Chapter 3:

Modeling Data in the Organization

Objectives

- Definition of terms
- 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

Business Rules

- Statements that define or constrain some aspect of the business
- Assert business structure
- Control/influence business behavior
- Expressed in terms familiar to end users
- Automated through DBMS software

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

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
- Follows standard syntax

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
  - Concise description of essential data meaning
  - Achieved by consensus, and iteratively refined

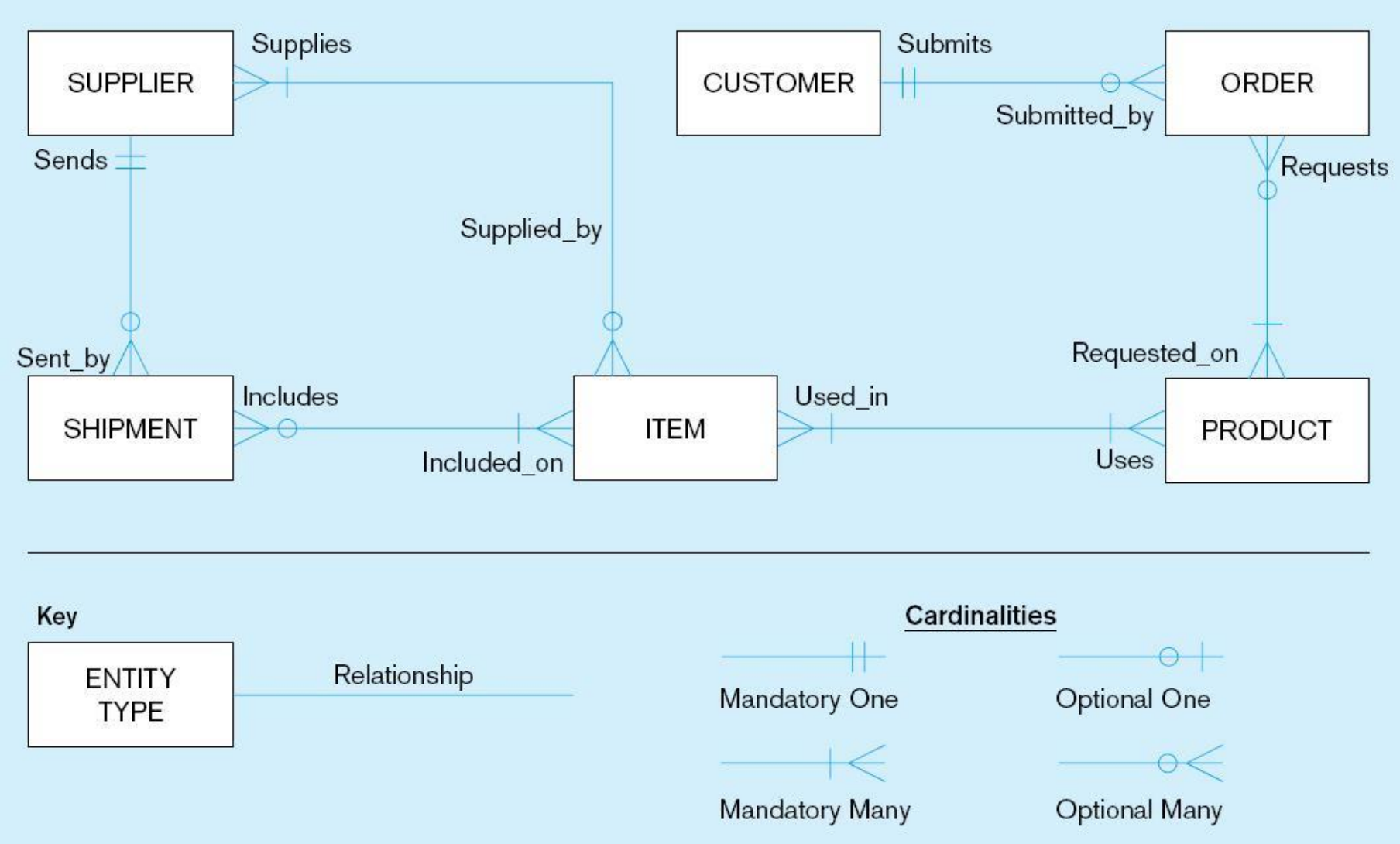
E-R Model Constructs

- 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 (Figure 3-1)

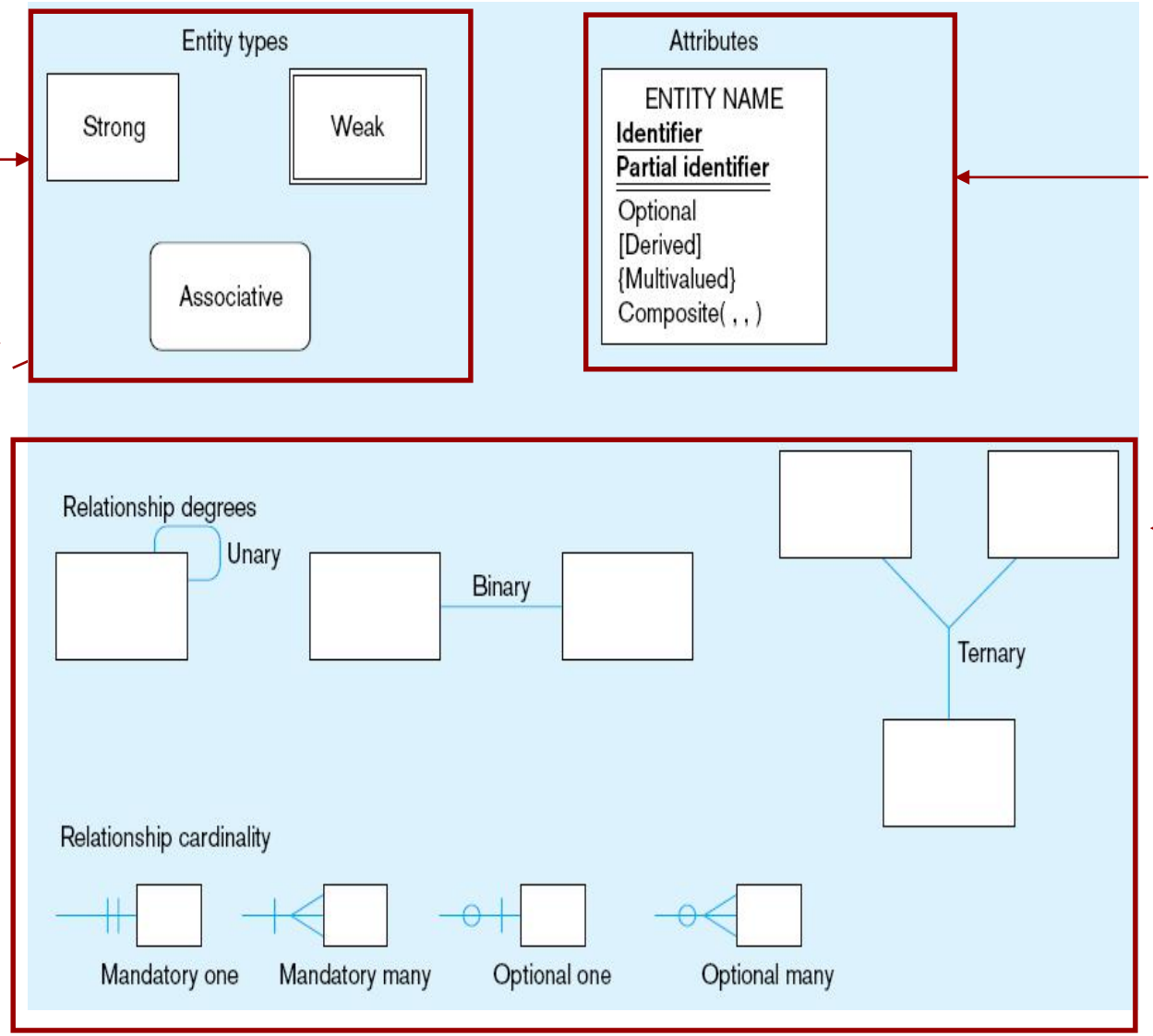


Basic E-R notation (Figure 3-2)

