Modern Systems Analysis and Design

Seventh Edition

Jeffrey A. Hoffer
Joey F. George
Joseph S. Valacich

Chapter 8

Structuring System Data Requirements

Learning Objectives

- ✓ Concisely define each of the following key data modeling terms: entity type, attribute, multivalued attribute, relationship, degree, cardinality, business rule, associative entity, trigger, supertype, subtype.
- ✓ Draw an entity-relationship (E-R) diagram to represent common business situations.
- ✓ Explain the role of conceptual data modeling in the overall analysis and design of an information system.

Learning Objectives (Cont.)

- ✓ Explain the role of prepackaged database models (patterns) in data modeling.
- ✓ Distinguish between unary, binary, and ternary relationships and give an example of each.
- ✓ Define four basic types of business rules in a conceptual data model.
- ✓ Relate data modeling to process and logic modeling as different views of describing an information system.

Conceptual Data Modeling

- **Conceptual data modeling**: a detailed model that captures the overall structure of data in an organization
  - Independent of any database management system (DBMS) or other implementation considerations

Conceptual Data Modeling (Cont.)

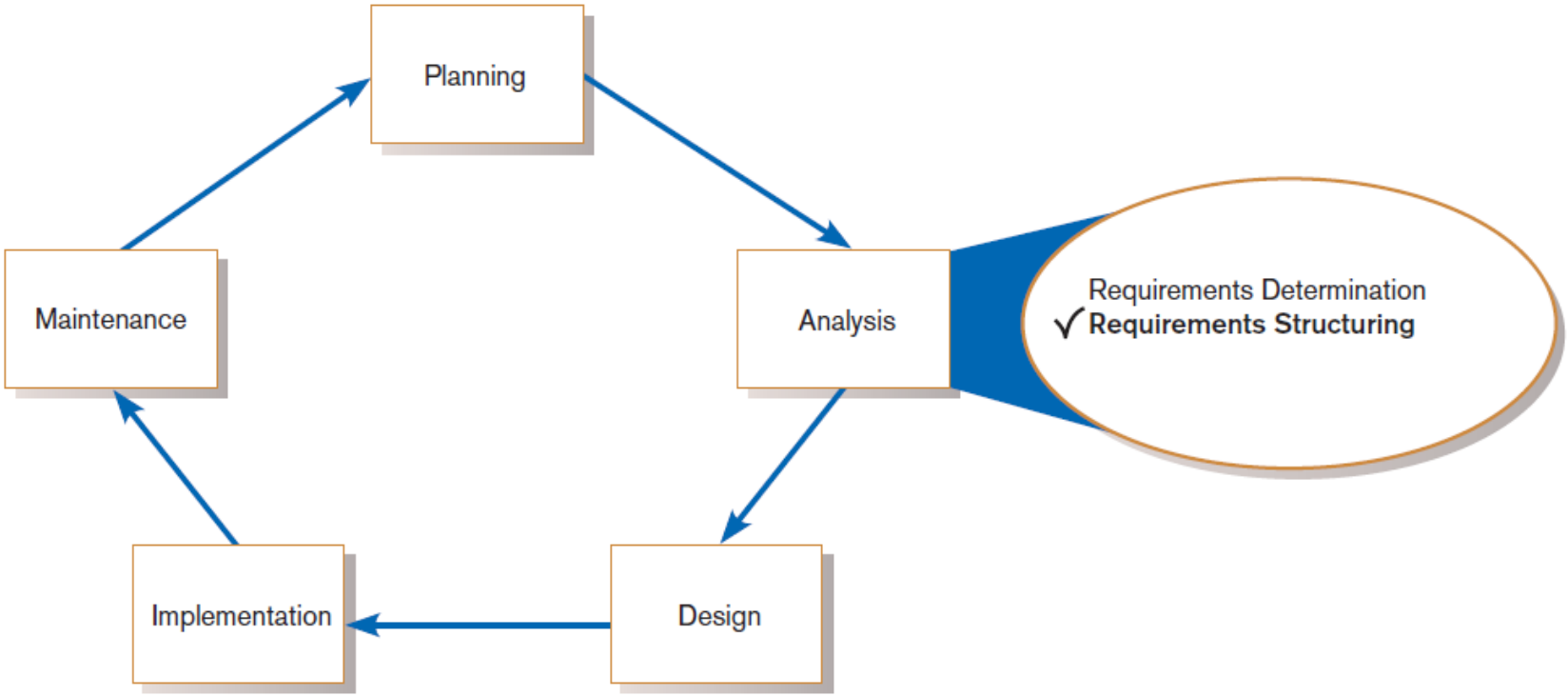


FIGURE 8-1
Systems development life cycle with analysis phase highlighted

The Conceptual Data Modeling Process

- Develop a data model for the current system.
- Develop a new conceptual data model that includes all requirements of the new system.
- In the design stage, the conceptual data model is translated into a physical design.
- Project repository links all design and data modeling steps performed during SDLC.

Conceptual Data Modeling (Cont.)

