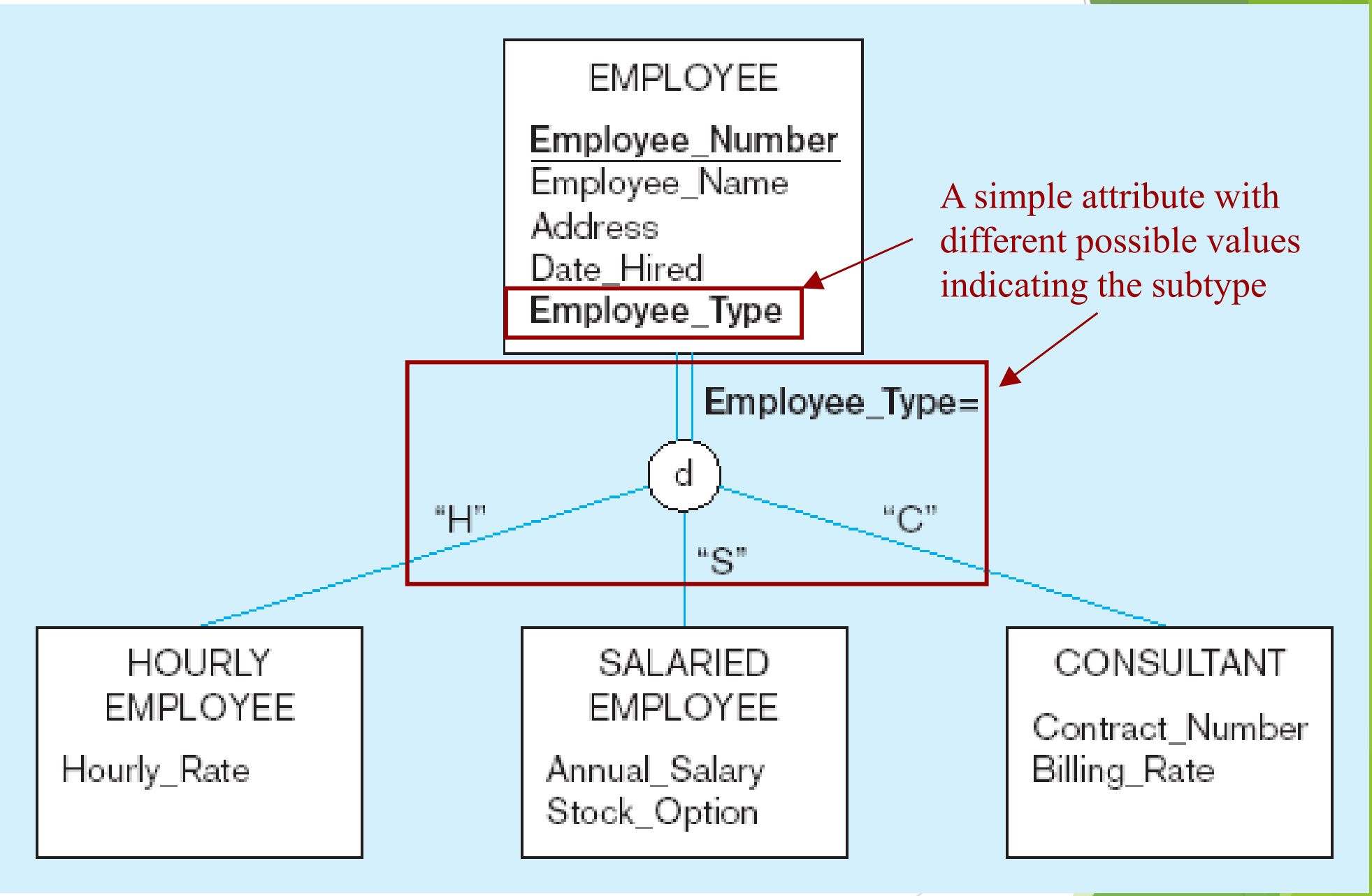
b) Overlap rule



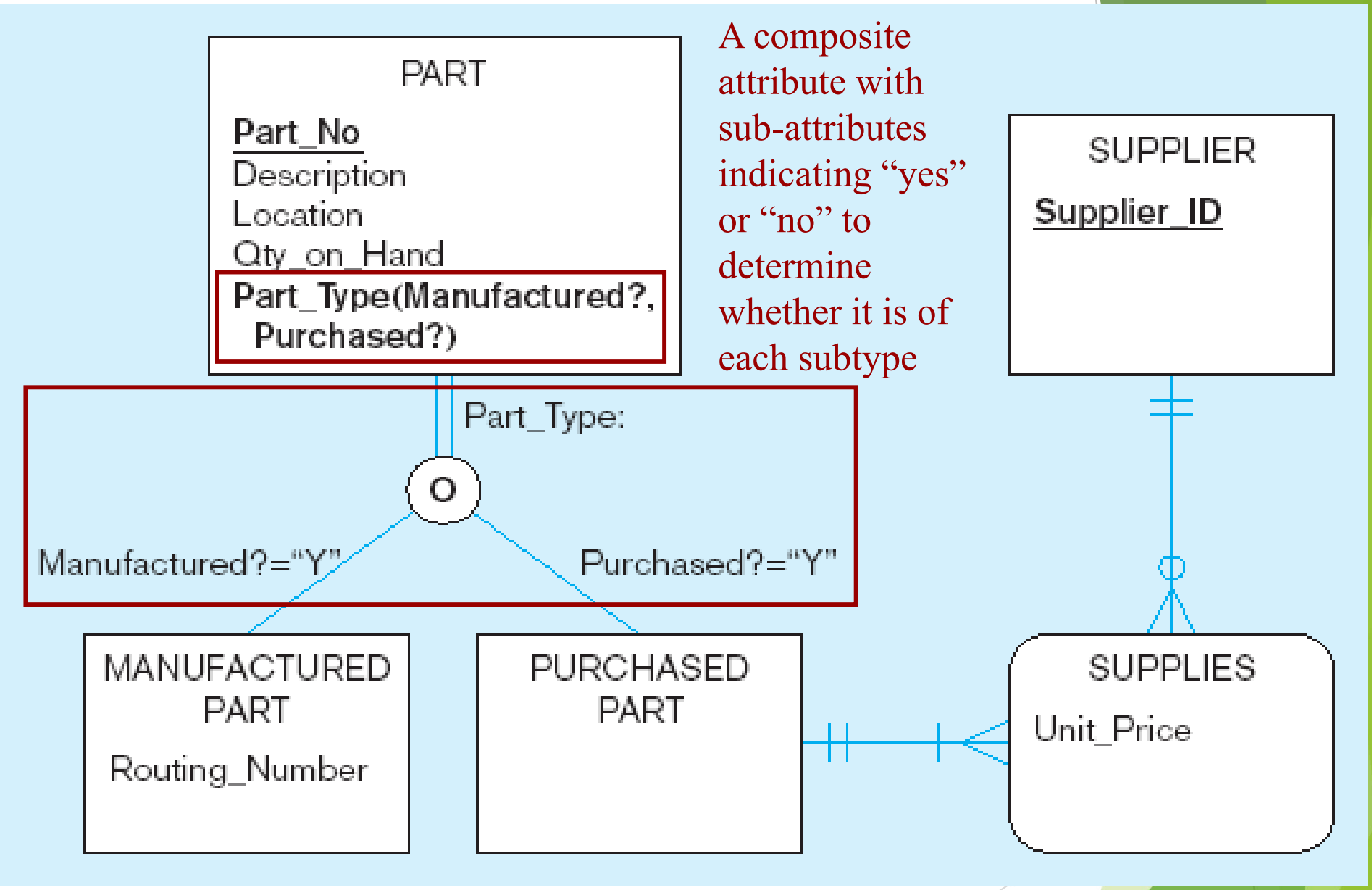
Constraints in Supertype/ Subtype Discriminators

- *Subtype Discriminator*: An attribute of the supertype whose values determine the target subtype(s)
  - If we have the **Disjoint** rule—a *simple* attribute with alternative values to indicate the possible subtypes
  - If we have the **Overlapping** rule—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