Entity symbols

A special entity that is also a relationship

Relationship degrees specify number of entity types involved



Attribute symbols

Relationship symbols

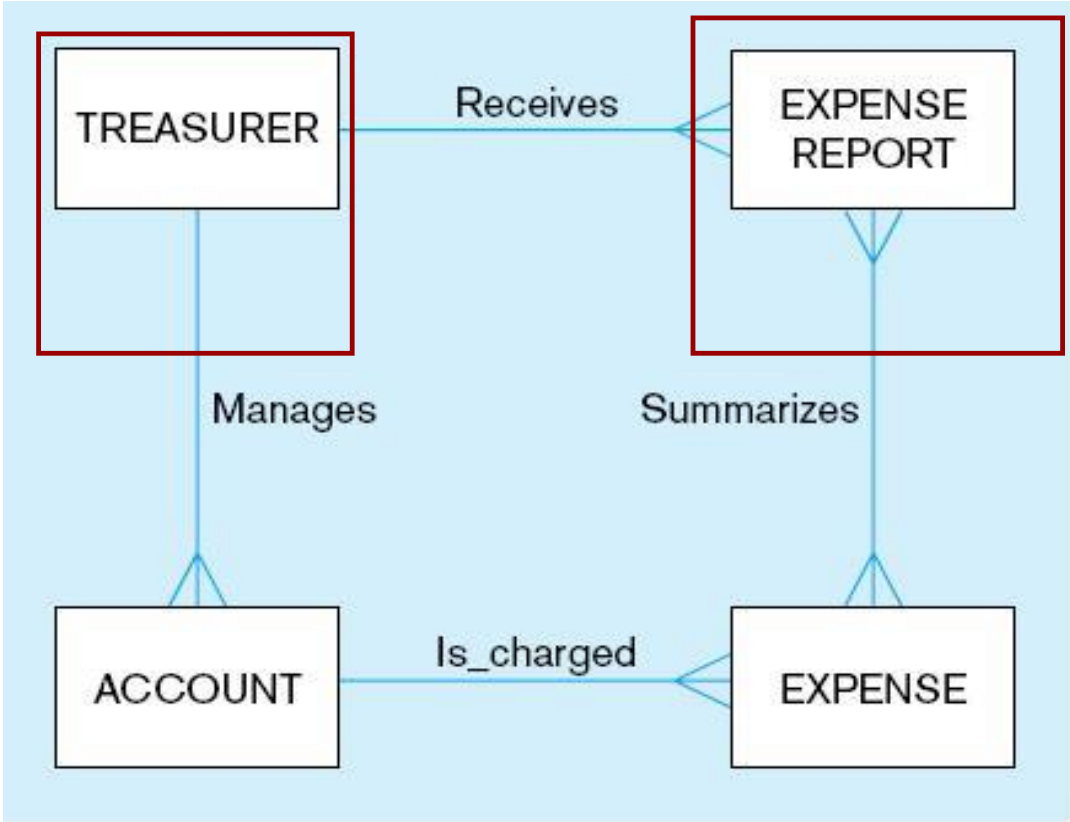
Relationship cardinalities specify how many of each entity type is allowed

What Should an Entity Be?

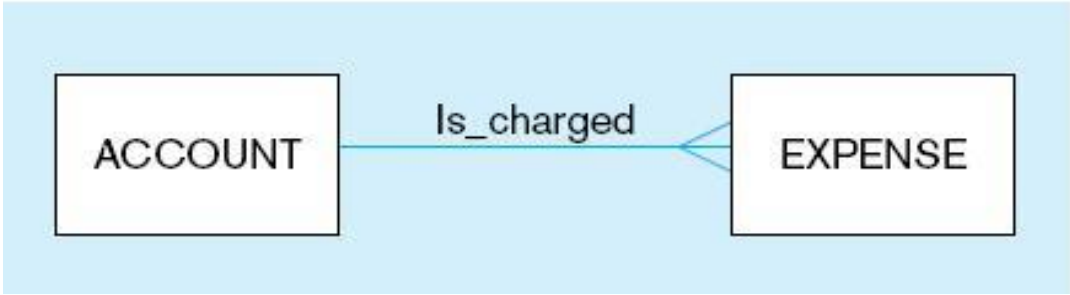
- SHOULD BE:
  - An object that will have many instances in the database
  - An object that will be composed of multiple attributes
  - An object that we are trying to model
- SHOULD NOT BE:
  - A user of the database system
  - An output of the database system (e.g., a report)

Figure 3-4 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

Figure 3-7 A **composite** attribute

An attribute broken into component parts

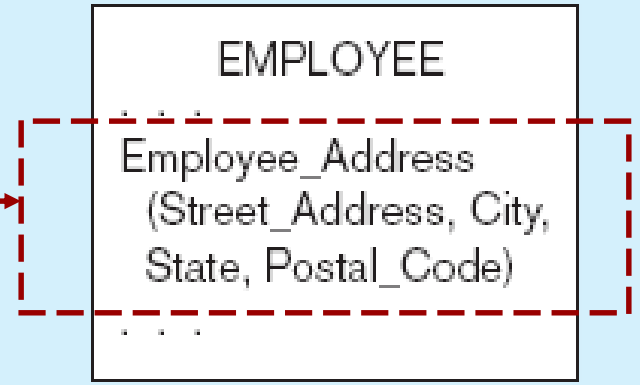


Figure 3-8 Entity with **multivalued** attribute (Skill) and **derived** attribute (Years_Employed)

Multivalued
an employee can have more than one skill

Derived
from date employed and current date

Figure 3-9 Simple and composite identifier attributes

The identifier is boldfaced and underlined

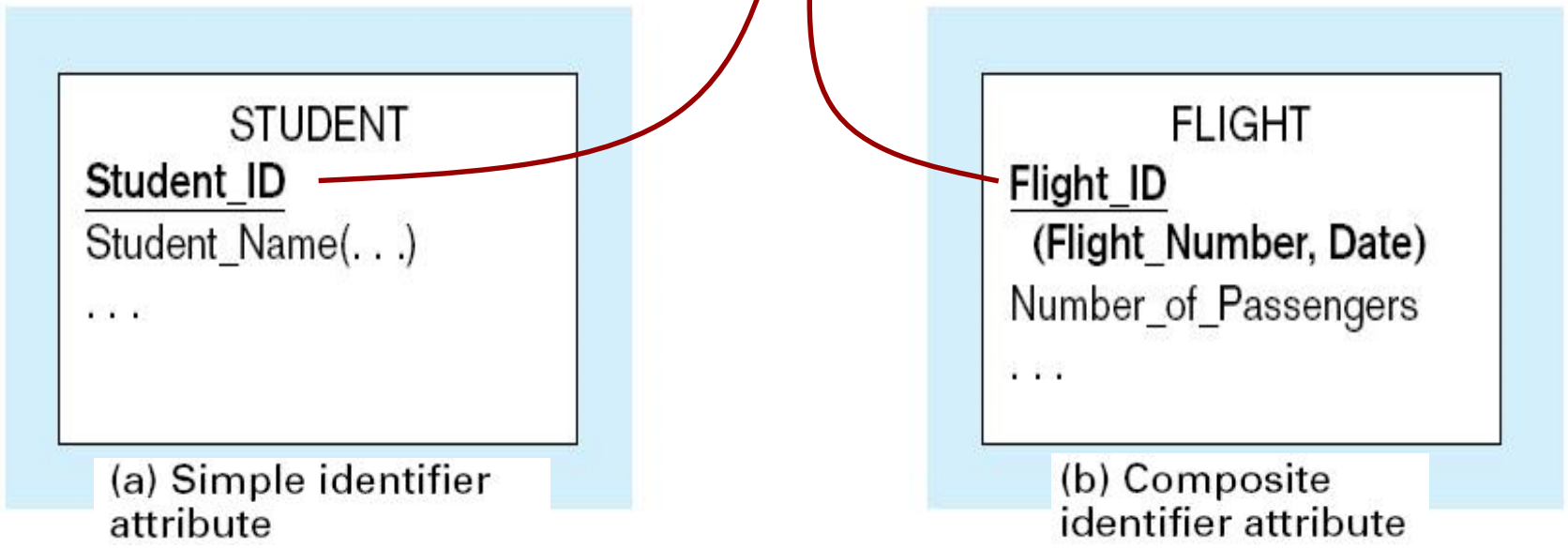
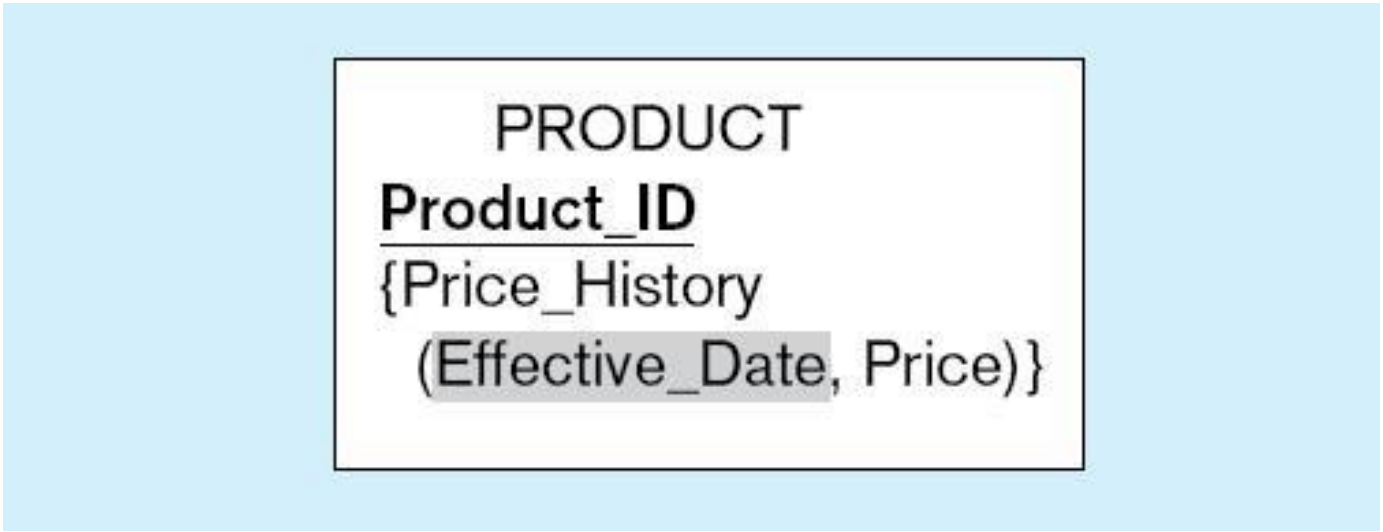