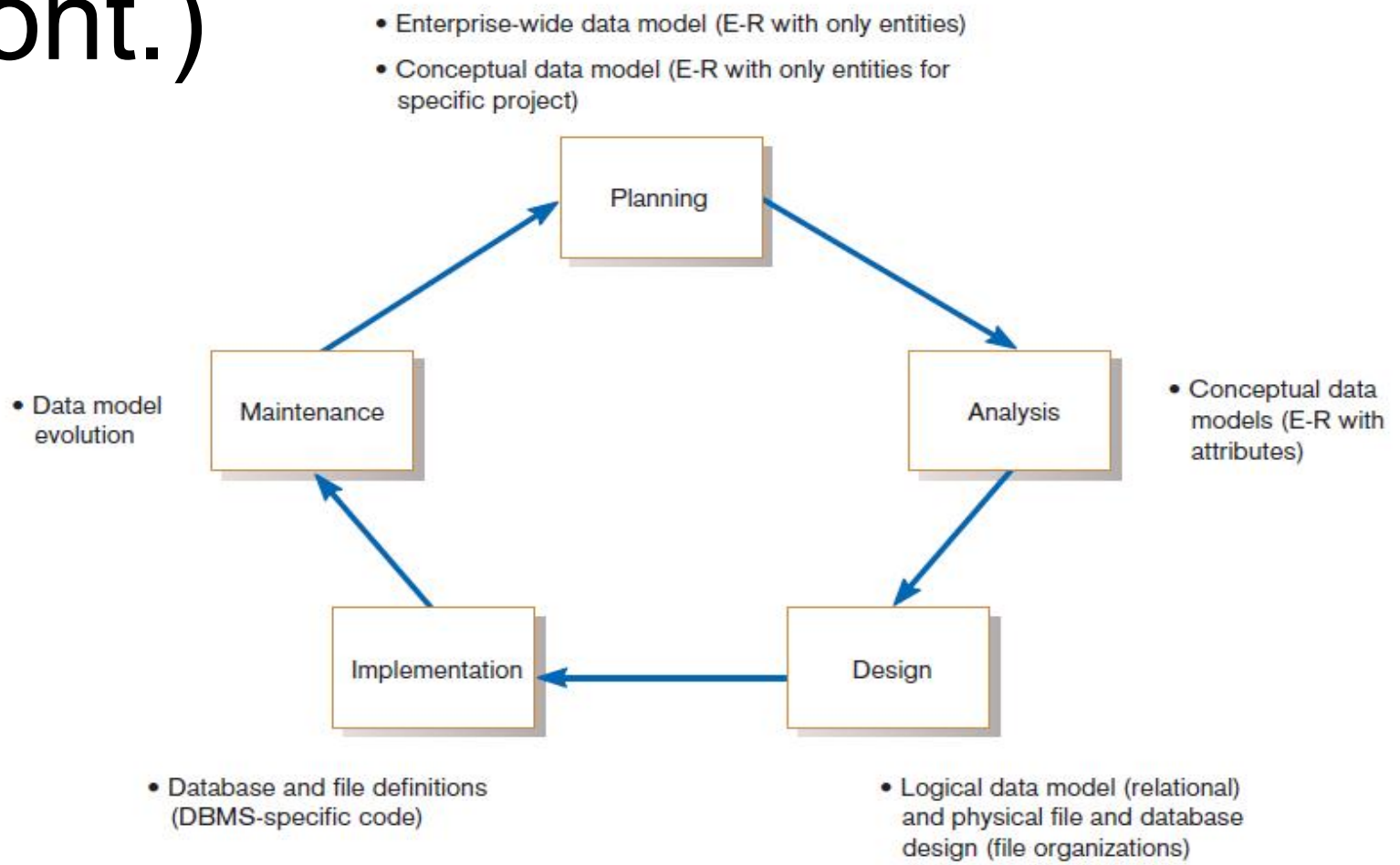


FIGURE 8-2
Relationship between data modeling and the SDLC

Deliverables and Outcome

- Entity-relationship (E-R) diagram or UML class diagram
  - Entities (or classes) – categories of data, represented as rectangles
  - Relationships (or associations) – lines between the entities
- Set of entries about data objects to be stored in repository project dictionary, or data modeling software
  - Repository links data, process, and logic models of an information system.
  - Data elements included in the data flow diagram (DFD) must appear in the data model and vice versa.
  - Each data store in a process model must relate to business objects represented in the data model.

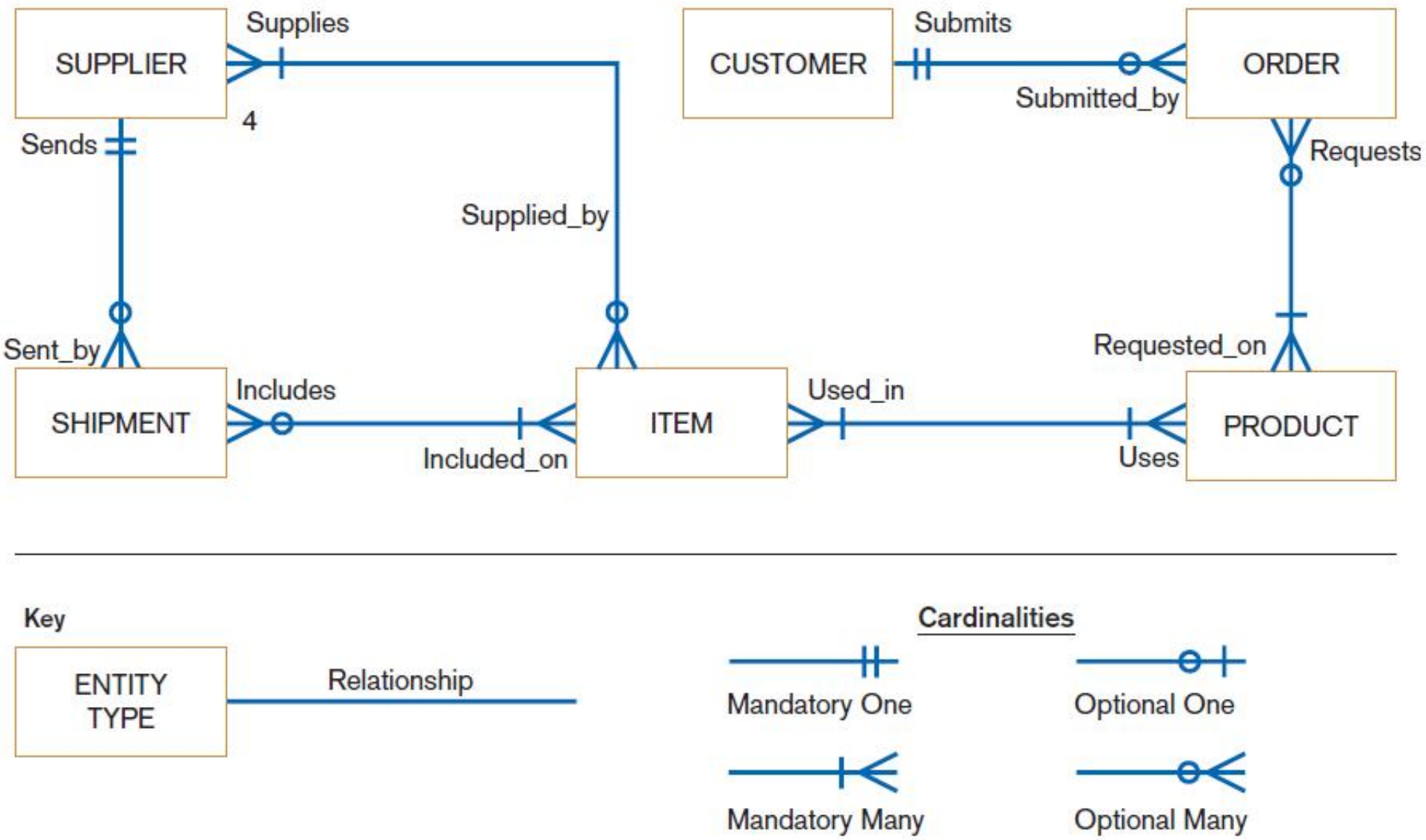


FIGURE 8-3
Sample conceptual data model

Gathering Information for Conceptual Data Modeling

- Two perspectives on data modeling:
  - *Top-down approach* for a data model is derived from an intimate understanding of the business.
  - *Bottom-up approach* for a data model is derived by reviewing specifications and business documents.