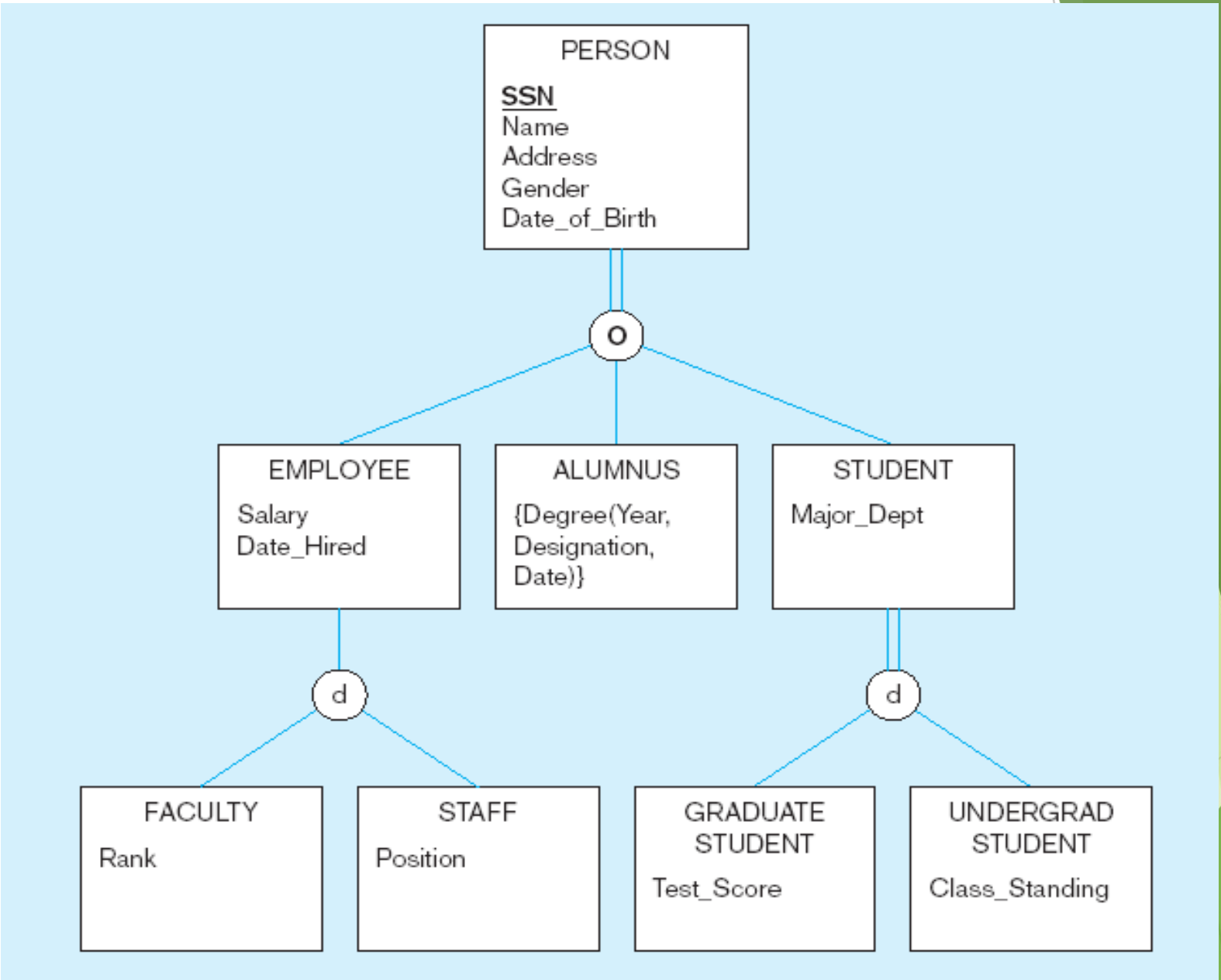
Introducing a subtype discriminator (***disjoint*** rule)



Subtype discriminator (***overlap*** rule)