Figure 3-19 Simple example of time-stamping



This attribute is both multivalued *and* composite

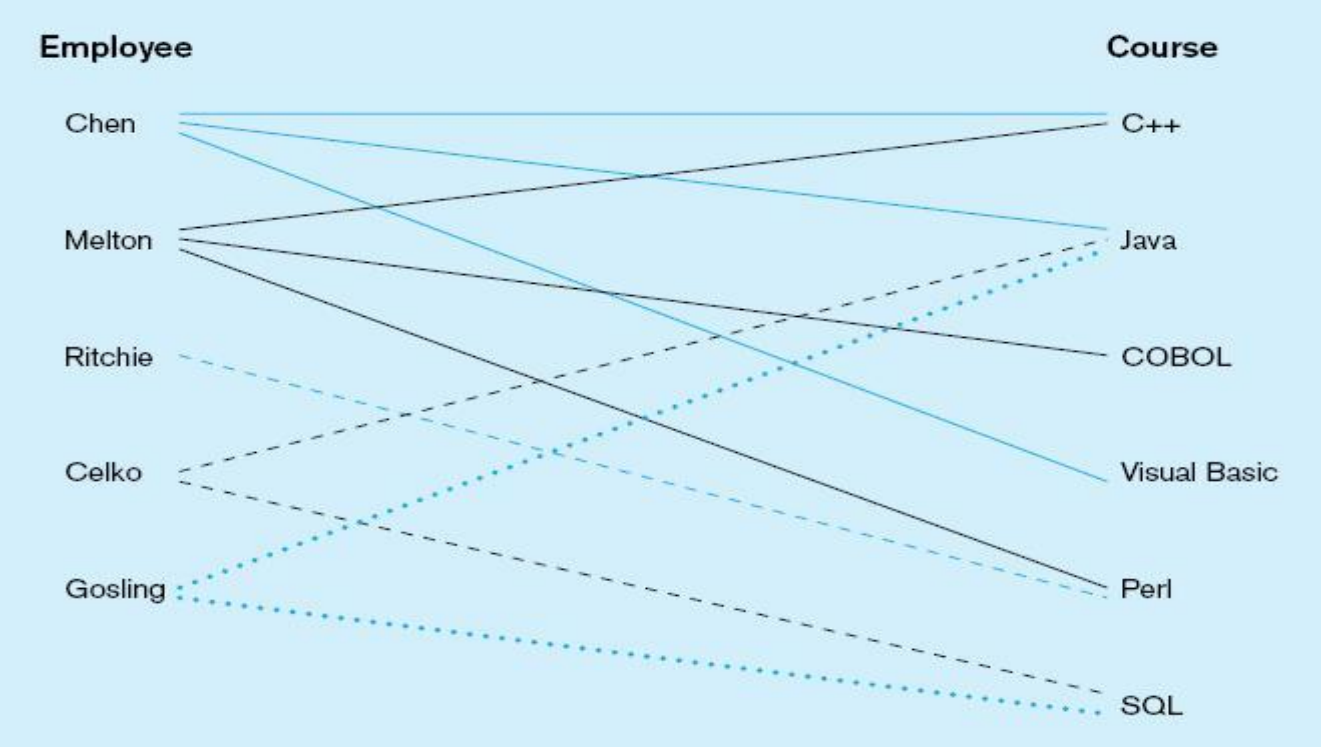
More on Relationships

- 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

Figure 3-10 Relationship types and instances

a) Relationship type

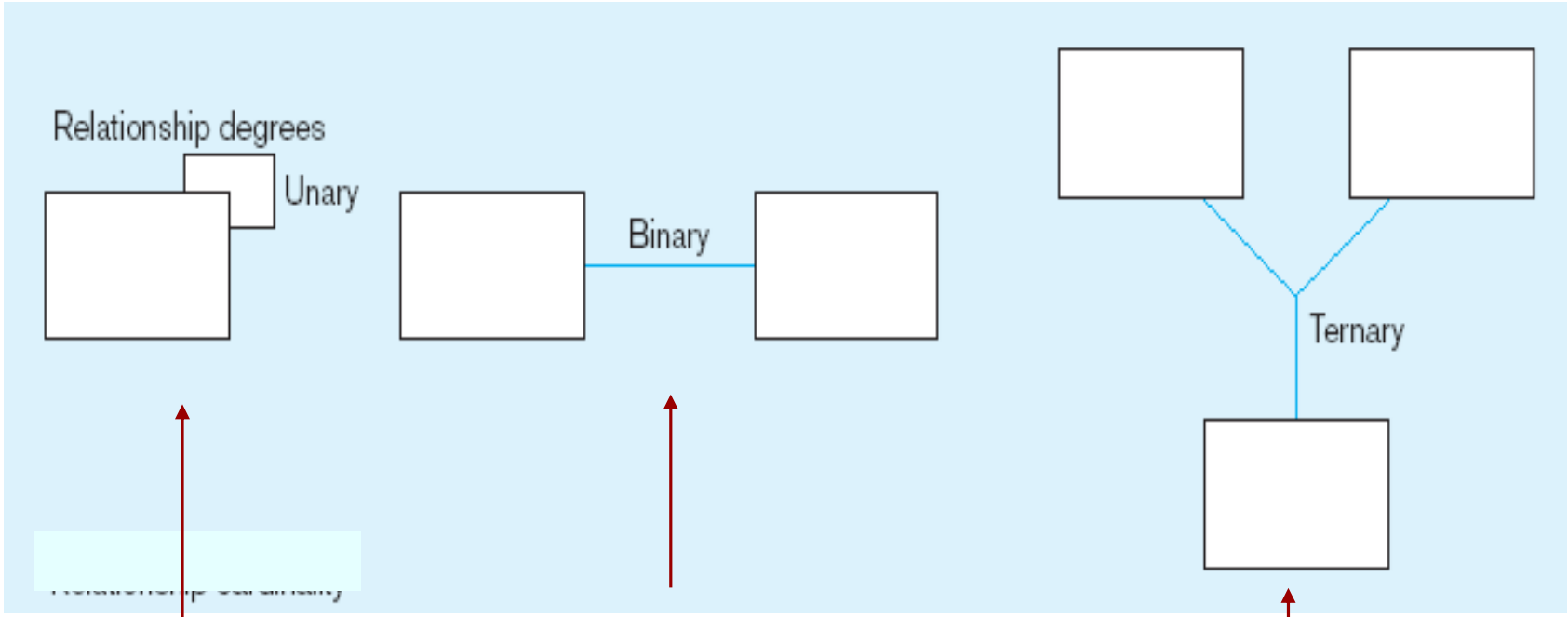
b) Relationship instances



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 – from Figure 3-2



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

Figure 3-12 Examples of relationships of different degrees

a) Unary relationships

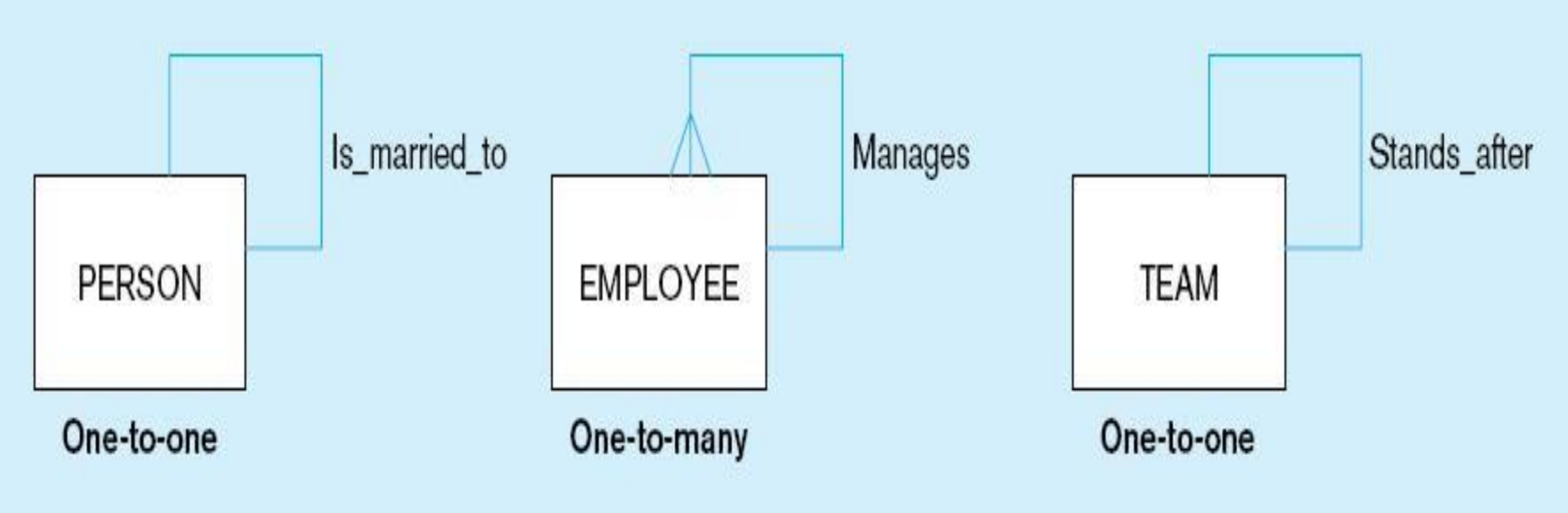


Figure 3-12 Examples of relationships of different degrees (cont.)