Gathering Information for Conceptual Data Modeling (Cont.)

- Requirements Determination Questions for Data Modeling:
  - What are subjects/objects of the business?
    - Data entities and descriptions
  - What unique characteristics distinguish between subjects/objects of the same type?
    - Primary keys

Gathering Information for Conceptual Data Modeling (Cont.)

- What characteristics describe each subject/object?
  - Attributes and secondary keys
- How do you use the data?
  - Security controls and user access privileges
- Over what period of time are you interested in the data?
  - Cardinality and time dimensions

Gathering Information for Conceptual Data Modeling (Cont.)

- Are all instances of each object the same?
  - Supertypes, subtypes, and aggregations
- What events occur that imply associations between objects?
  - Relationships and cardinalities
- Are there special circumstances that affect the way events are handled?
  - Integrity rules, cardinalities, time dimensions

Introduction to Entity-Relationship (E-R) Modeling

- **Entity-Relationship data model** (**E-R model**): a detailed, logical representation of the entities, associations and data elements for an organization or business area
- **Entity-relationship diagram** (**E-R diagram**): a graphical representation of an E-R model

Introduction to Entity-Relationship (E-R) Modeling

- The E-R model is expressed in terms of:
  - Data entities in the business environment.
  - Relationships or associations among those entities.
  - Attributes or properties of both the entities and their relationships.

Introduction to E-R Modeling (Cont.)

- **Entity**: a person, place, object, event or concept in the user environment about which data is to be maintained
- **Entity type**: collection of entities that share common properties or characteristics
- **Entity instance**: single occurrence of an entity type

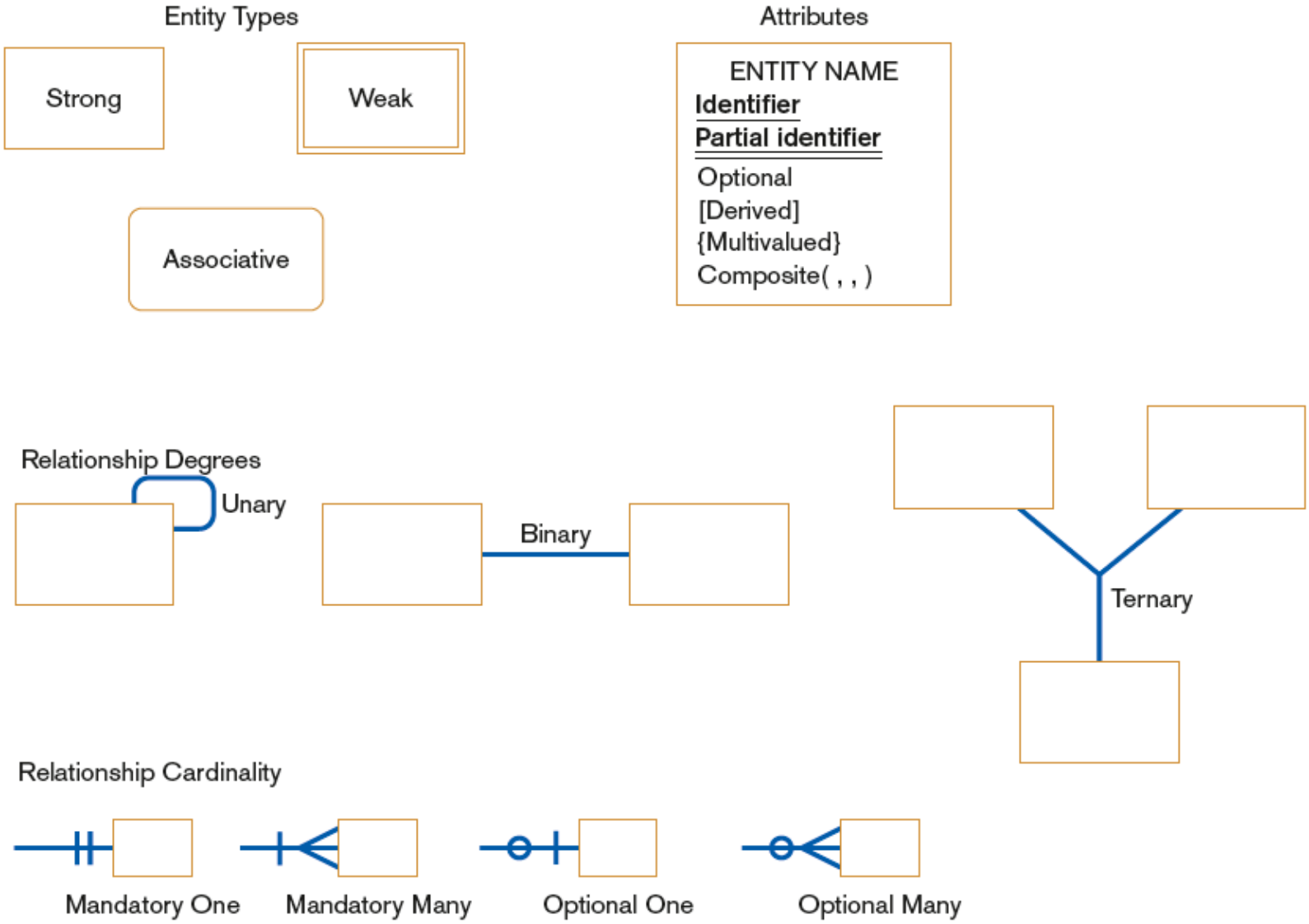


FIGURE 8-5 Basic E-R notation

Naming and Defining Entity Types

- An entity type name should be:
  - *A singular noun.*
  - *Descriptive and specific to the organization.*
  - *Concise.*
- *Event entity type* should be named for the *result of the event*, not the activity or process of the event.

Naming and Defining Entity Types (Cont.)

- An entity type definition:
  - Includes a statement of *what the unique characteristic(s) is (are) for each instance*.
  - Makes clear *what entity instances are included and not included* in the entity type.
  - Often includes a description of *when an instance of the entity type is created or deleted*.