Example of supertype/subtype hierarchy

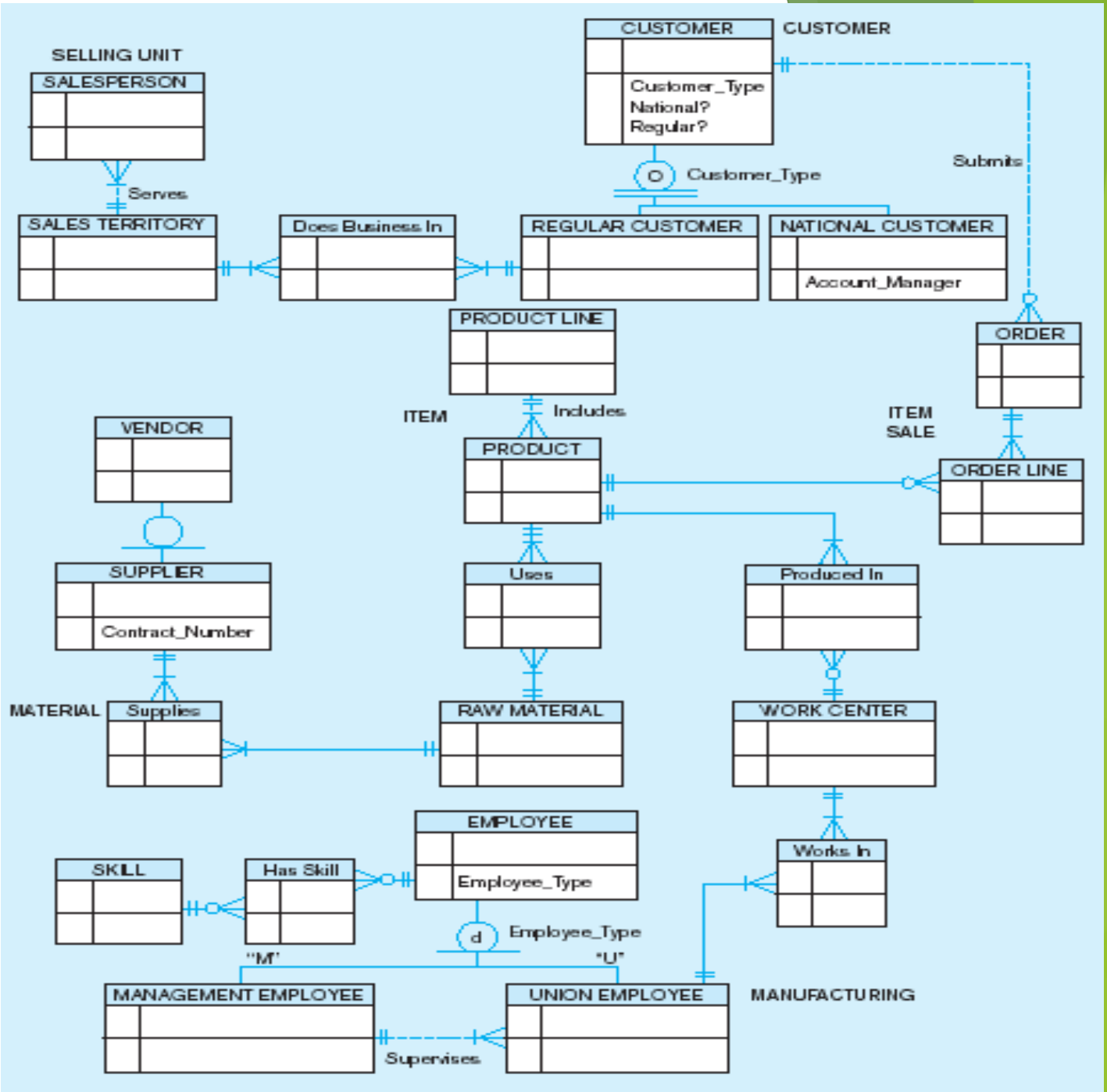


Entity Clusters

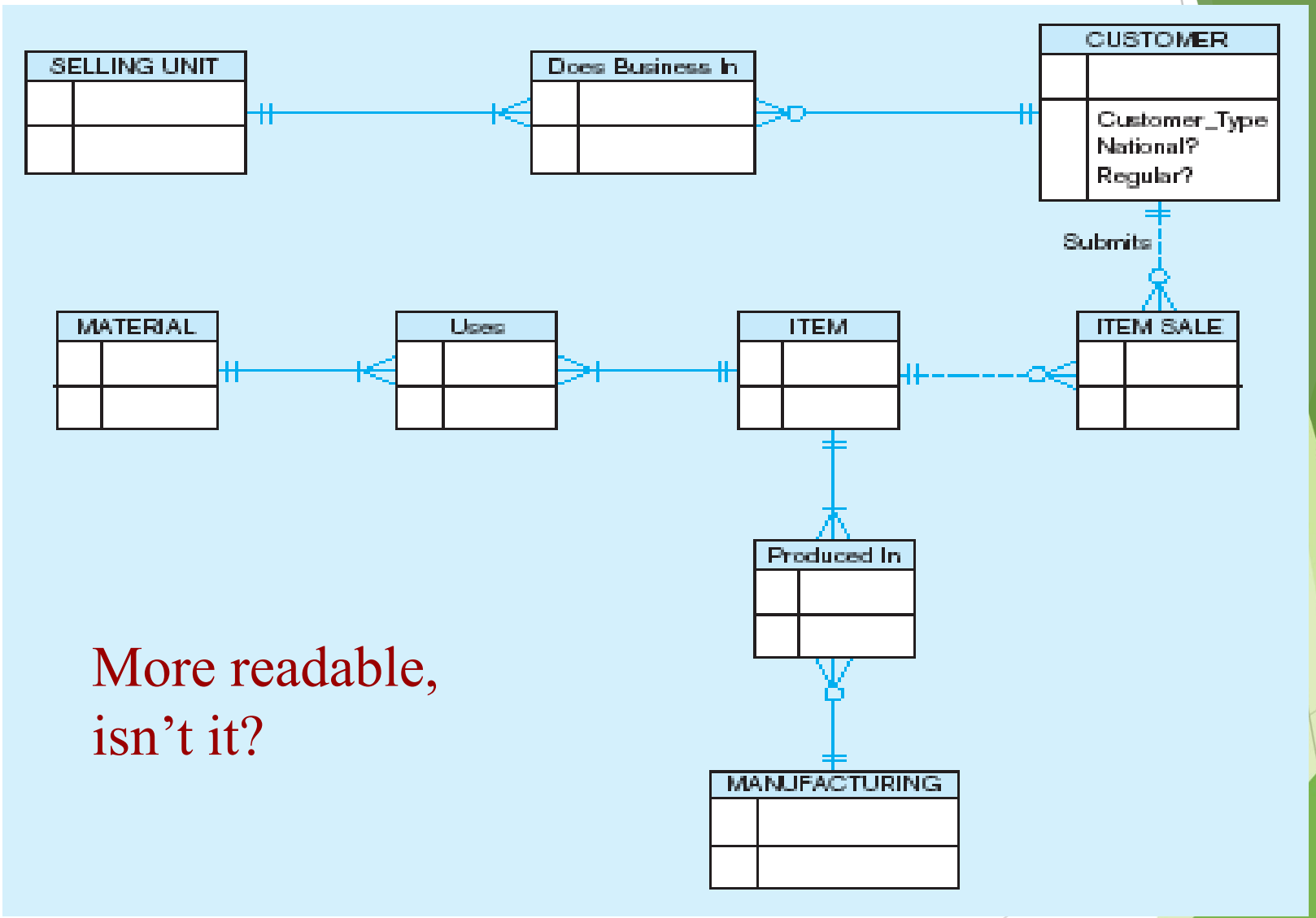
- E-R diagrams are difficult to read when there are too many entities and relationships
- Solution: Group entities and relationships into *entity clusters*
- **Entity cluster**: Set of one or more entity types and associated relationships grouped into a single abstract entity type