b) Binary relationships

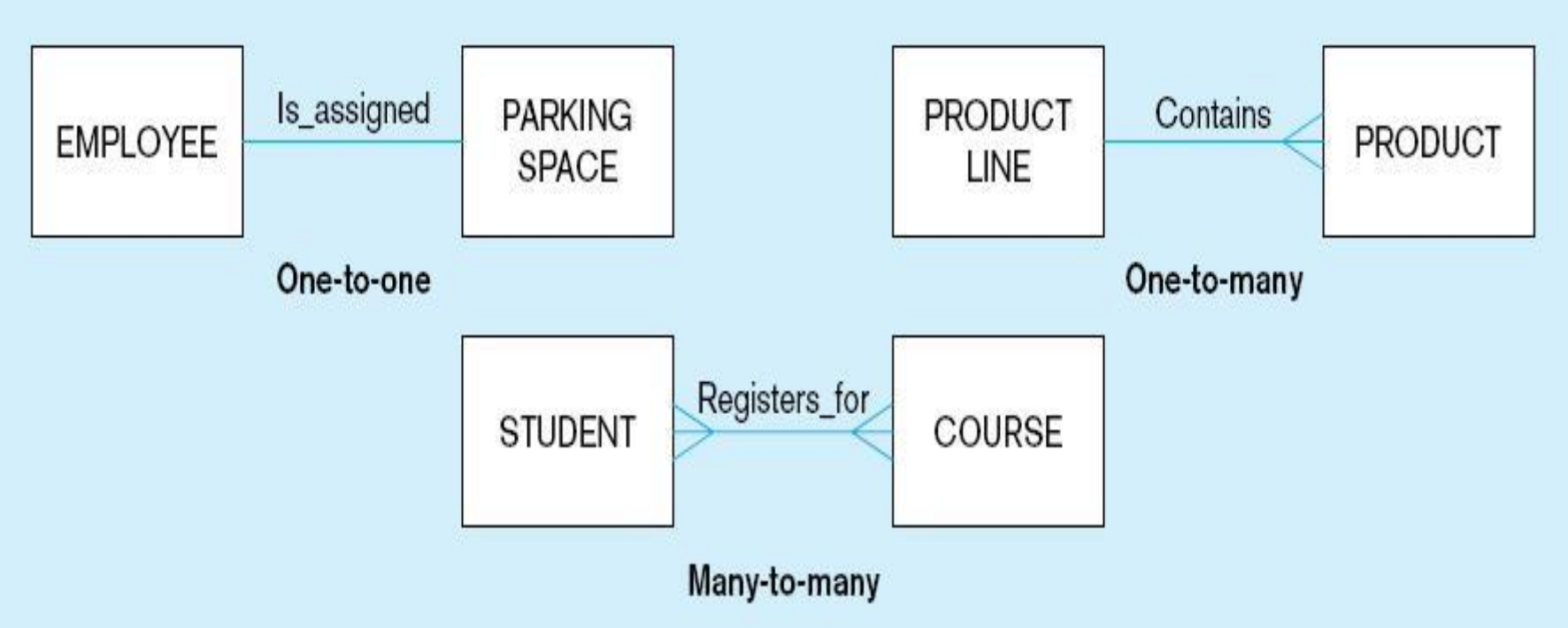
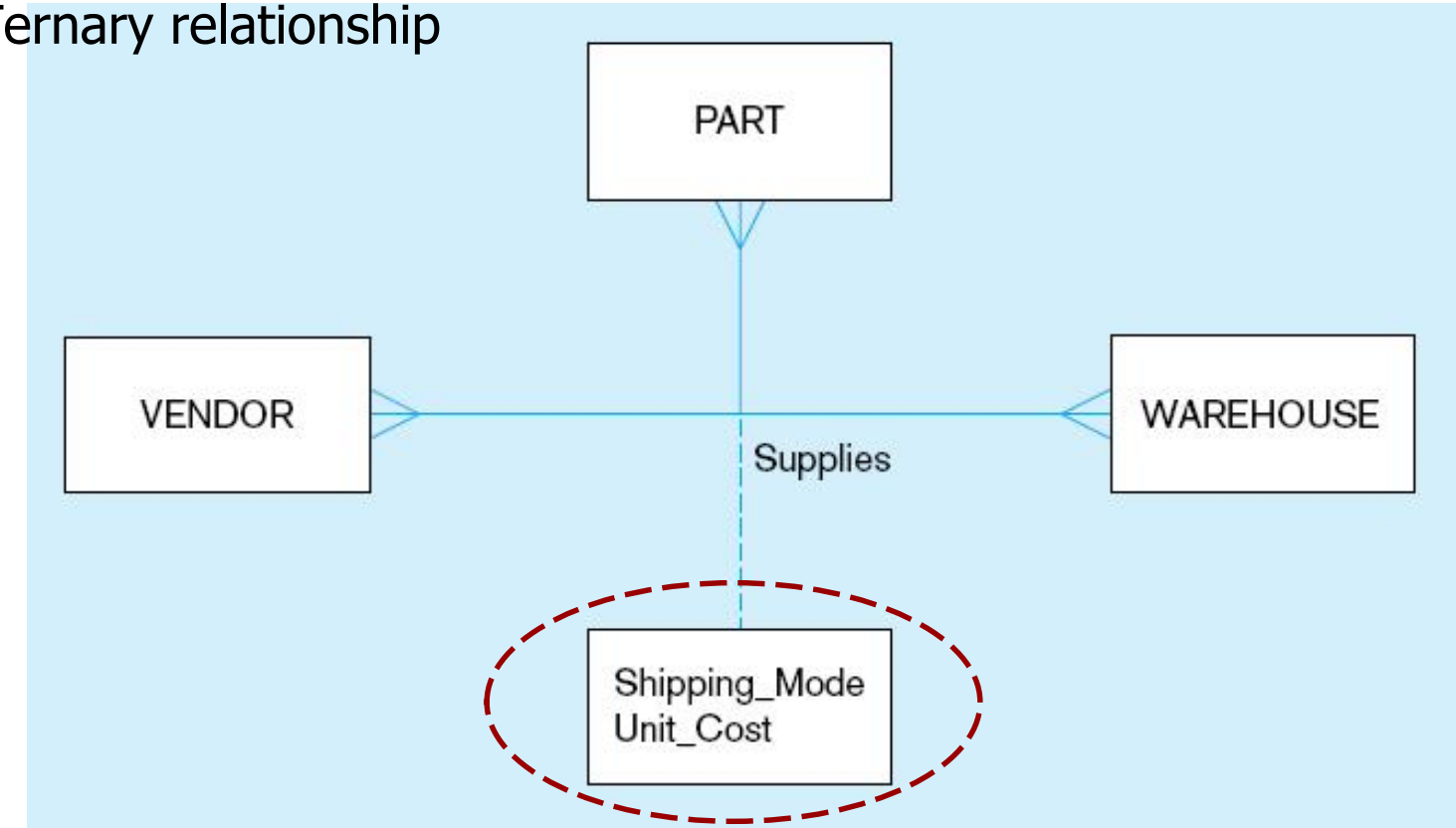


Figure 3-12 Examples of relationships of different degrees (cont.)