Naming and Defining Entity Types (Cont.)

- For some entity types the definition must specify:
  - *When an instance might change into an instance of another entity type*.
  - *What history is to be kept about entity instances*.

Attributes

- **Attribute**: a named property or characteristic of an entity that is of interest to the organization
  - Naming an attribute: i.e. Vehicle_ID
  - Place its name inside the rectangle for the associated entity in the E-R diagram.

Naming and Defining Attributes

- An attribute name is a *noun* and should be *unique*.
- To make an attribute name unique and for clarity, *each attribute name should follow a standard format*.
- *Similar attributes of different entity types should use similar but distinguishing names*.

Naming and Defining Attributes (Cont.)

- An attribute definition:
  - *States what the attribute is and possibly why it is important*.
  - Should make it clear *what is included and what is not included*.
  - Contains any *aliases* or alternative names.
  - States *the source of values for the attribute*.

Naming and Defining Attributes (Cont.)

- An attribute definition should indicate:
  - *If a value for the attribute is required or optional*.
  - *If a value for the attribute may change.*
  - Any *relationships that attribute has with other attributes.*

Candidate Keys and Identifiers.

- **Candidate key**: an attribute (or combination of attributes) that uniquely identifies each instance of an entity type
- **Identifier**: a candidate key that has been selected as the unique, identifying characteristic for an entity type

Candidate Keys and Identifiers (Cont.)

- Selection rules for an identifier
  - Choose a candidate key that will not change its value.
  - Choose a candidate key that will never be null.
  - Avoid using intelligent keys.
  - Consider substituting single value surrogate keys for large composite keys.

Other Attribute Types

- **Multivalued attribute**: an attribute that may take on more than one value for each entity instance
- **Repeating group**: a set of two or more multivalued attributes that are logically related

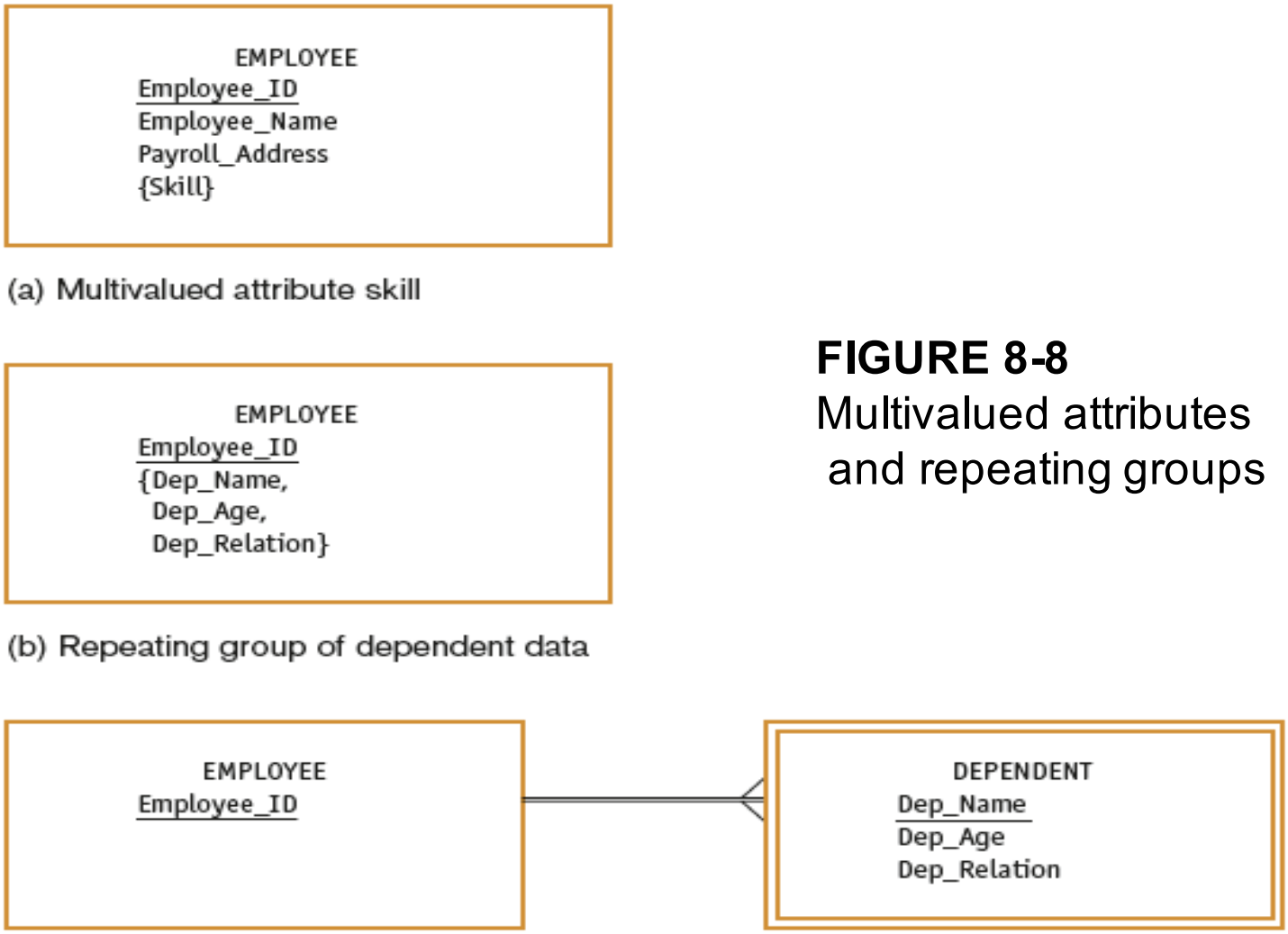


FIGURE 8-8
Multivalued attributes and repeating groups

Other Attribute Types

- **Required attribute**: an attribute that must have a value for every entity instance
- **Optional attribute**: an attribute that may not have a value for every entity instance
- **Composite attribute**: an attribute that has meaningful component parts
- **Derived attribute**: an attribute whose value can be computed from related attribute values

Relationships

- **Relationship**: an association between the instances of one or more entity types that is of interest to the organization
- **Degree**: the number of entity types that participate in a relationship