Possible entity clusters for Pine Valley Furniture in Microsoft Visio

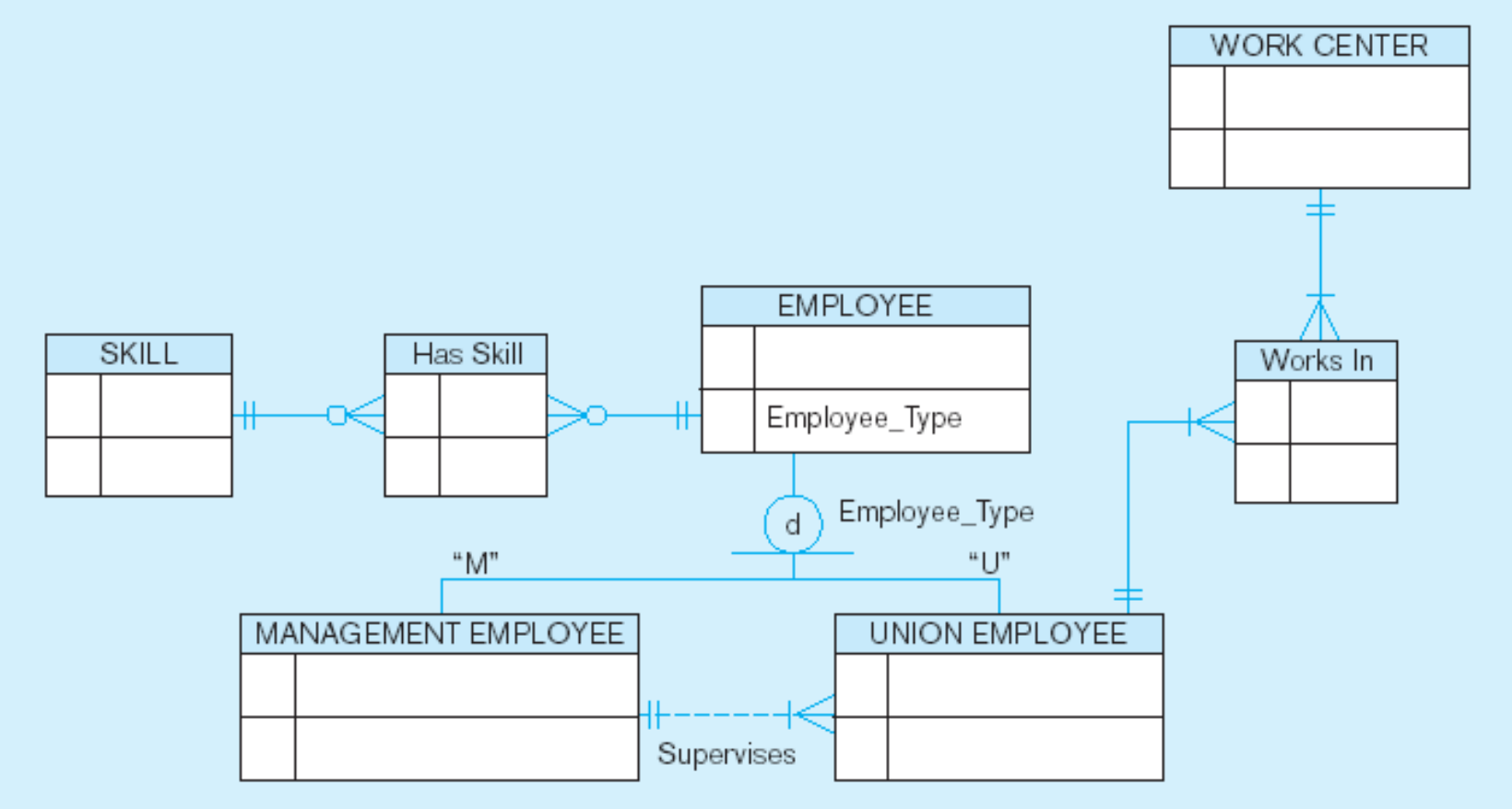
Related groups of entities could become clusters