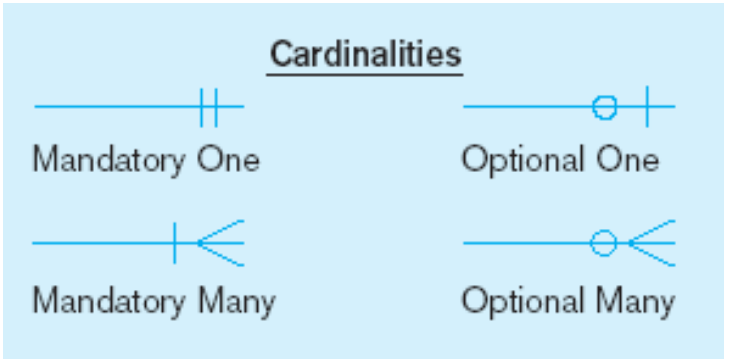
c) Ternary relationship

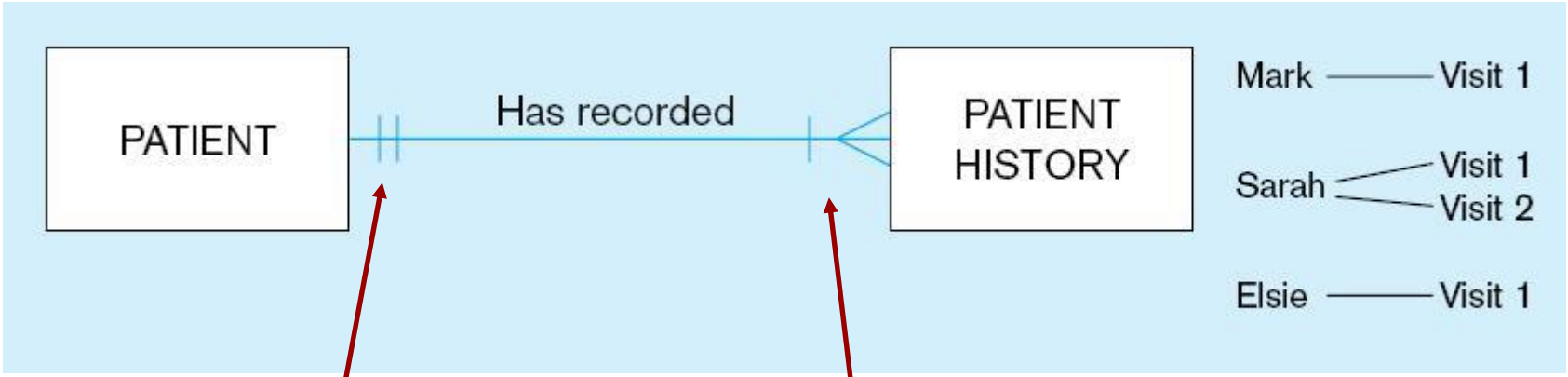


Note: a relationship can have attributes of its own

Figure 3-17 Examples of cardinality constraints

a) Mandatory cardinalities

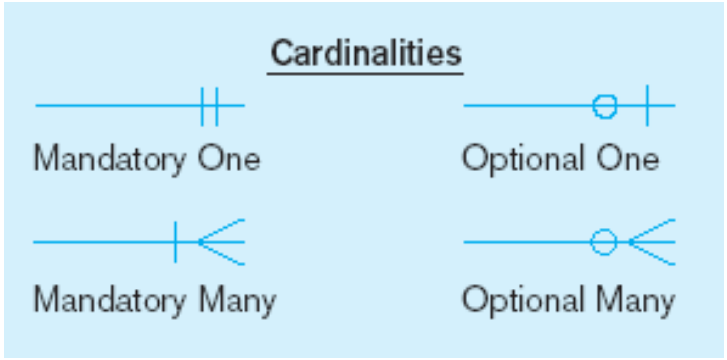




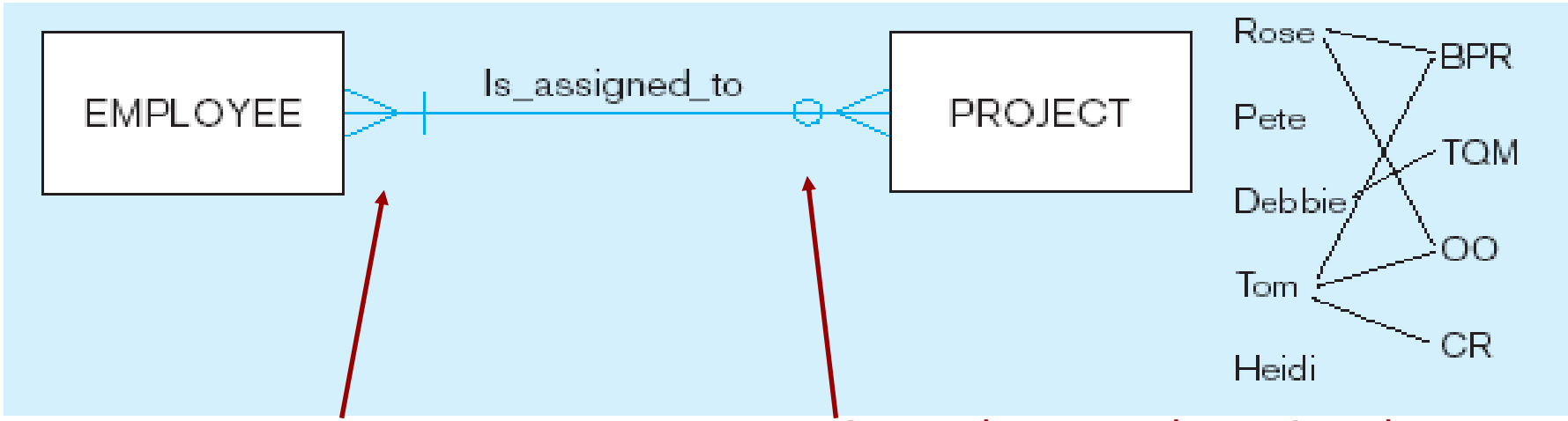
A patient history is recorded for one and only one patient

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

Figure 3-17 Examples of cardinality constraints (cont.)



b) One optional, one mandator

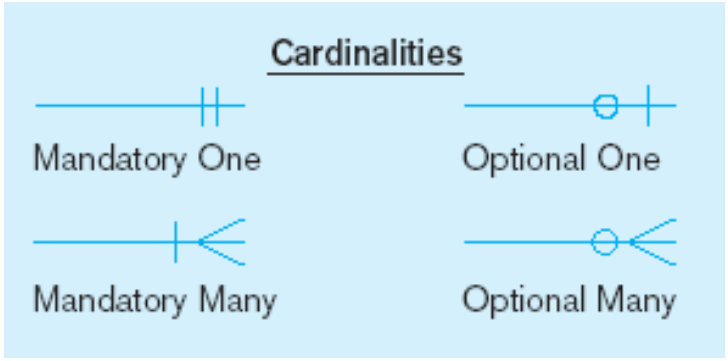


A project must be assigned to at least one employee, and may be assigned to many

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

Figure 3-17 Examples of cardinality constraints (cont.)

c) Optional cardinalities



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

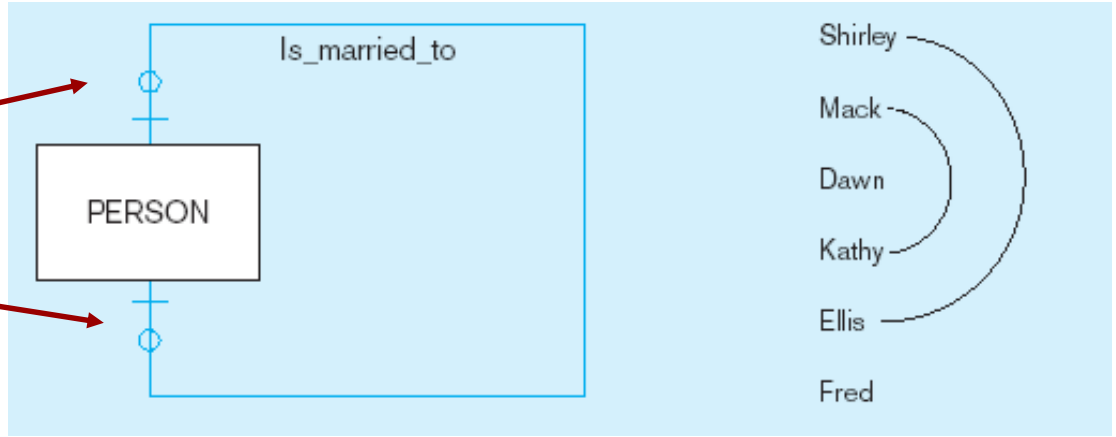
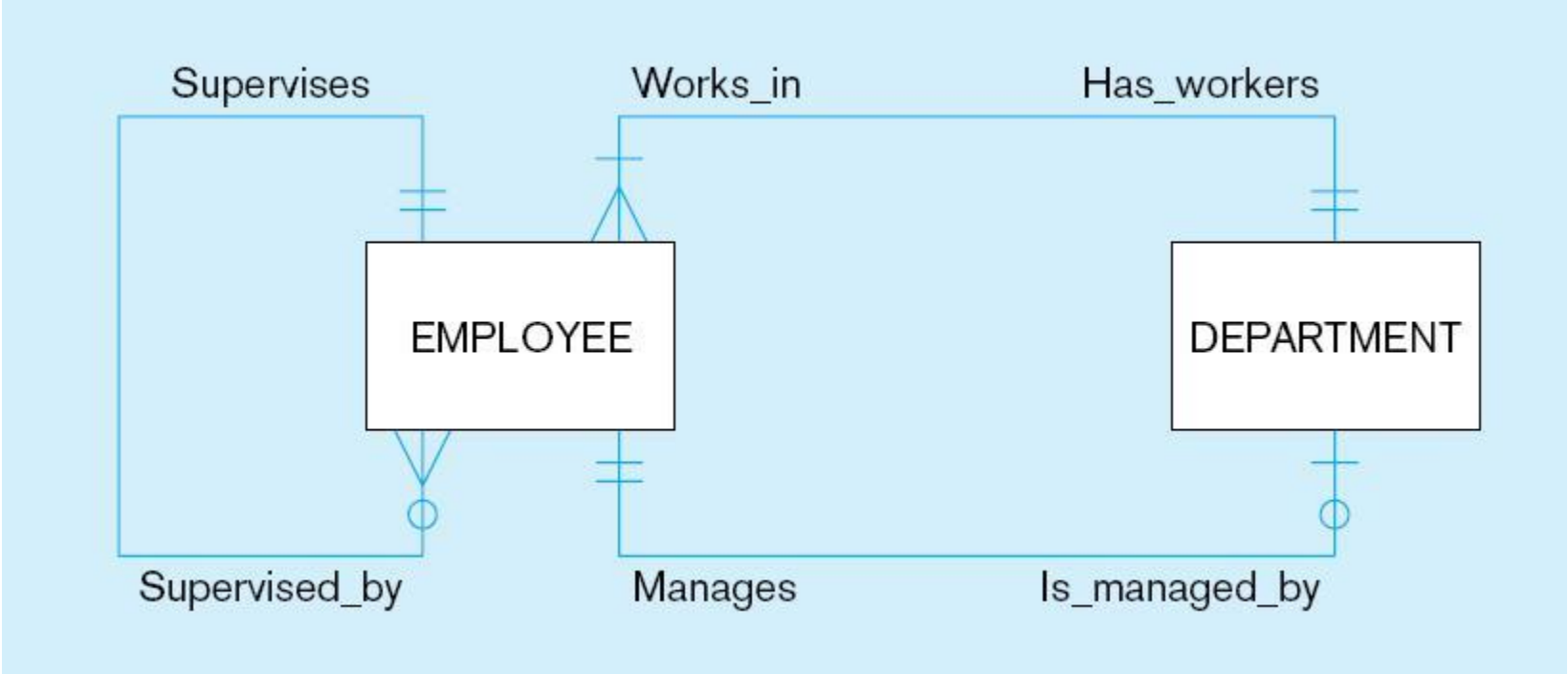