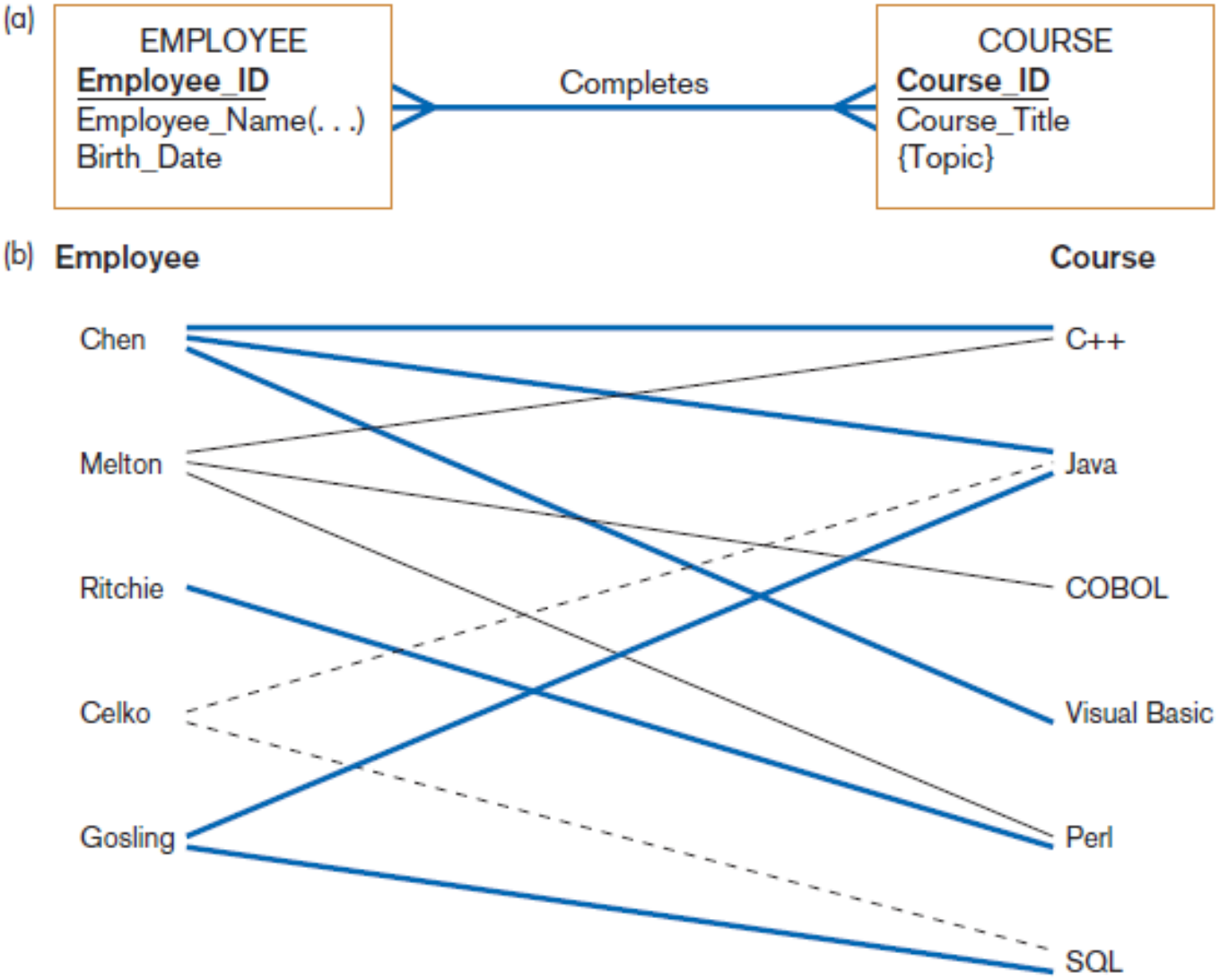


Figure 8-10
Relationship type and instances
(a) Relationship type (Completes)
(b) Relationship instances

Conceptual Data Modeling and the E-R Model

- **Unary relationship**: a relationship between the instances of one entity type
  - Also called a *recursive relationship*
- **Binary relationship**: a relationship between instances of two entity types
  - Most common type of relationship encountered in data modeling
- **Ternary relationship**: a simultaneous relationship among instances of three entity types

