

**Princess Noura university
Department of Computer Science**

**Chapter 8
Analyzing System Data
Requirements**

Learning Objectives

- ✓ Define key data modeling terms.
- ✓ Draw entity-relationship (E-R) to represent common business situations.
- ✓ Explain the role of conceptual data modeling in IS analysis and design.
- ✓ Distinguish between unary, binary, and ternary relationships.
- ✓ Define four types of business rules.
- ✓ Relate data modeling to process and logic modeling.

Figure 7-1 Systems development life cycle with the analysis phase highlighted

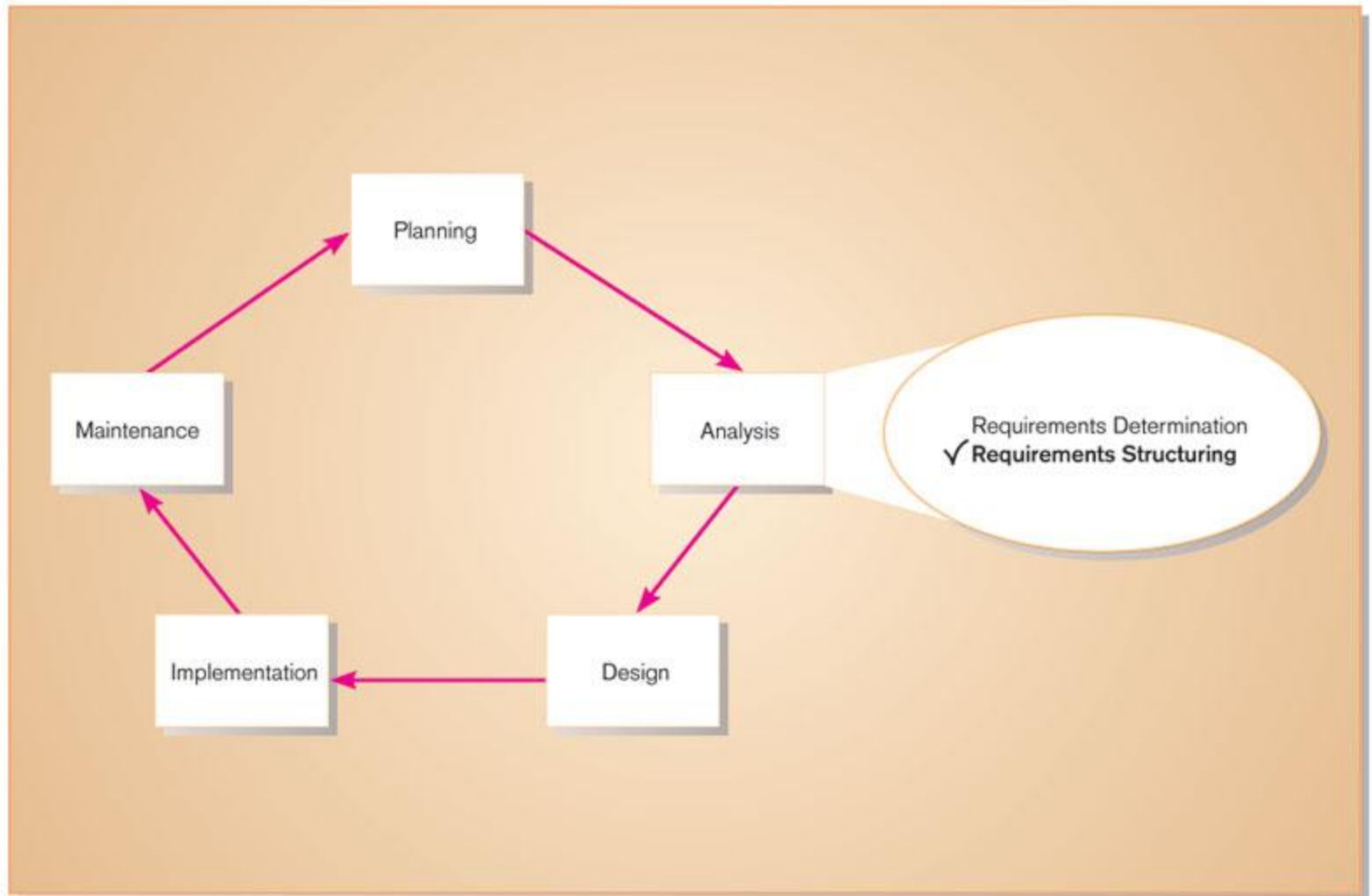
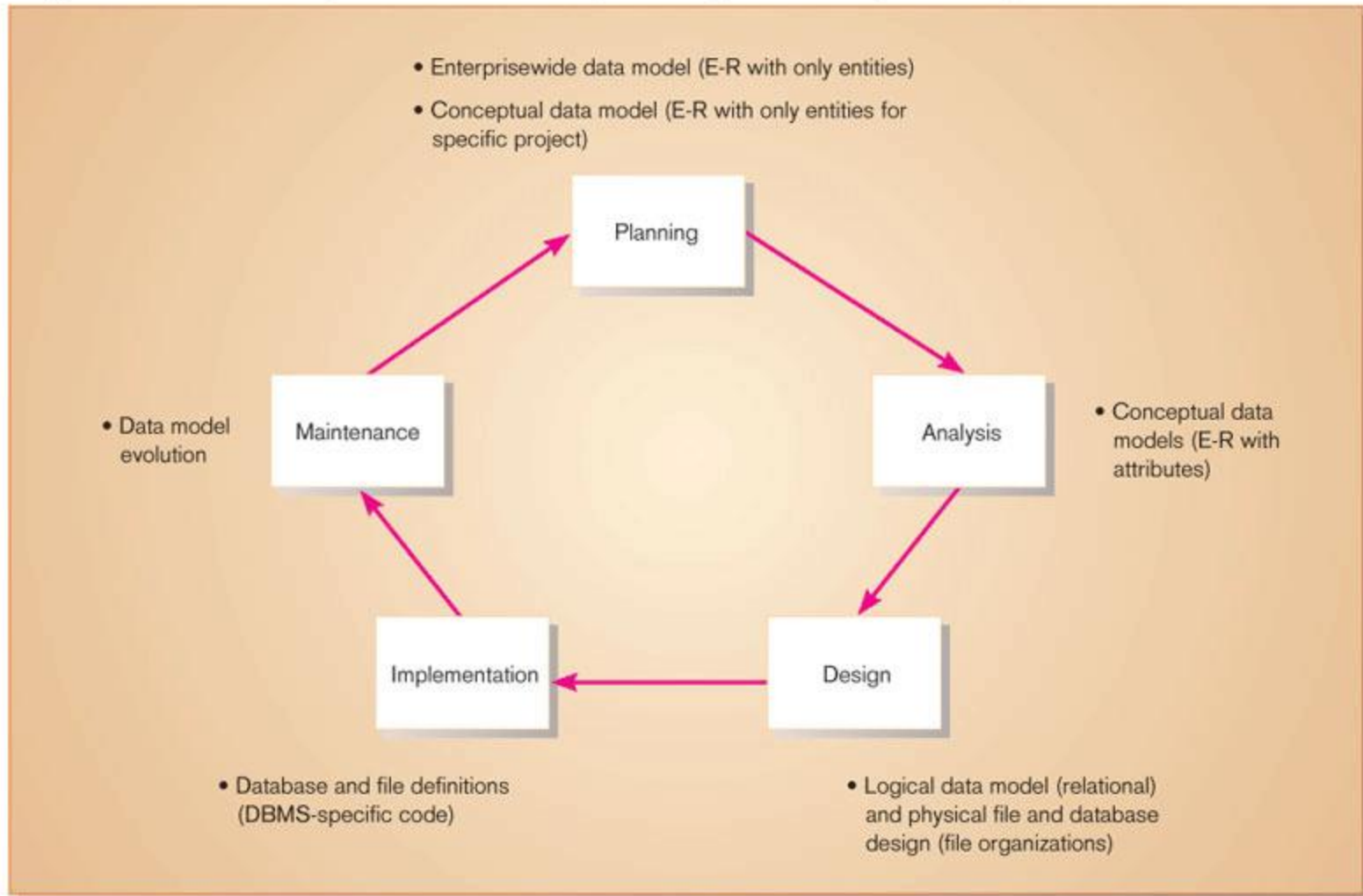