Figure 3-21 Examples of multiple relationships

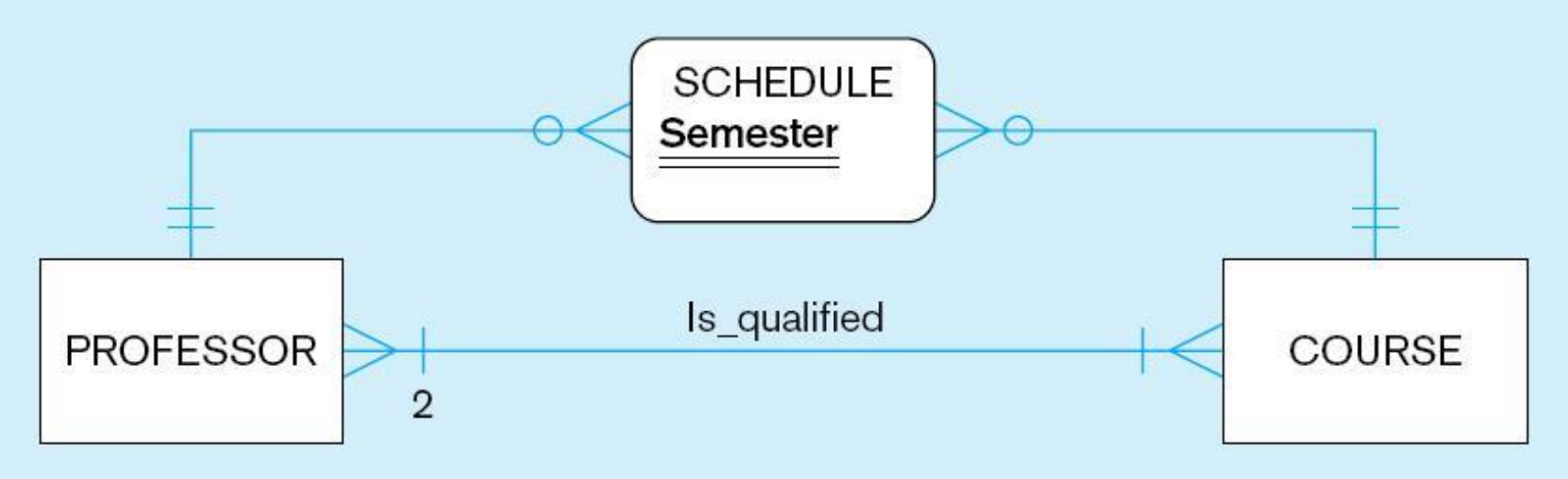
a) Employees and departments



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

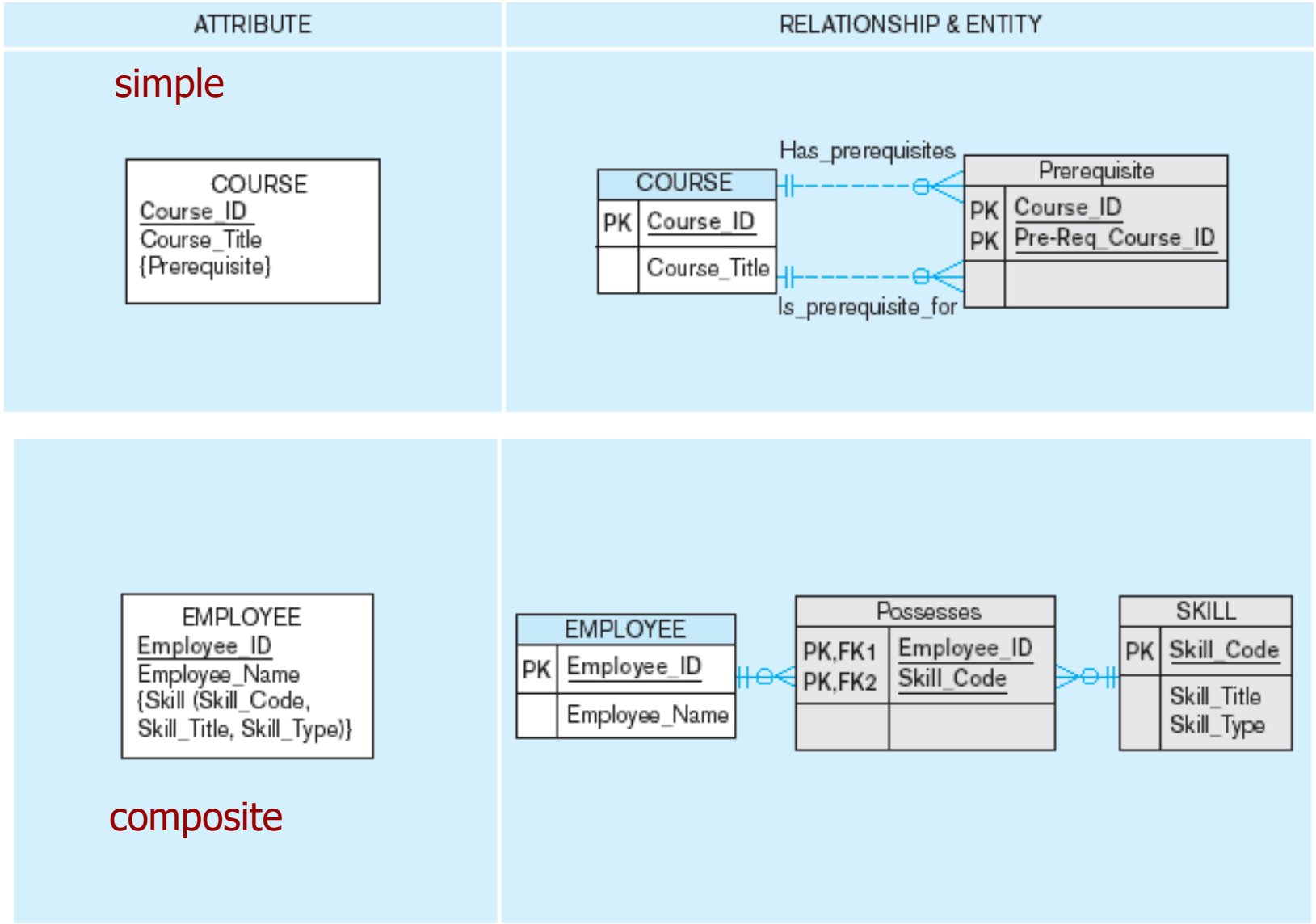
Figure 3-21 Examples of multiple relationships (cont.)