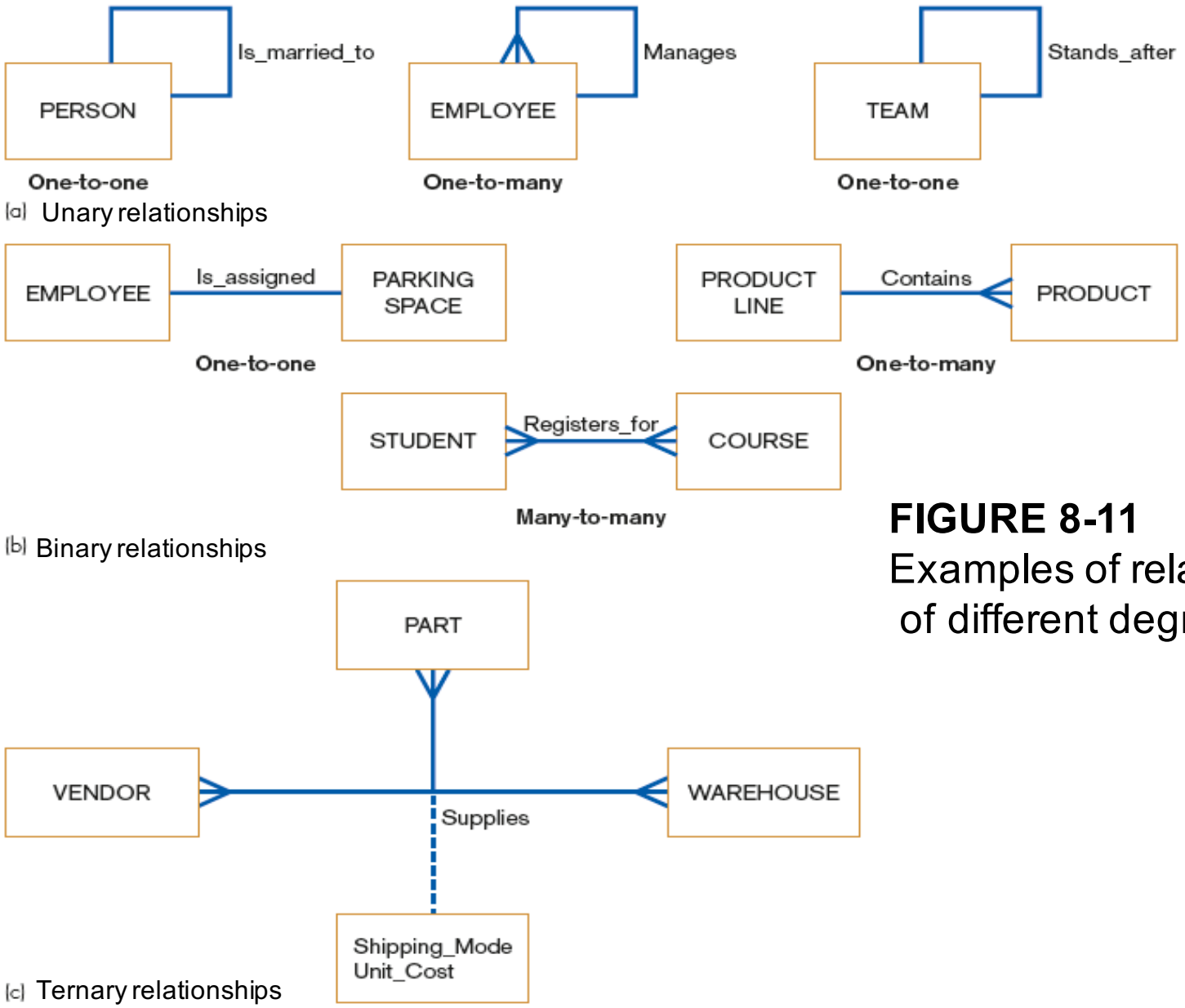


FIGURE 8-11
Examples of relationships of different degrees

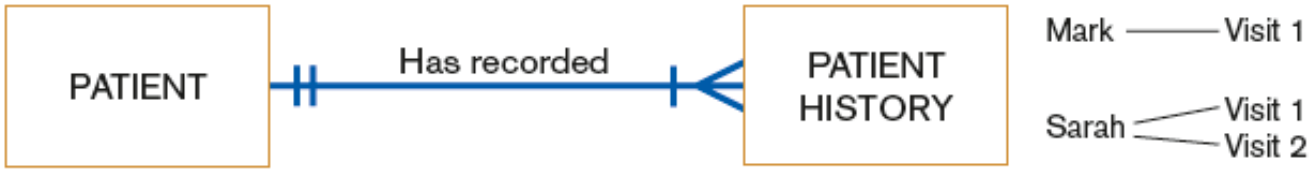
Cardinalities in Relationships

- **Cardinality**: the number of instances of entity B that can (or must) be associated with each instance of entity A
- Minimum Cardinality
  - The minimum number of instances of entity B that may be associated with each instance of entity A
- Maximum Cardinality
  - The maximum number of instances of entity B that may be associated with each instance of entity A

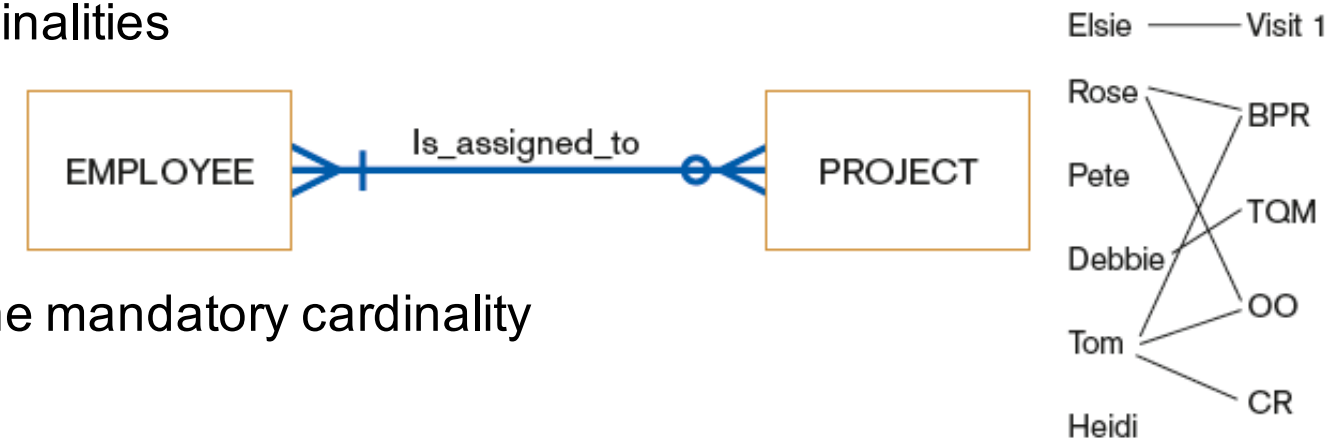
Cardinalities in Relationships (Cont.)

- Mandatory vs. Optional Cardinalities
  - Specifies whether an instance must exist or can be absent in the relationship

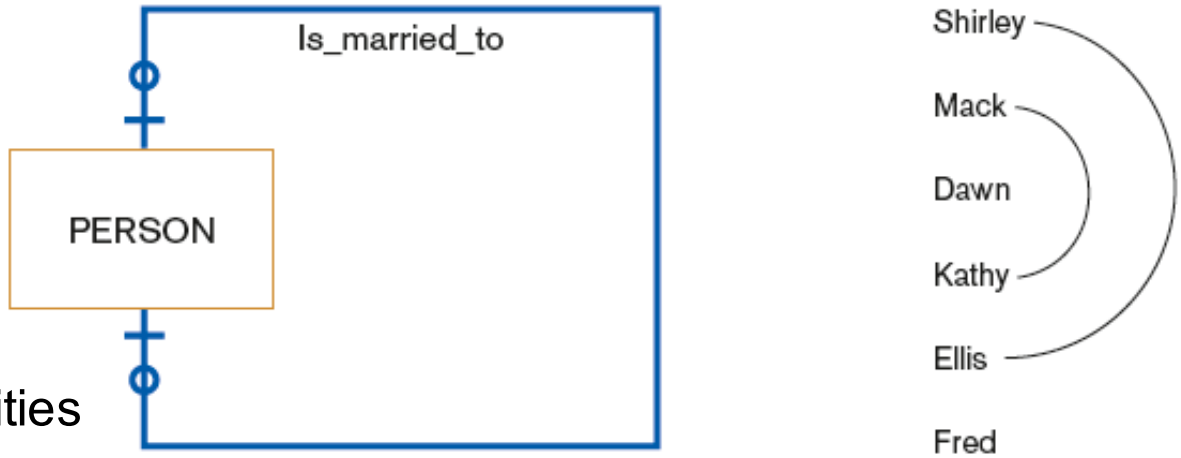
FIGURE 8-14 Examples of cardinality constraints



(a) Mandatory cardinalities



(b) One optional, one mandatory cardinality



(c) Optional cardinalities

Naming and Defining Relationships

- A relationship name is a verb phrase; avoid vague names.
- A relationship definition:
  - Explains what action is to be taken and possibly why it is important.
  - Gives examples to clarify the action.