Figure 9-2 Relationship between data modeling and the systems development life cycle



Conceptual Data Modeling

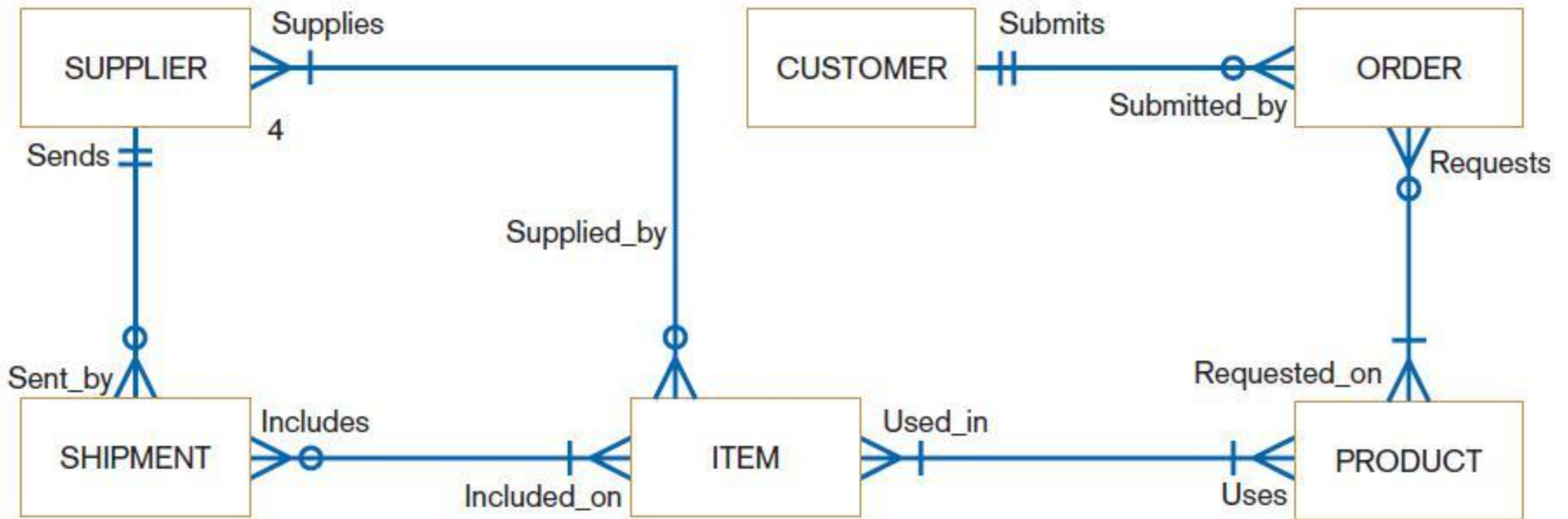
- ◆ A detailed model that captures the overall structure of data in an organization
- ◆ Independent of any database management system (DBMS) or other implementation considerations

Process of Conceptual Data Modeling

- ◆ 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

Deliverables and Outcome (cont.)

- ◆ Primary deliverable is an entity-relationship (E-R) diagram
- ◆ Second deliverable is a set of entries about data objects to be stored in repository or project dictionary.
 - Repository links data, process, and logic models of an information system.
 - Data elements included in the 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.



Key



Cardinalities



8-3: Sample conceptual data model

Gathering Information for Conceptual Data Modeling

◆ Two perspectives

■ Top-down

- ◆ Data model is derived from an intimate understanding of the business.

■ Bottom-up

- ◆ Data model is derived by reviewing specifications and business documents-computer displays, reports and business forms.

Requirements Determination Questions for Data Modeling

- ◆ What are subjects/objects of the business? “people, places, things, events..etc”
 - Data entities and descriptions
- ◆ What unique characteristics distinguish between subjects/objects of the same type?
 - Primary keys
- ◆ What characteristics describe each subject/object?
 - Attributes and secondary keys
- ◆ How do you use the data? “do you refer to the data, modify it and destroy it”
 - Security controls and user access privileges

Requirements Determination

Questions for Data Modeling (cont.)

- ◆ Over what period of time are you interested in the data?
 - Cardinality and time dimensions
- ◆ Are all instances of each object the same? “are there special kinds of each object that are described or handled differently?”
 - Supertypes, subtypes, and aggregations
- ◆ What events occur that imply associations between objects?
 - Relationships and cardinalities

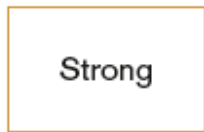
Introduction to Entity-Relationship (E-R) Modeling

- ◆ Entity-Relationship (E-R) data model
 - A detailed, logical representation of the entities, associations and data elements for an organization or business area.
- ◆ Entity-Relationship (E-R) diagram
 - A graphical representation of an E-R model.

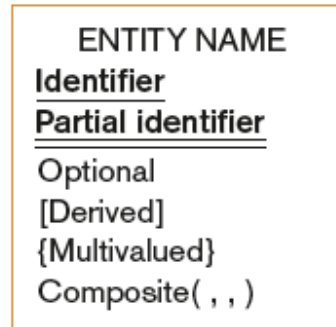
Basic E-R Notation

- ◆ Notation uses three main constructs
 - Data entities
 - Relationships
 - Attributes

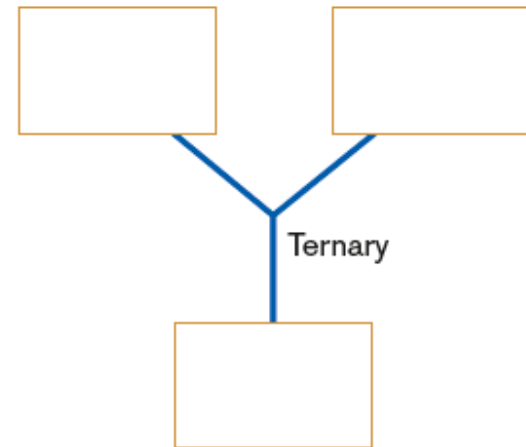
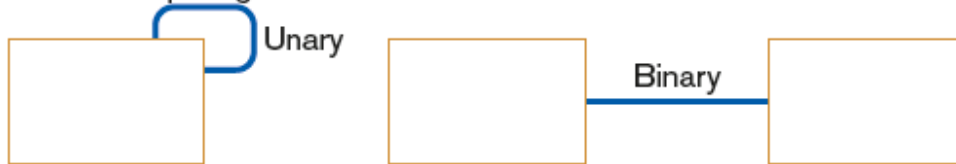
Entity Types



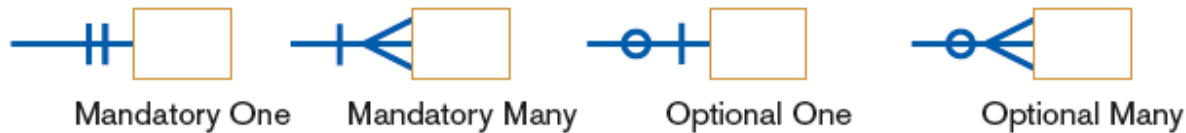
Attributes



Relationship Degrees



Relationship Cardinality



8-5: Basic E-R notation

Entities

- An entity is a person, place, object, event or concept about which data is to be maintained
- Some examples:
 - Person: EMPLOYEE, STUDENT, PATIENT
 - Place: STORE, WAREHOUSE, STATE
 - Object: MACHINE, BUILDING, PRODUCT
 - Event: SALE, REGISTRATION
 - Concept: ACCOUNT, COURSE

Entities cont.