b) Professors and courses (fixed lower limit constraint)



Here, min cardinality constraint is 2

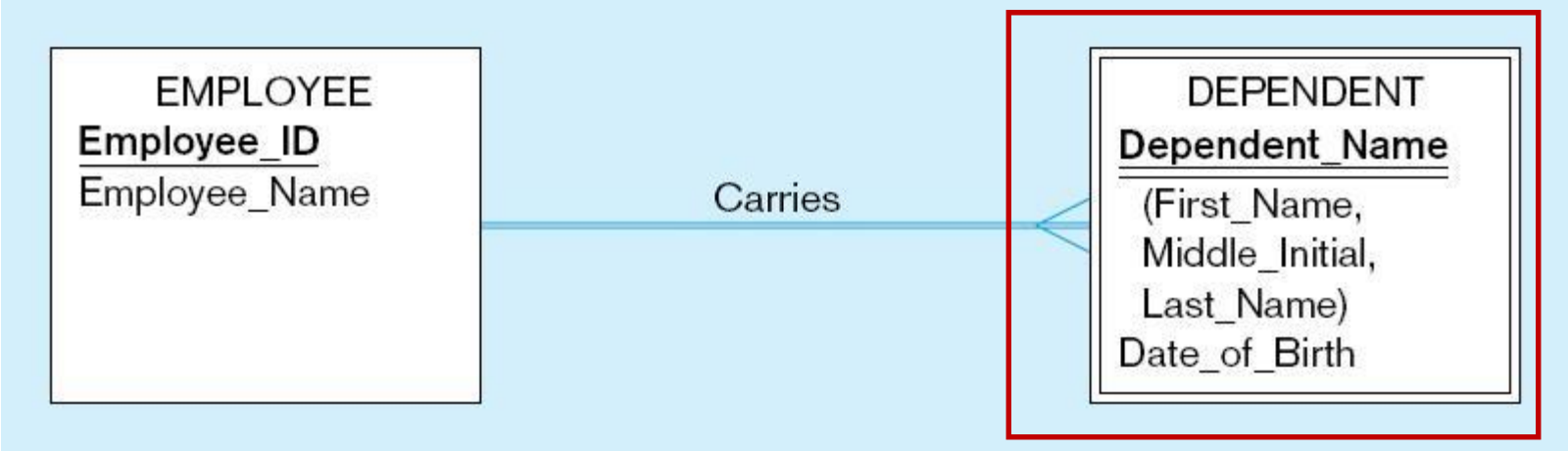
Figure 3-15a and 3-15b Multivalued attributes can be represented as relationships



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 (Figure 3-5)



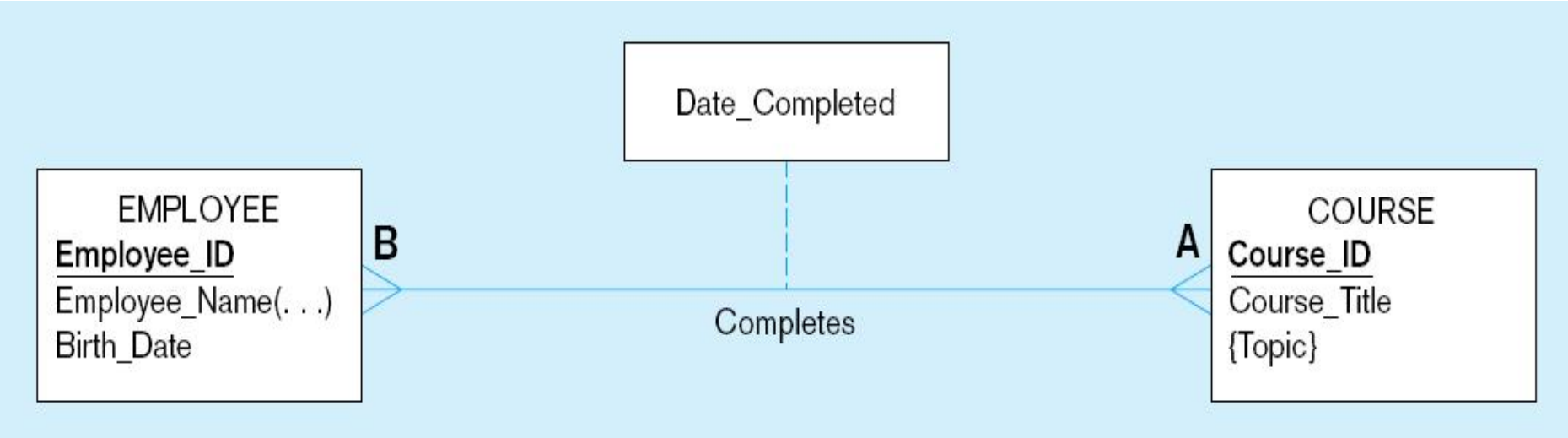
Strong entity

Weak entity

Associative Entities

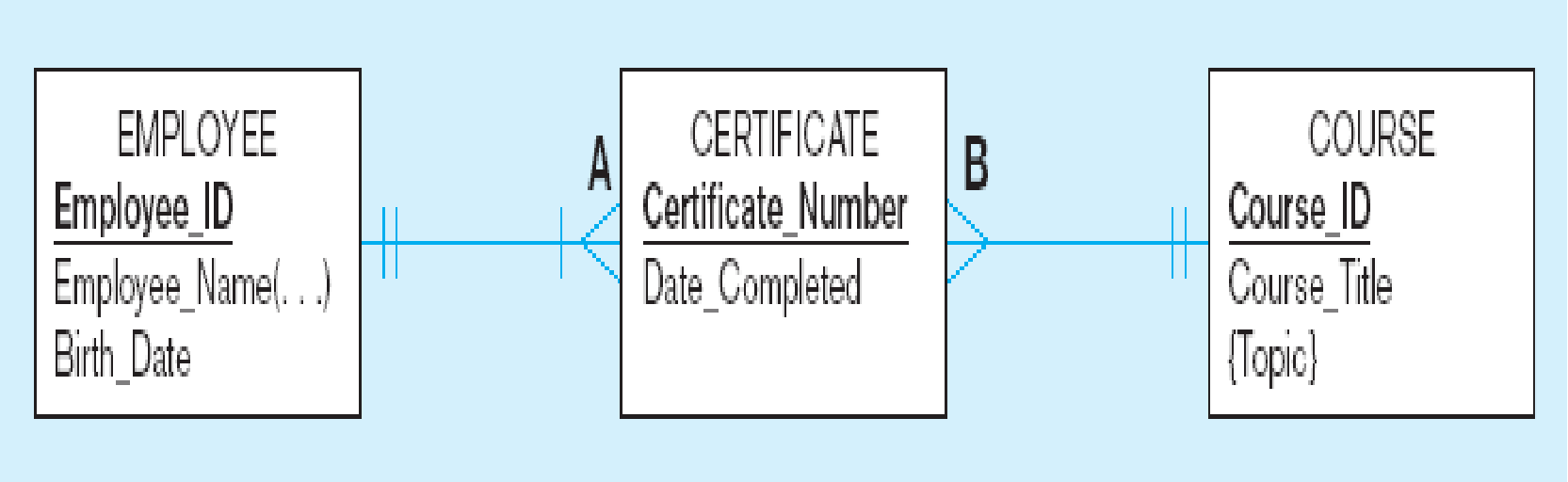
- An **entity**–has attributes
- A **relationship**–links entities together
- When should a *relationship with attributes* instead be an *associative entity*?
  - All relationships for the associative entity should be many
  - The associative entity could have meaning independent of the other entities
  - 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

Figure 3-11a 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*

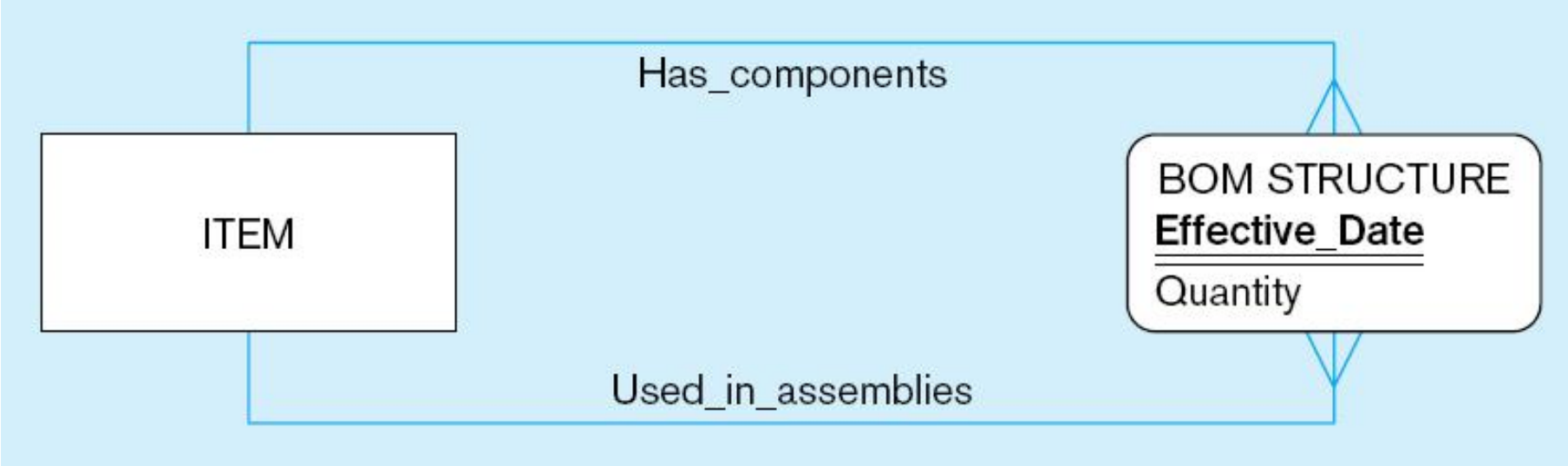
Figure 3-11b An associative entity (CERTIFICATE)