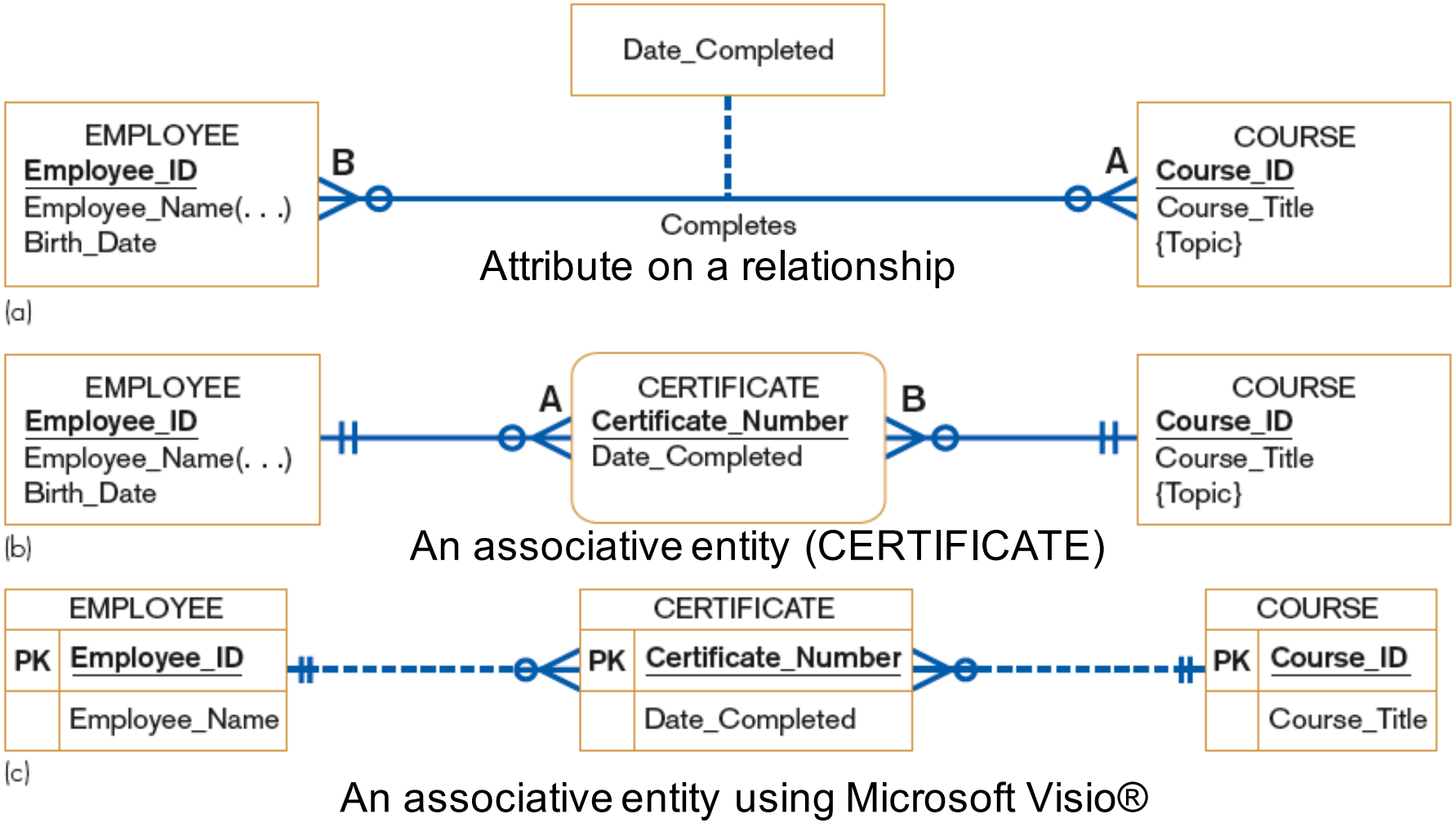
Naming and Defining Relationships (Cont.)

- A relationship definition should:
  - Explain any optional participation.
  - Explain the reason for any explicit maximum cardinality other than many.
  - Explain any restrictions on participation in the relationship.
  - Explain the extent of history that is kept in the relationship.
  - Explain whether an entity instance involved in a relationship instance can transfer participation to another relationship instance.

Associative Entities

- **Associative Entity**: an entity type that associates the instances of one or more entity types and contains attributes that are peculiar to the relationship between those entity instances
  - Sometimes called a gerund
- The data modeler chooses to model the relationship as an entity type.

FIGURE 8-15 An associative entity



Summary of Conceptual Data Modeling with E-R Diagrams

- The purpose of E-R diagramming is to capture the richest possible understanding of the meaning of the data necessary for an information system or organization.

Representing Supertypes and Subtypes

- **Subtype**: a subgrouping of the entities in an entity type
  - Is meaningful to the organization
  - Shares common attributes or relationships distinct from other subgroupings
- **Supertype**: a generic entity type that has a relationship with one or more subtypes

Representing Supertypes and Subtypes (Cont.)

- Business Rules for Supertype/subtype Relationships:
  - **Total specialization** specifies that each entity instance of the supertype must be a member of some subtype in the relationship.
  - **Partial specialization** specifies that an entity instance of the supertype does not have to belong to any subtype, and may or may not be an instance of one of the subtypes.

Representing Supertypes and Subtypes (Cont.)

- **Disjoint rule** specifies that if an entity instance of the supertype is a member of one subtype, it cannot simultaneously be a member of any other subtype.
- **Overlap rule** specifies that an entity instance can simultaneously be a member of two (or more) subtypes.