Entity type:

- Collection of entities with common characteristics
- We use a concise, simple noun and capital letters in naming the entity.

Entity instance:

- Single occurrences of an entity

Entities cont.

- ◆ An entity name must be:
 - ◆ Singular noun (ex. CUSTOMER)
 - ◆ Descriptive (ex. PURCHASE ORDER)
 - ◆ Concise (ex. REGISTRATION)
 - ◆ The result of the event not the activity (ex. ASSIGNMENT)
 - ◆ Capital letters

Attributes

- ◆ An attribute is a named property or characteristic of an entity
- ◆ Examples:
 - STUDENT: Student_ID, Student_Name, Home_Address, Phone_Number, Major
 - EMPLOYEE: Employee_ID, Employee_Name, Payroll_Address, Skill
- ◆ We use initial capital letter, followed by lowercase letters, and nouns in naming an attribute.

Attributes cont.

◆ An attribute name:

- Noun
- Unique
- Similar attributes use similar but distinguishing names.

Identifier Attributes

◆ Candidate key

- Attribute (or combination of attributes) that uniquely identifies each instance of an entity type.

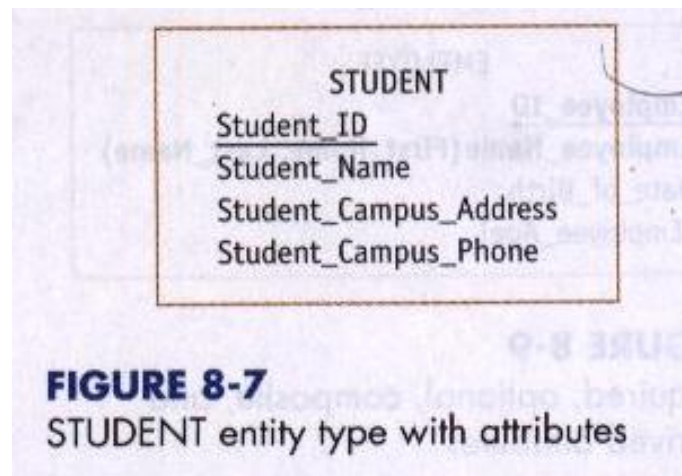
◆ Identifier

- A candidate key that has been selected as the unique identifying characteristic for an entity type.
- Represented by placing solid underline below the identifier.

Identifier Attributes cont.

- ◆ Selection rules for an identifier
 1. Choose a candidate key that will not change its value.
 2. Choose a candidate key that will never be null.
 3. Use single value keys for large composite keys.

Example:

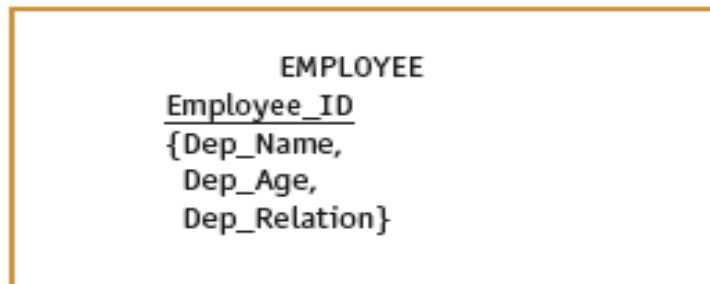


Multivalued Attributes

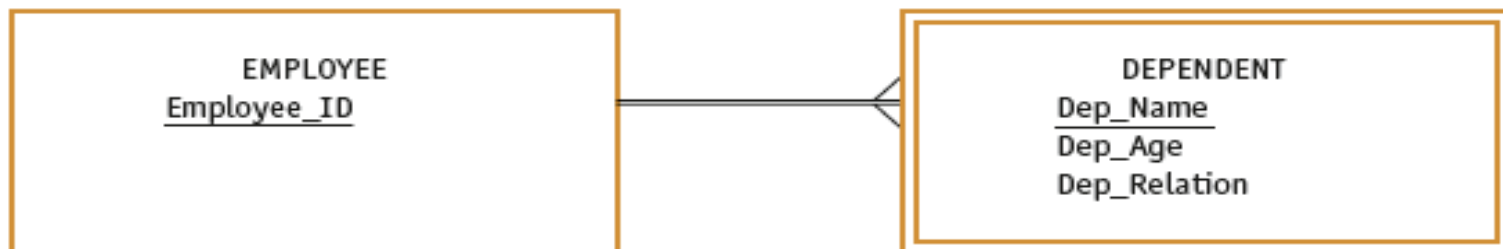
- ◆ An attribute that may take on more than one value for each entity instance
- ◆ (ex. Employee Skills)
- ◆ Represented on E-R Diagram in two ways:
 - Special symbol
 - weak entity



(a) Multivalued attribute skill



(b) Repeating group of dependent data



(c) Weak entity for dependent data

Attributes cont.

◆ Composite attribute

- An attribute that has a meaningful component parts.

◆ Derived attribute

- An attribute whose value can be computed from related attribute values.

◆ Example:

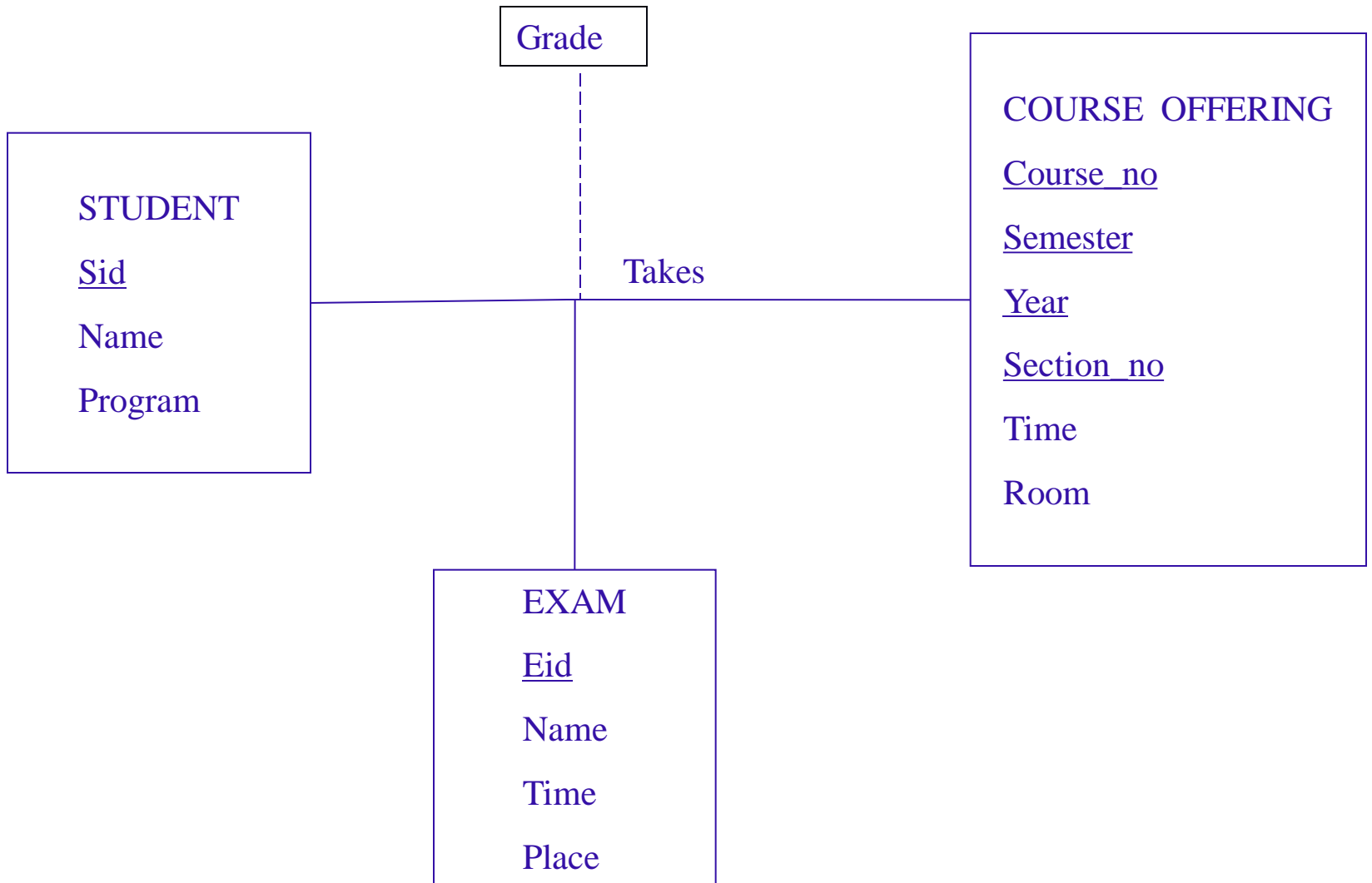
```
EMPLOYEE
Employee_ID
Employee_Name(First,Last)
Date_of_Birth
[Employee_Age]
```