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

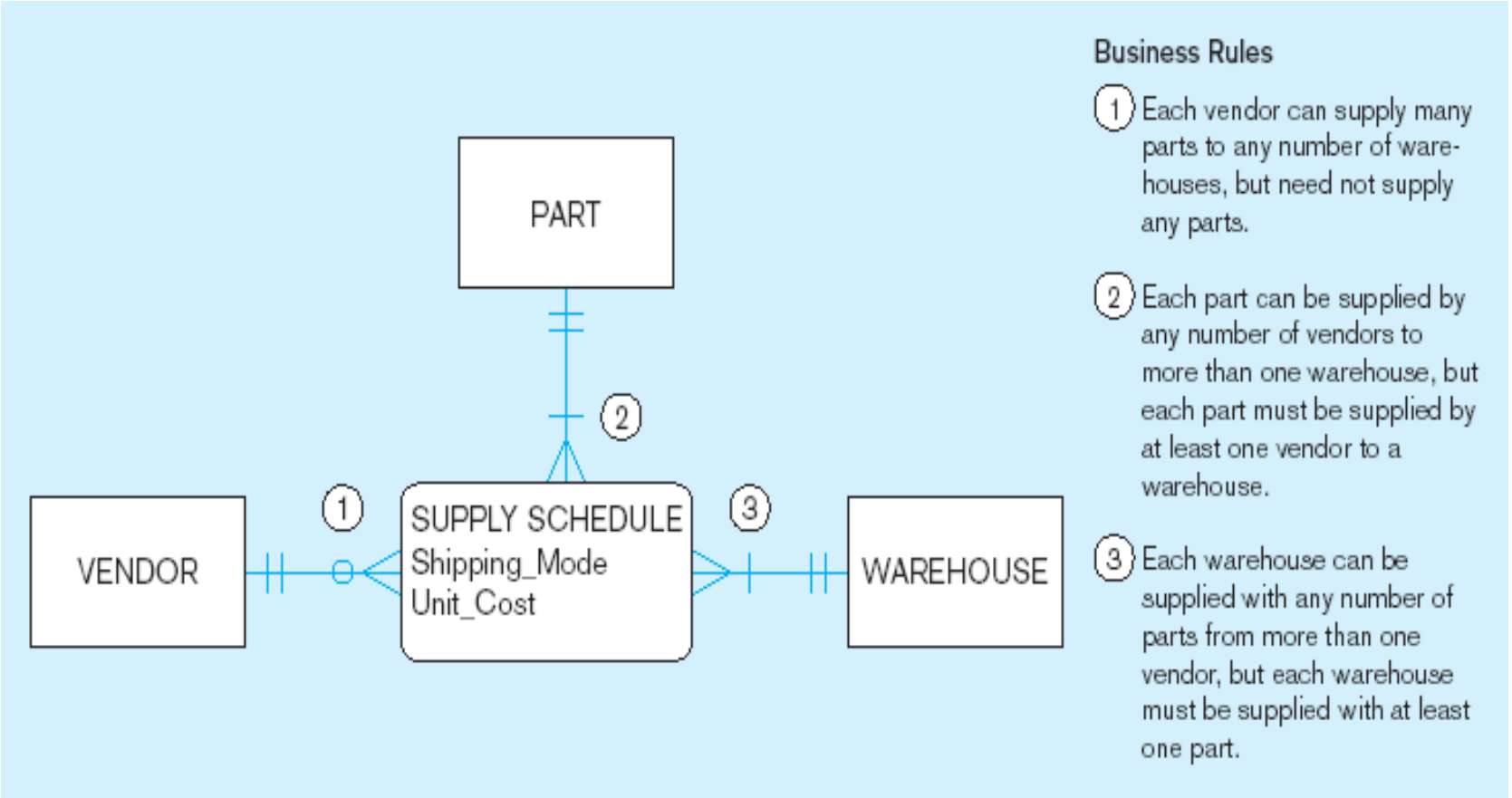
Note that the many-to-many cardinality between entities in Figure 3-11a has been replaced by two one-to-many relationships with the associative entity

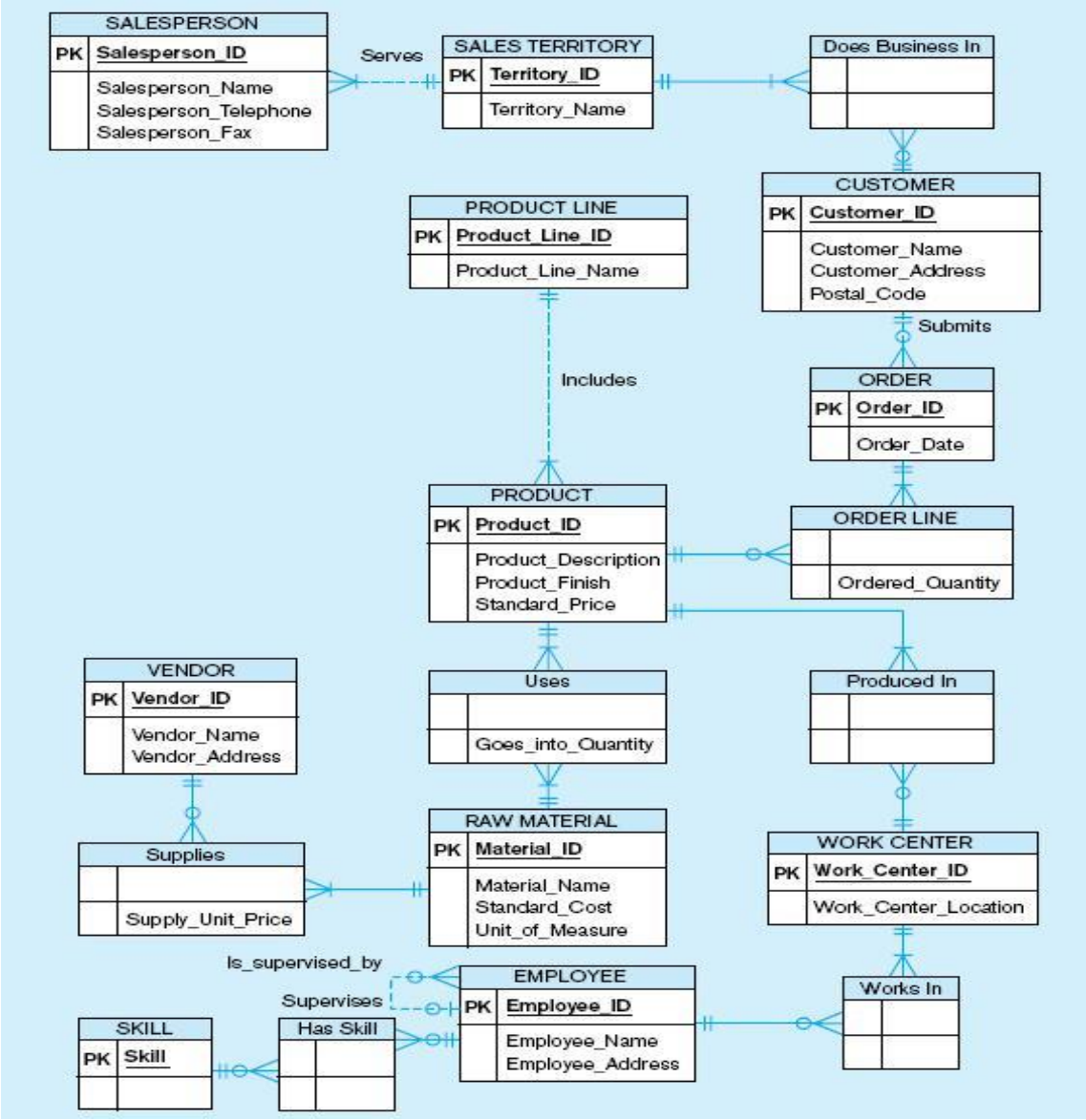
Figure 3-13c An associative entity – bill of materials structure



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

Figure 3-18 Ternary relationship as an associative entity





Microsoft Visio Notation for Pine Valley Furniture E-R diagram

(Figure 3-22)

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