E-R diagram of PVF entity clusters



Manufacturing entity cluster



Detail for a single cluster

Business Rules

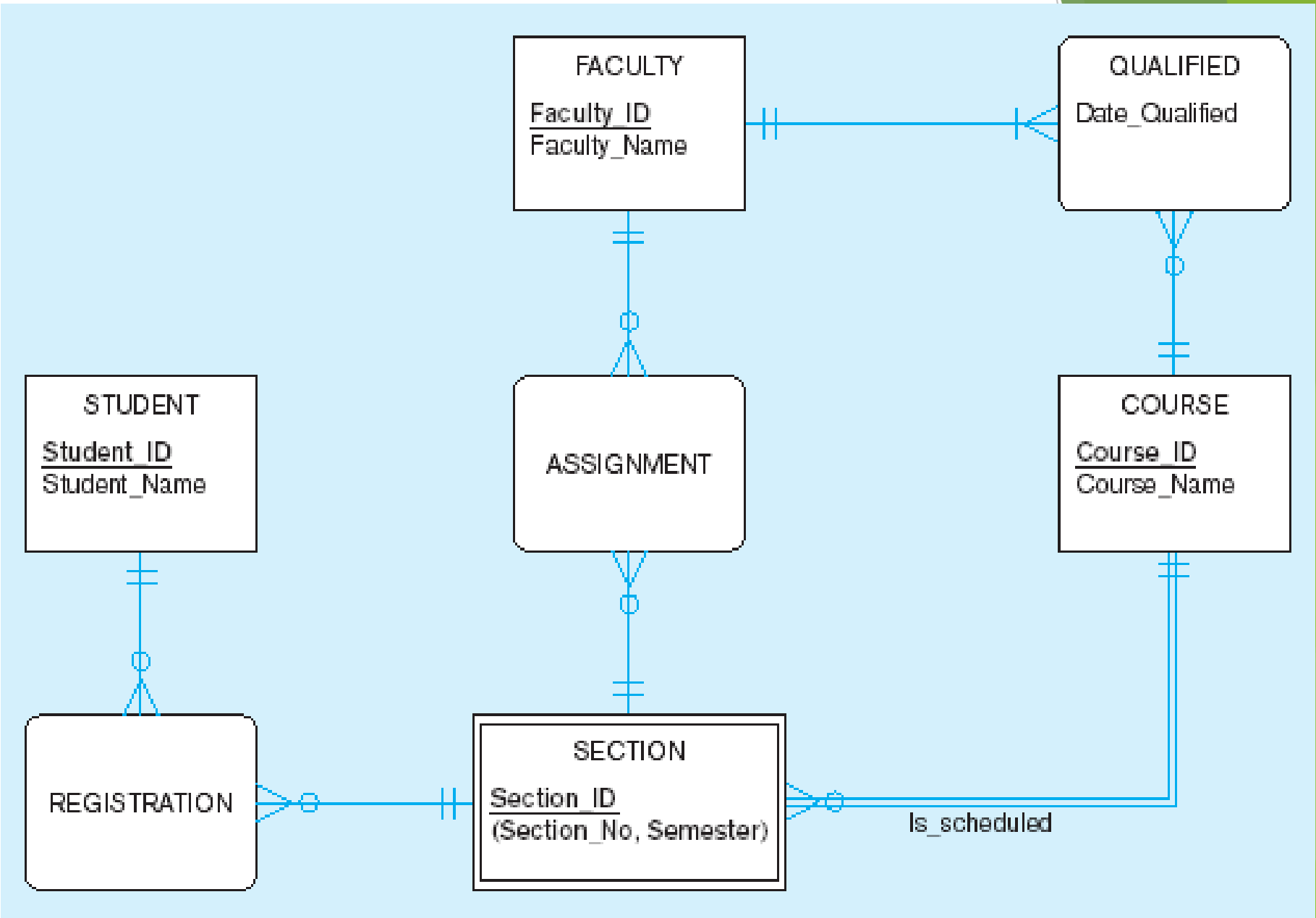
- Statements that *define* or *constrain* some aspect of the business
- Classification of business rules:
  - Structural assertion—rule expressing static structure. Includes attributes, relationships, and definitions
  - Derivation—rule derived from other knowledge, often in the form of a formula using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

Stating an Action Assertion

- Anchor Object—an object on which actions are limited
- Action—creation, deletion, update, or read
- Corresponding Object—an object influencing the ability to perform an action on the anchor object