Example

- ❖ Consider a database that records the grades of **students** in different **exams** of different **course offerings**.
 - a) assume that course offerings include time, room; and can be identified uniquely by course_no, semester, year, section_no.
 - b) students, including student-id, name, and program;
 - c) exam, including eid, name, place, time.

Furthermore, grades of students in different exams of course offerings must be appropriately modeled.

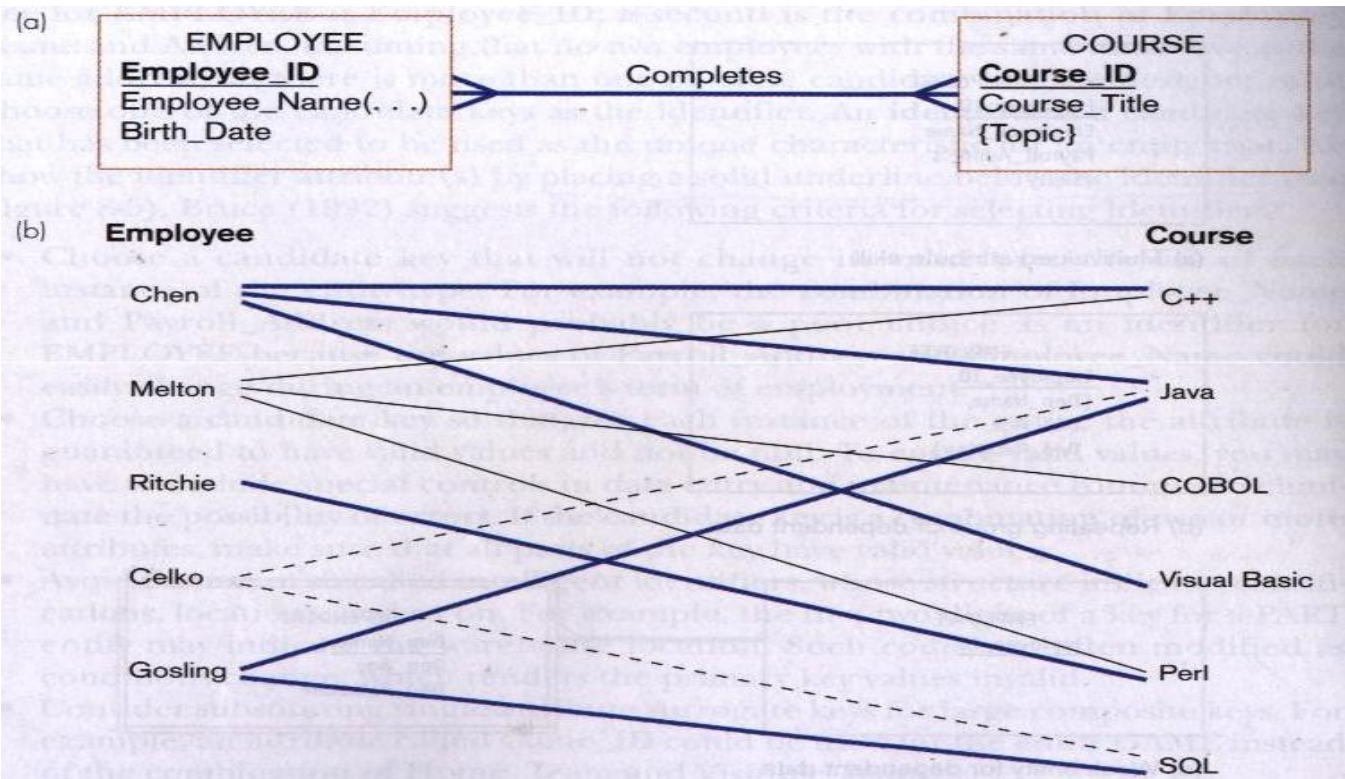
- ❖ Construct an E-R diagram that models exams as entities, and uses a ternary relationship, for the above database.



E-R diagram for marks database.

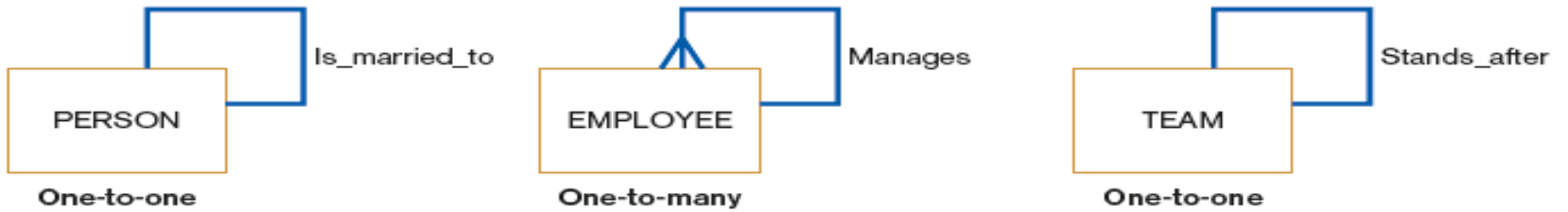
Relationship

- ◆ Association between the instances of one or more entity types.
- ◆ A relationship name is verb phrase.

