Modeling Data in the Organization

- Concisely define each of the following key terms: **business rule, term, fact, entity-relationship model (E-R model), entity-relationship diagram (E-R diagram), entity, entity type, entity instance, strong entity type, weak entity type, identifying owner, identifying relationship, attribute, required attribute, optional attribute, composite attribute, simple attribute, multivalued attribute, derived attribute, identifier, composite identifier, relationship type, relationship instance, associative entity, degree, unary relationship, binary relationship, ternary relationship, cardinality constraint, minimum cardinality, maximum cardinality,** and **time stamp**.
- State reasons why many system developers believe that data modeling is the most important part of the systems development process.
- Write good names and definitions for entities, relationships, and attributes.
- Distinguish unary, binary, and ternary relationships and give a common example of each.
- Model each of the following constructs in an E-R diagram: composite attribute, multivalued attribute, derived attribute, associative entity, identifying relationship, and minimum and maximum cardinality constraints.
- Draw an E-R diagram to represent common business situations.
- Convert a many-to-many relationship to an associative entity type.
- Model simple time-dependent data using time stamps and relationships in an E-R diagram.

Business Rules

- Business rules, the foundation of data models, are derived from policies, procedures, events, functions, and other business objects, and they state constraints on the organization.
- Business rules represent the language and fundamental structure of an organization (Hay, 2003).
- Business rules formalize the understanding of the organization by organization owners, managers, and leaders with that of information systems architects.
- Business rules are important in data modeling because they govern how data are handled and stored.
- Examples of basic business rules are data names and definitions. This lecture explains guidelines for the clear naming and definition of data objects in a business.
- In terms of conceptual data modeling, names and definitions must be provided for the main data objects: entity types (e.g., Customer), attributes (Customer Name), and relationships (Customer Places Orders).
- Other business rules may state constraints on these data objects.
- These constraints can be captured in a data model, such as an entity-relationship diagram, and associated documentation.
- Additional business rules govern the people, places, events, processes, networks, and objectives of the organization, which are all linked to the data requirements through other system documentation.

E-R Models

- After decades of use, the E-R model remains the mainstream approach for conceptual data modeling. Its popularity stems from factors such as relative ease of use, widespread computer-aided software engineering (CASE) tool support, and the belief that entities and relationships are natural modeling concepts in the real world.

- The E-R model is most used as a tool for communications between database designers and end users during the analysis phase of database development.

- The E-R model is used to construct a conceptual data model, which is a representation of the structure and constraints of a database that is independent of software (such as a database management system).

There is not a Standard Notation for E-R Modeling

- The E-R model was introduced in a key article by Chen (1976), in which he described the main constructs of the E-R model—entities and relationships—and their associated attributes.
- The model has subsequently been extended to include additional constructs by Chen and others; for example, see Teorey et al. (1986) and Storey (1991).
- The E-R model continues to evolve, but unfortunately there is not yet a standard notation for E-R modeling. Song et al. (1995) present a side-by-side comparison of 10 different E-R modeling notations, explaining the major advantages and disadvantages of each approach.
- Because data modeling software tools are now commonly used by professional data modelers, we adopt for use a variation of the notation used in professional modeling tools.
- A reference material was posted on the lecture slides page to help translate between our notation and other popular E-R diagramming notations.

Modeling is Important

- Many systems developers believe that data modeling is the most important part of the systems development process for the following reasons:
  - The characteristics of data captured during data modeling are crucial in the design of databases, programs, and other system components. The facts and rules captured during the process of data modeling are essential in assuring data integrity in an information system.
  - Data rather than processes are the most complex aspect of many modern information systems and hence require a central role in structuring system requirements. Often the goal is to provide a rich data resource that might support any type of information inquiry, analysis, and summary.
  - Data tend to be more stable than the business processes that use that data. Thus, an information system design that is based on a data orientation should have a longer useful life than one based on a process orientation.