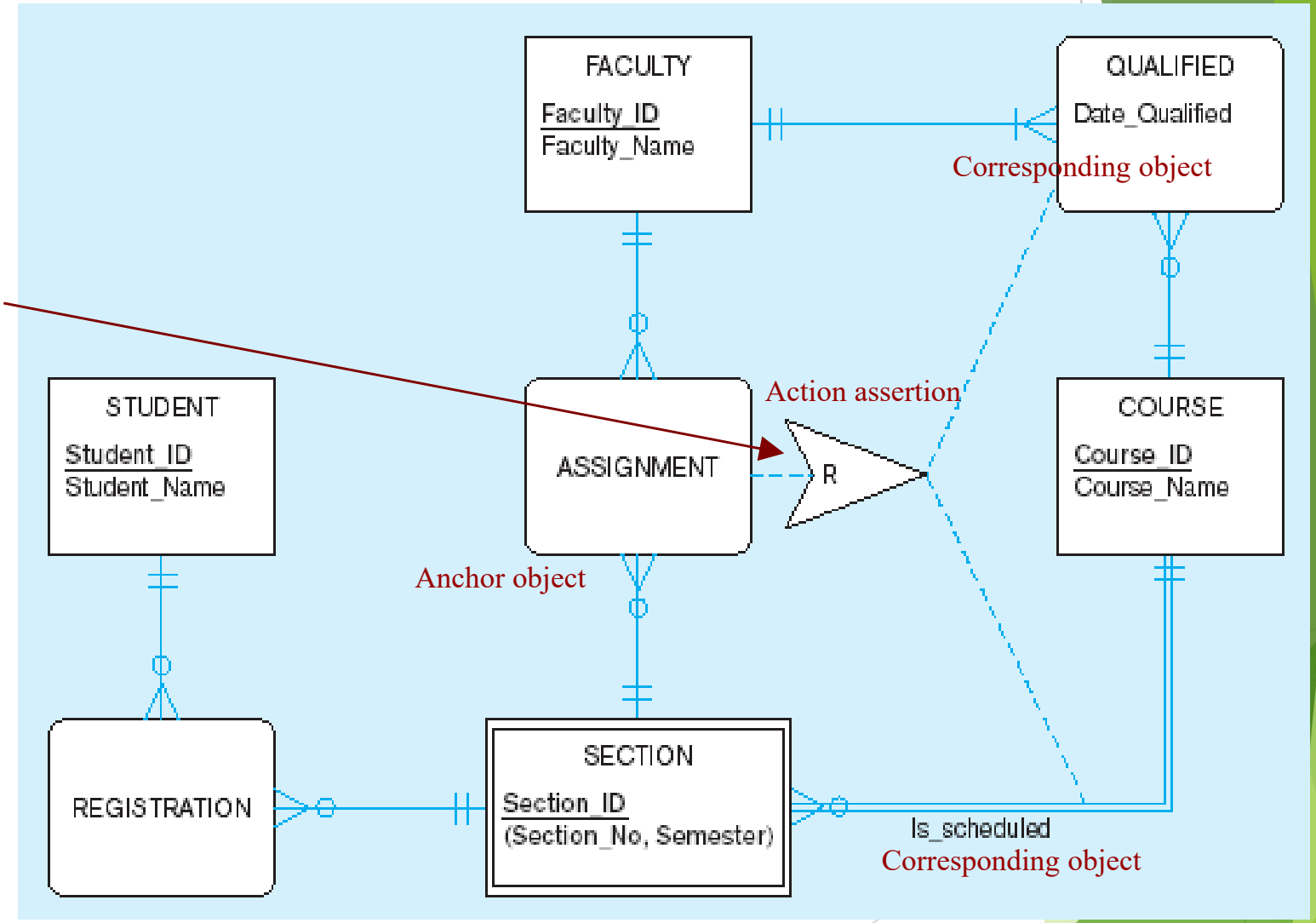
Action assertions identify corresponding objects that constrain the ability to perform actions on anchor objects

Data model segment for class scheduling



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



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 ***U**pper **LIM**it*