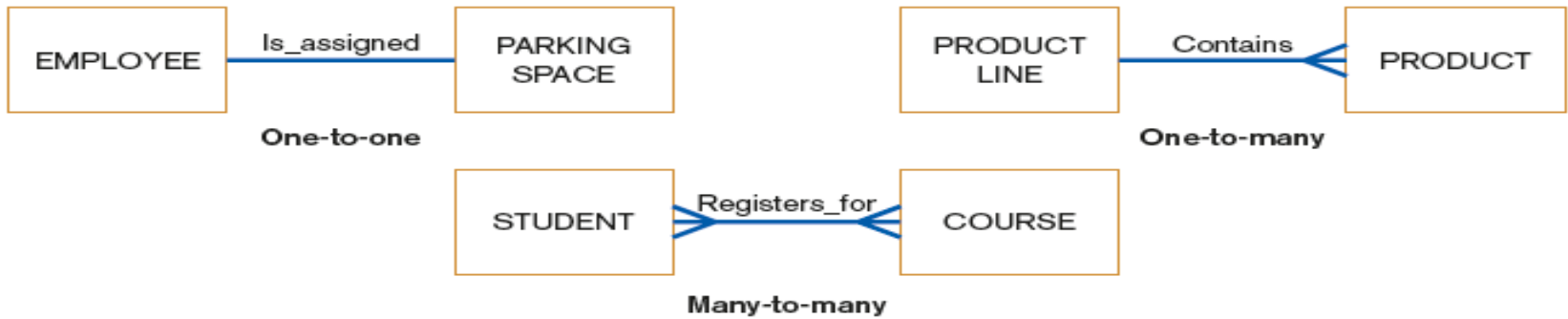


Degree of Relationship

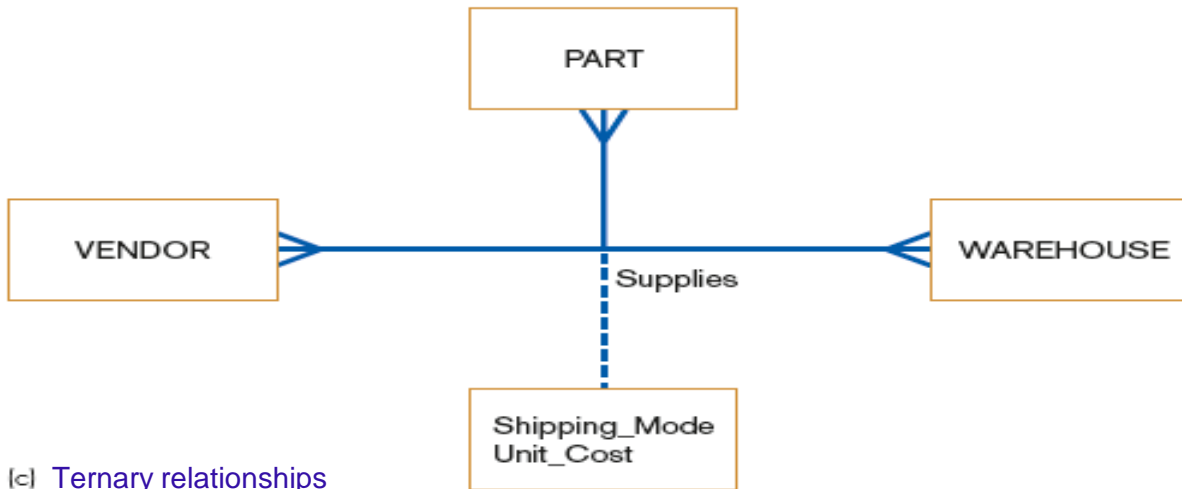
- ◆ Degree: number of entity types that participate in a relationship
- ◆ Three cases
 - **Unary:** between two instances of one entity type
 - **Binary:** between the instances of two entity types (the most common)
 - **Ternary:** among the instances of three entity types