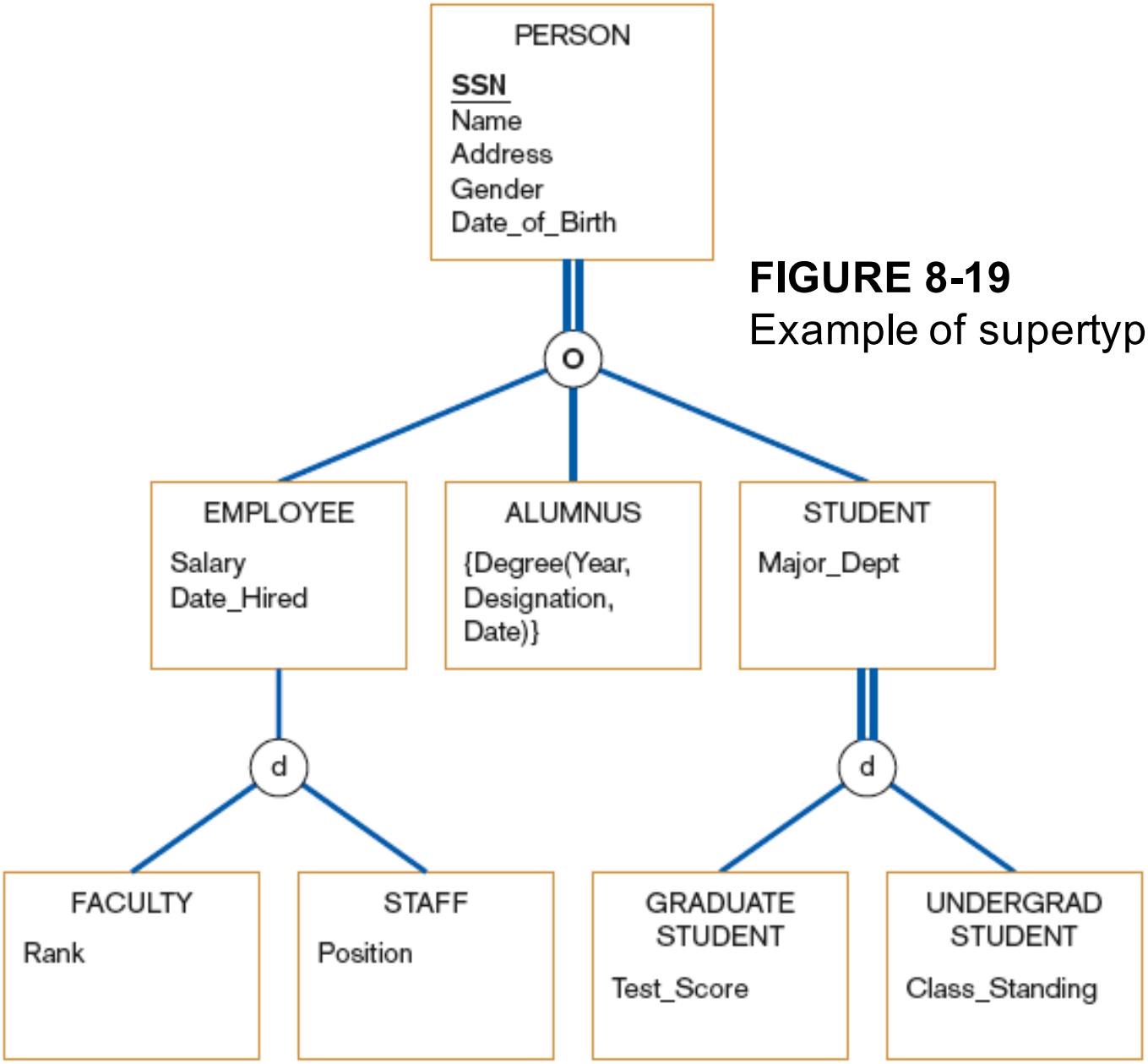


FIGURE 8-19
Example of supertype/subtype hierarchy

Business Rules

- **Business rules**: specifications that preserve the integrity of the logical data model
  - Captured during requirements determination
  - Stored in CASE repository as they are documented

Business Rules (Cont.)

- Four basic types of business rules are:
  - *Entity integrity*: unique, non-null identifiers
  - *Referential integrity constraints*: rules governing relationships between entity types
  - *Domains*: constraints on valid values for attributes
  - *Triggering operations*: other business rules that protect the validity of attribute values

Domains

- **Domain**: the set of all data types and values that an attribute can assume
- Several advantages
  - Verify that the values for an attribute are valid
  - Ensure that various data manipulation operations are logical
  - Help conserve effort in describing attribute characteristics

Triggering Operations

- **Trigger**: an assertion or rule that governs the validity of data manipulation operations such as insert, update and delete

Triggering Operations

- Includes the following components:
  - *User rule*: statement of the business rule to be enforced by the trigger
  - *Event*: data manipulation operation that initiates the operation
  - *Entity Name*: name of entity being accessed or modified
  - *Condition*: condition that causes the operation to be triggered
  - *Action*: action taken when the operation is triggered

Role of Packaged Conceptual Data Models – Database Patterns

- Packaged data models provide generic models that can be customized for a particular organization's business rules.
- **Universal data models** are templates for
  - one or more core subject areas such as:
    - Customers, products, accounts, documents
  - and/or core business functions such as:
    - Purchasing, accounting, receiving, etc.

Role of Packaged Conceptual Data Models – Database Patterns (Cont.)

- **Industry-specific data models** are designed to be used by organizations within specific industries.
- These models are based on the premise that data model patterns for organizations are similar within a particular industry.

Benefits of Database Patterns and Packaged Data Models

- Dramatically reduced implementation times and costs
  - Provides a starting point for asking requirements questions
- Higher-quality models
  - Represent “best practice” data modeling techniques and data model components whose quality often exceeds that which can be achieved by internal development teams, given typical organizational pressures

Electronic Commerce Application: Conceptual Data Modeling

- Five general categories of information were identified for Pine Valley Furniture's WebStore.
- Next step was to define each item.
- The final step was to identify the interrelationships between the four entities.

Summary

- In this chapter you learned how to:

  - ✓ Concisely define each of the following key data modeling terms: entity type, attribute, multivalued attribute, relationship, degree, cardinality, business rule, associative entity, trigger, supertype, subtype.
  - ✓ Draw an entity-relationship (E-R) diagram to represent common business situations.
  - ✓ Explain the role of conceptual data modeling in the overall analysis and design of an information system.

Summary (Cont.)

- In this chapter you learned how to:
  - ✓ Explain the role of prepackaged database models (patterns) in data modeling.
  - ✓ Distinguish between unary, binary, and ternary relationships and give an example of each.
  - ✓ Define four basic types of business rules in a conceptual data model.
  - ✓ Relate data modeling to process and logic modeling as different views of describing an information system.