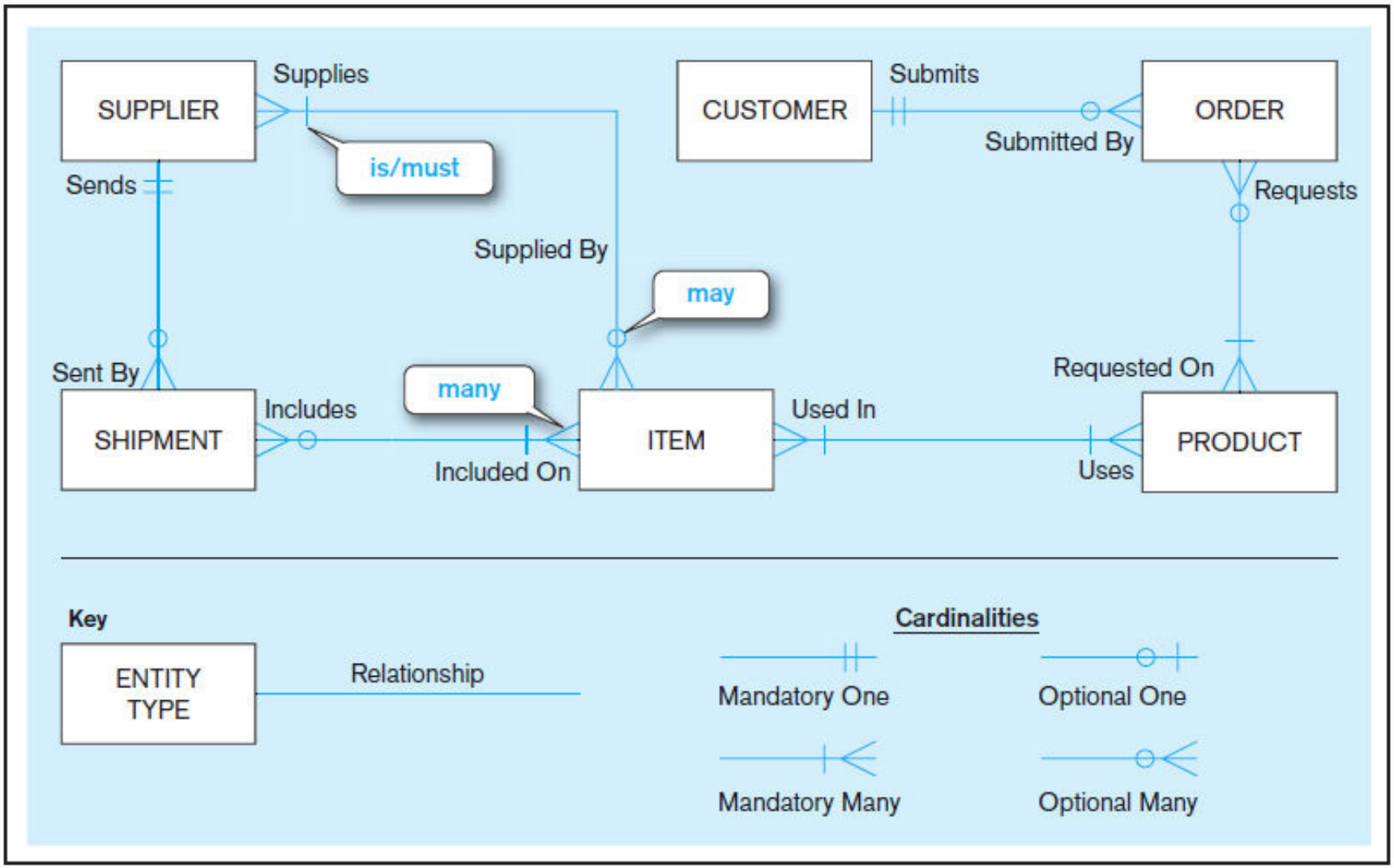
You may not have to Develop a Data Model from Scratch