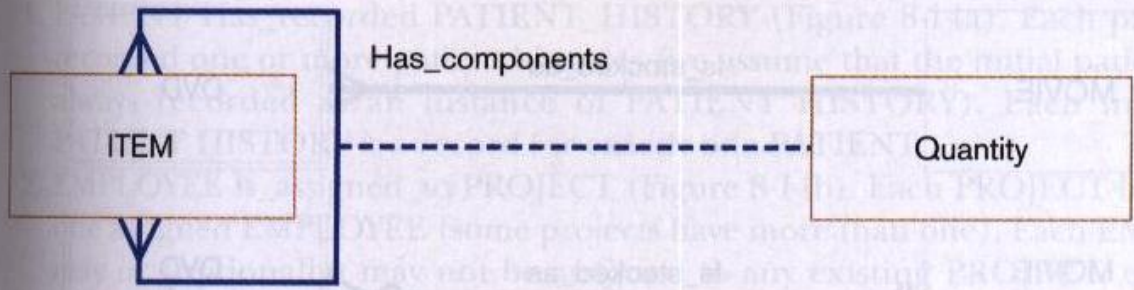
(a) Unary relationships



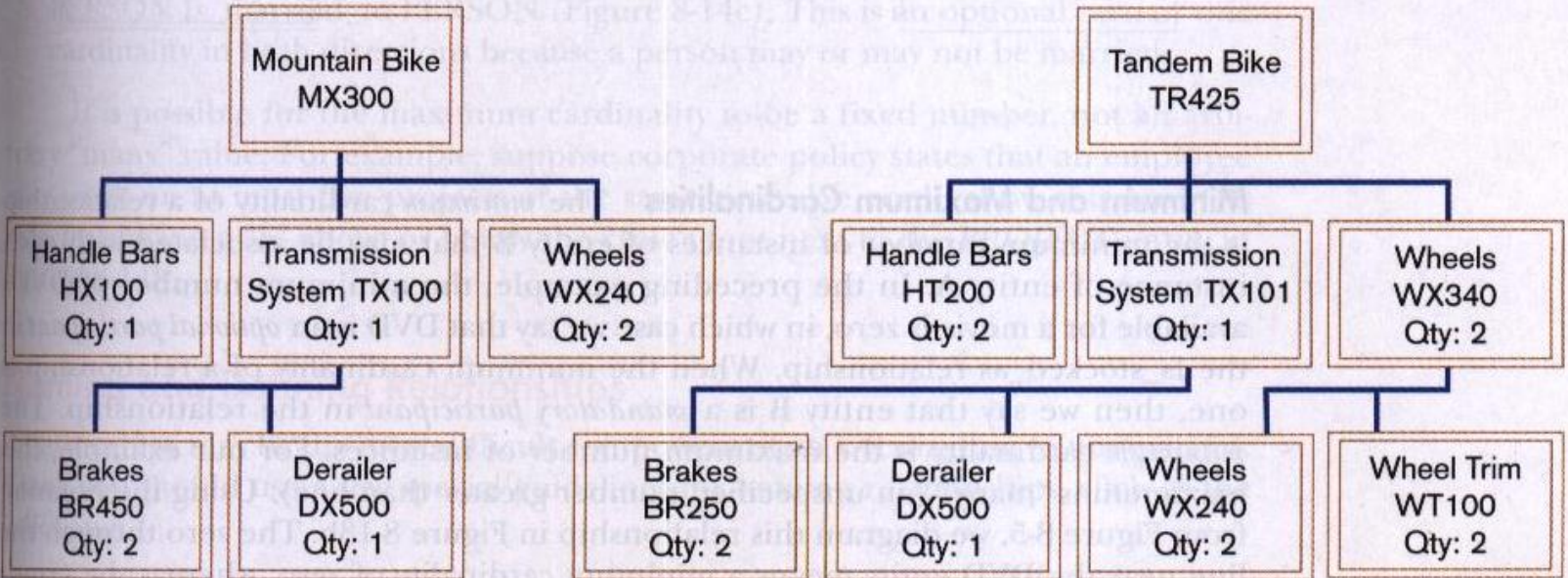
(b) Binary relationships



(c) Ternary relationships



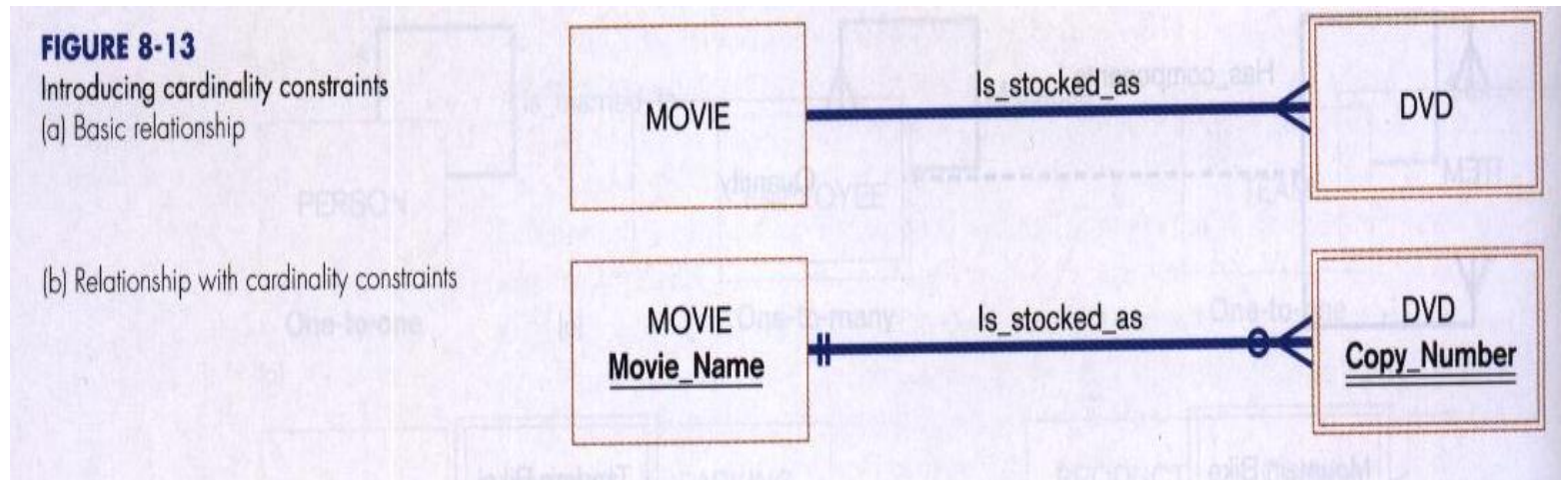
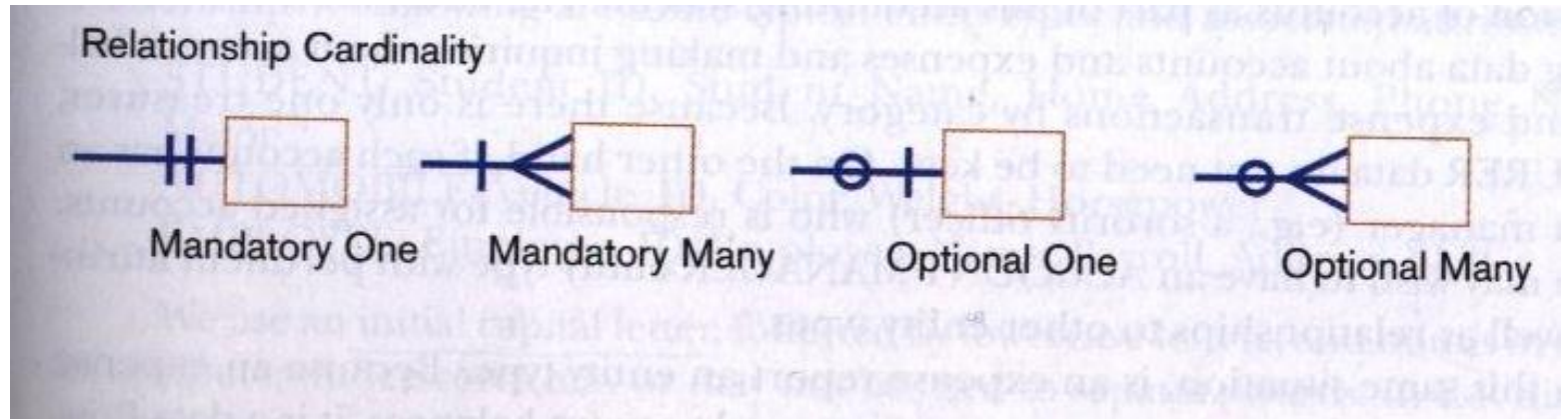
(a)

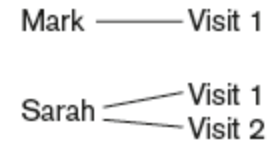


(b)

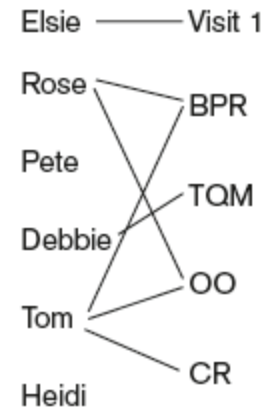
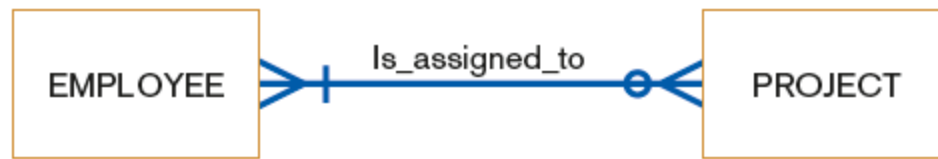
Cardinality

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

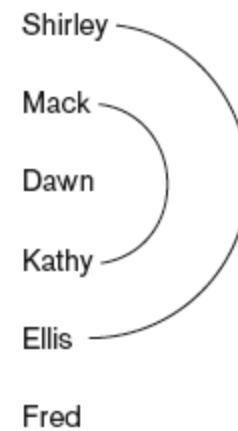
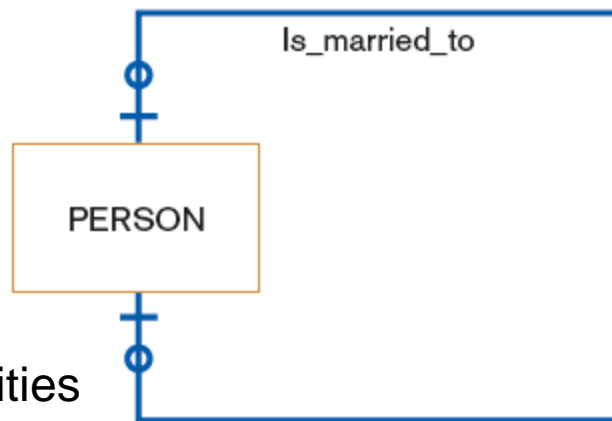




(a) Mandatory cardinalities



(b) One optional, one mandatory cardinality



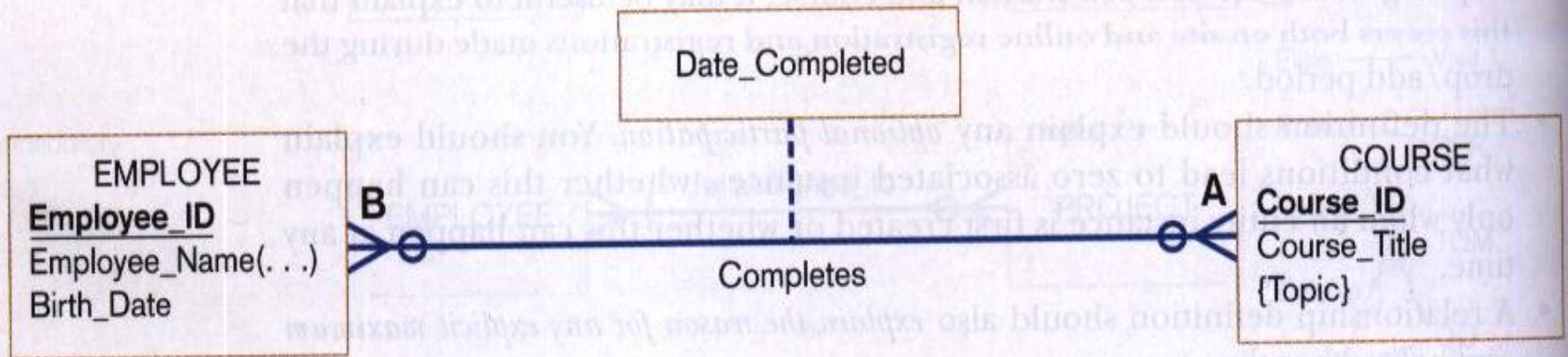
(c) Optional cardinalities

Associative Entities

- ◆ An entity type that associates the instances of one or more entity types and contains attributes that are unusual to the relationship between those entity instances
- ◆ An associative entity is:
 - An entity
 - A relationship
- ◆ This is the preferred way of illustrating a relationship with attributes

◆ Example:

Employee_ID	Course_Name	Date_Completed
549-23-1948	Basic Algebra	March 2006
629-16-8407	Software Quality	June 2009
816-30-0458	Software Quality	February 2009
549-23-1948	C Programming	May 2009



(a)



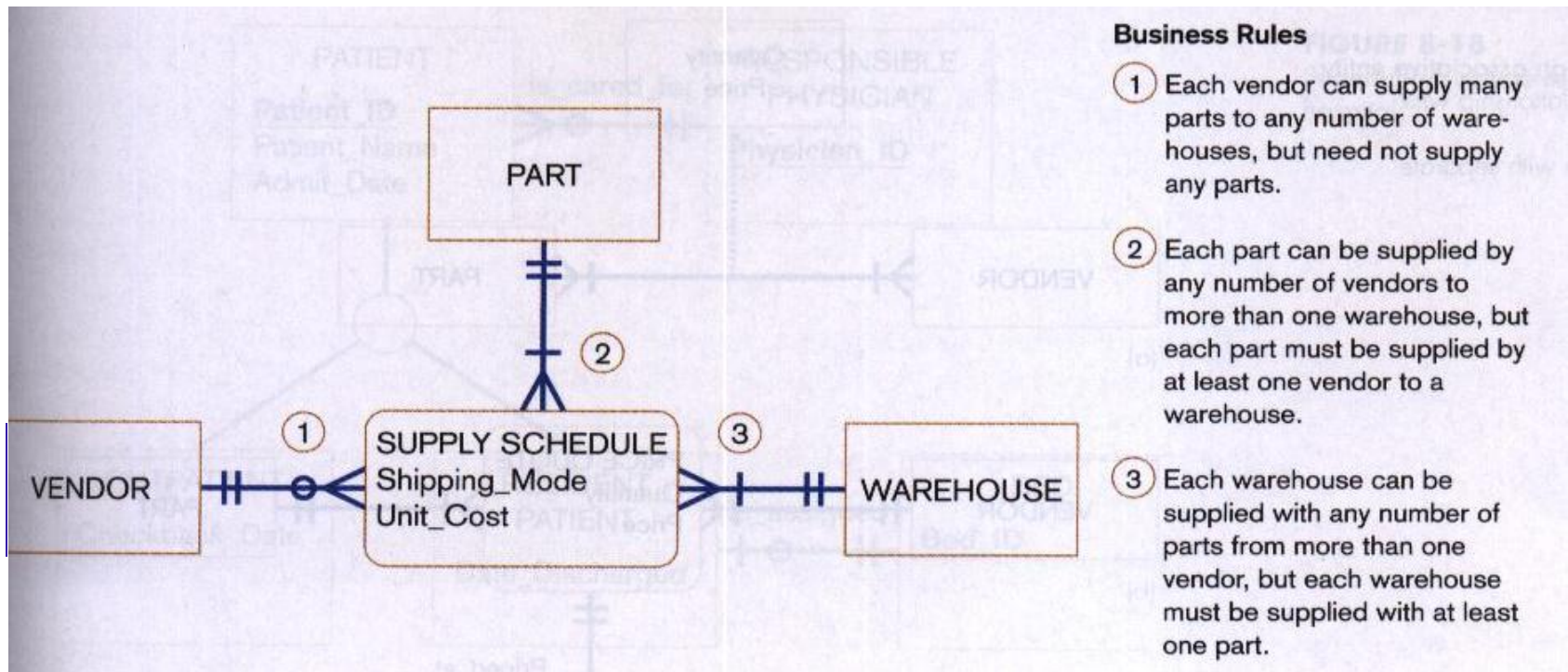
(b)

(a) A relationship with an attribute

(b) An associative entity

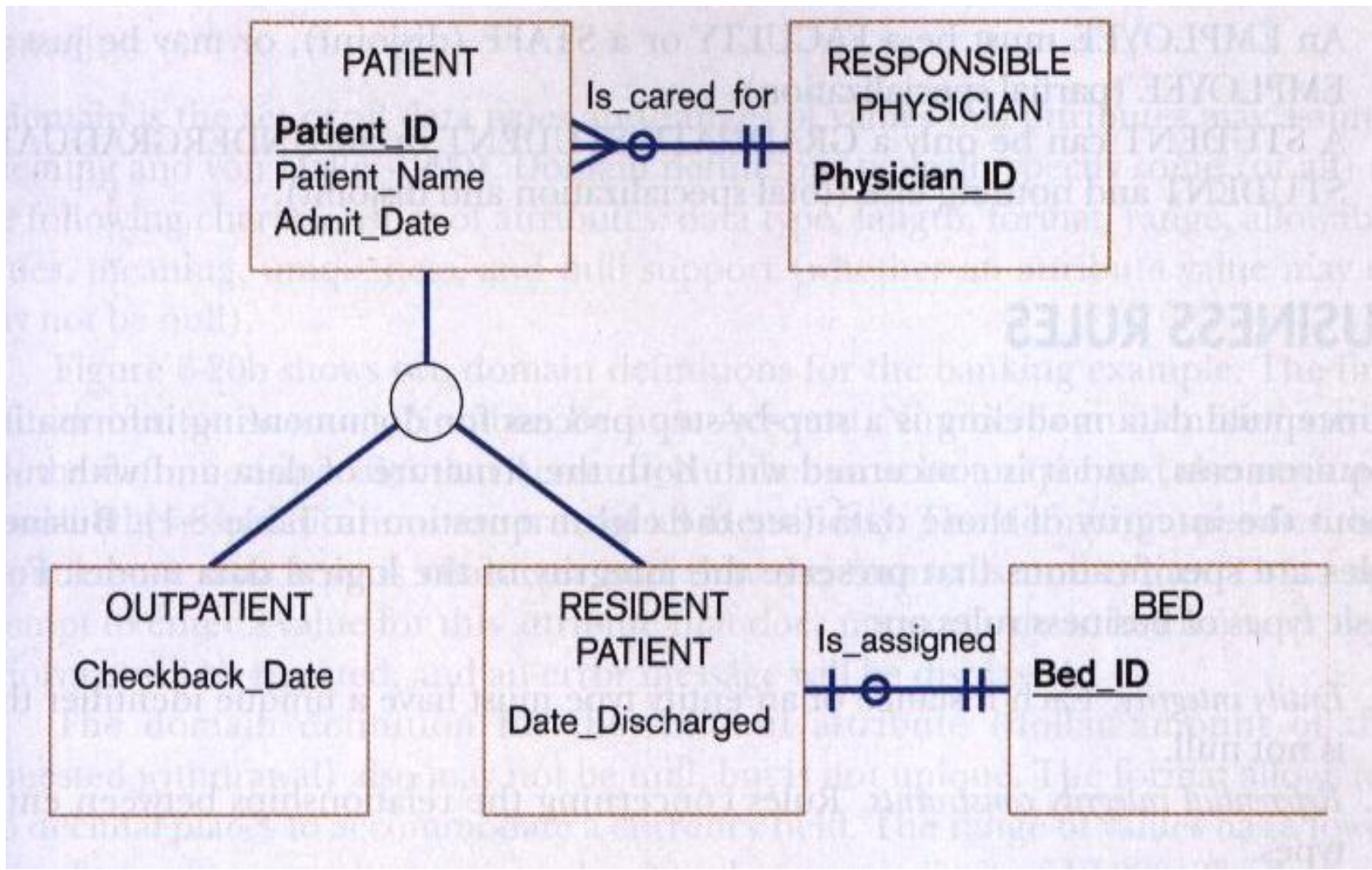
Slide 29) attribute in relationship.

An associative entity:



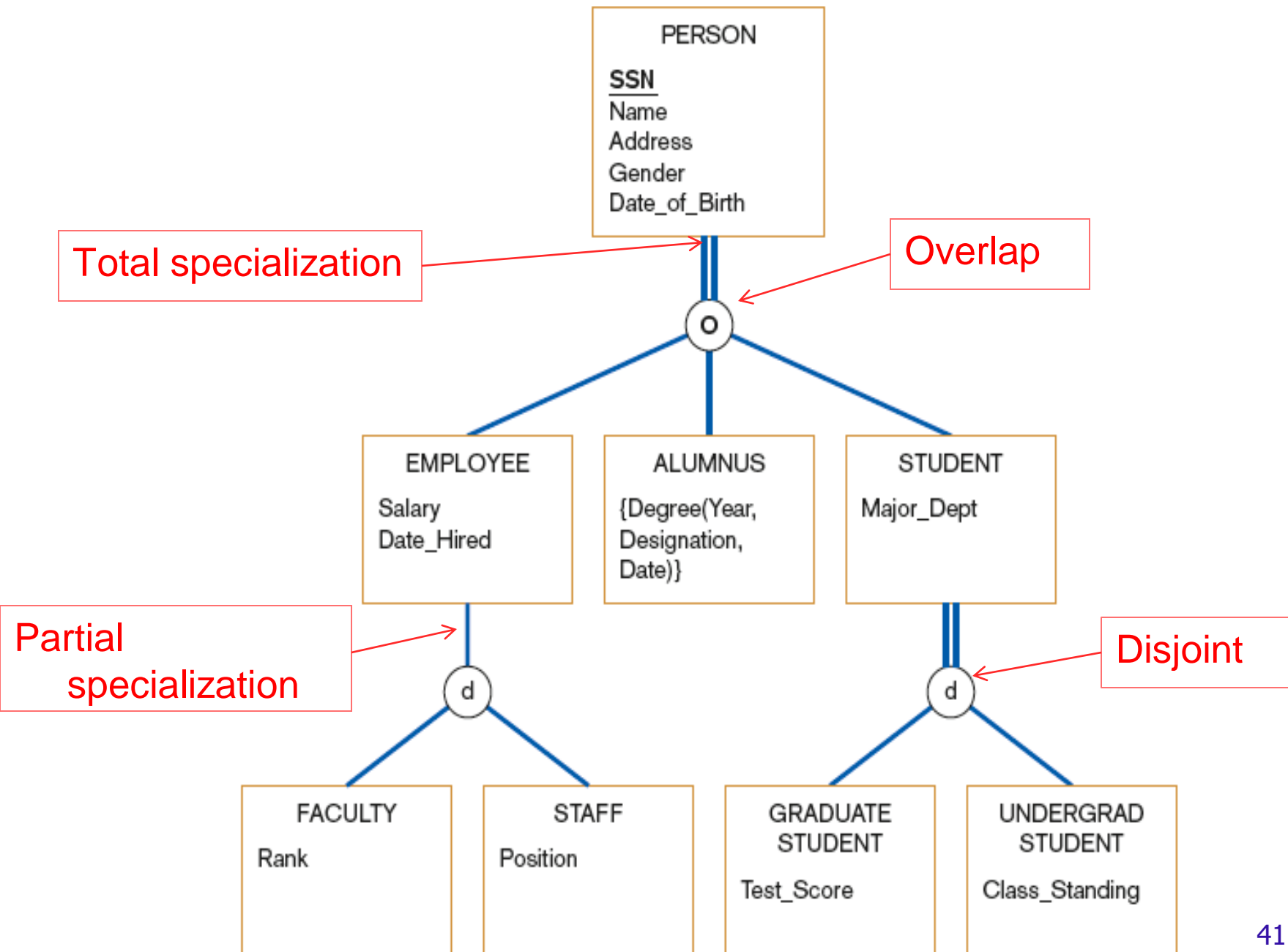
Supertypes and Subtypes

- ◆ Subtype: a sub grouping of the entities in an entity type that shares common attributes or relationships distinct from other subtypes
- ◆ Supertype: a generic entity type that has a relationship with one or more subtype



Rules for Supertype/Subtypes Relationships

- ◆ Total specialization rule: an entity instance of the supertype must be an instance of one of the subtypes
- ◆ Partial specialization rule: an entity instance of the supertype may or may not be an instance of one of the subtypes
- ◆ Disjoint rule: an entity instance of the supertype can be an instance of only one subtype
- ◆ Overlap rule: an entity instance of the supertype may be an instance of multiple subtypes



Business Rules

- ◆ Specifications that preserve the integrity of the logical data model
- ◆ Four types
 - Entity integrity: each instance of an entity must have a unique, non-null identifiers
 - Referential integrity constraints: rules concerning the relationships between entities
 - Domains: constraints on valid values for attributes
 - Triggering operations: other business rules regarding attribute values

Domains

- ◆ The set of all data types and ranges of values that an attribute can assume
Ex. Data types, length, format, range, meaning .. Etc

- ◆ Several advantages
 1. Verify that the values for an attribute (stored, insert) are valid
 2. Help conserve effort in describing attribute characteristics

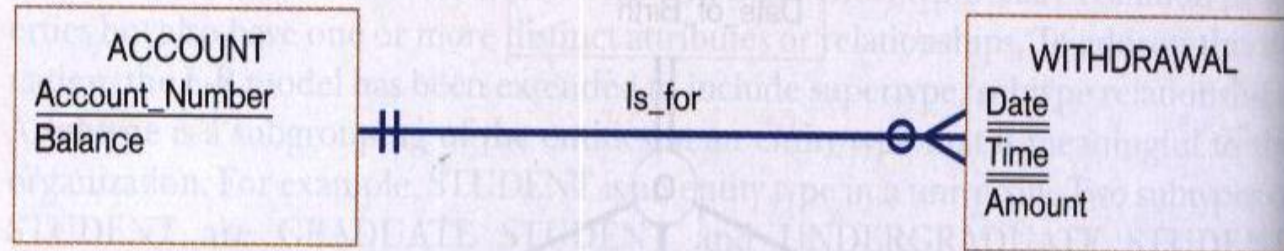
Triggering Operations

- ◆ A declaration or rule that governs the validity of data manipulation operations such as insert, update and delete
- ◆ 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

FIGURE 8-20

Examples of business rules

(a) Simple banking relationship



(b) Typical domain definitions

Name: Account_Number
Meaning: Customer account number in bank
Data type: Character
Format: nnn-nnnn
Uniqueness: Must be unique
Null support: Non-null

Name: Amount
Meaning: Dollar amount of transaction
Data type: Numeric
Format: 2 decimal places
Range: 0-10,000
Uniqueness: Nonunique
Null support: Non-null

(c) Typical triggering operation

User rule: WITHDRAWAL Amount may not exceed ACCOUNT Balance
Event: Insert
Entity Name: WITHDRAWAL
Condition: WITHDRAWAL Amount > ACCOUNT Balance
Action: Reject the insert transaction