- In an actual work environment, you may not have to develop a data model from scratch. Because of the increased acceptance of packaged software (for example, enterprise resource planning with a predefined data model) and purchased business area or industry data, your job of data modeling has a jump start.
- This is good because such components and patterns give you a starting point based on generally accepted practices.
- However, your job is not done for several reasons:
  - There are still many times when a new, custom-built application is being developed along with the associated database.
    - The business rules for the business area supported by this application need to be modeled.
  - Purchased applications and data models need to be customized for your particular setting.
    - Predefined data models tend to be very extensive and complex; hence, they require significant data modeling skill to tailor the models to be effective and efficient in a given organization.
    - Although this effort can be much faster, thorough, and accurate than starting from scratch, the ability to understand a particular organization to match the data model to its business rules is an essential task.

The E-R Model: An Overview

- An entity-relationship model (E-R model) is a detailed, logical representation of the data for an organization or for a business area.
- The E-R model is expressed in terms of entities in the business environment, the relationships (or associations) among those entities, and the attributes (or properties) of both the entities and their relationships.
- An E-R model is normally expressed as an entity-relationship diagram (E-R diagram, or ERD), which is a graphical representation of an E-R model.

Sample E-R diagram



The Entities in the Sample ERD are:

CUSTOMER	A person or an organization that has ordered or might order products. *Example*: L. L. Fish Furniture.
PRODUCT	A type of furniture made by Pine Valley Furniture that may be ordered by customers. Note that a product is not a specific bookcase, because individual bookcases do not need to be tracked. *Example*: A 6-foot, 5-shelf, oak bookcase called O600.
ORDER	The transaction associated with the sale of one or more products to a customer and identified by a transaction number from sales or accounting. *Example*: The event of L. L. Fish buying one product O600 and four products O623 on September 10, 2010.
ITEM	A type of component that goes into making one or more products and can be supplied by one or more suppliers. *Example*: A 4-inch ball-bearing caster called I-27–4375.
SUPPLIER	Another company that may provide items to Pine Valley Furniture. *Example*: Sure Fasteners, Inc.
SHIPMENT	The transaction associated with items received in the same package by Pine Valley Furniture from a supplier. All items in a shipment appear on one bill-of-lading document. *Example*: The receipt of 300 I-27-4375 and 200 I-27-4380 items from Sure Fasteners, Inc., on September 9, 2010.

The symbols at the end of each line on an ERD specify relationship cardinalities, which represent how many entities of one kind relate to how many entities of another kind.

1. A SUPPLIER may supply many ITEMs (by "may supply," we mean the supplier may not supply any items). Each ITEM is supplied by any number of SUPPLIERs (by "is supplied," we mean that the item must be supplied by at least one supplier). See annotations in Figure 2-1 that correspond to underlined words.
2. Each ITEM must be used in the assembly of at least one PRODUCT and may be used in many products. Conversely, each PRODUCT must use one or more ITEMs.
3. A SUPPLIER may send many SHIPMENTs. However, each shipment must be sent by exactly one SUPPLIER. Notice that sends and supplies are separate concepts. A SUPPLIER may be able to supply an item, but may not yet have sent any shipments of that item.
4. A SHIPMENT must include one (or more) ITEMs. An ITEM may be included on several SHIPMENTs.
5. A CUSTOMER may submit any number of ORDERs. However, each ORDER must be submitted by exactly one CUSTOMER. Given that a CUSTOMER may not have submitted any ORDERs, some CUSTOMERs must be potential, inactive, or some other customer possibly without any related ORDERs.
6. An ORDER must request one (or more) PRODUCTs. A given PRODUCT may not be requested on any ORDER, or may be requested on one or more orders.

There are actually two business rules for each relationship, one for each direction from one entity to the other. Note that each of these business rules roughly follows a certain grammar:

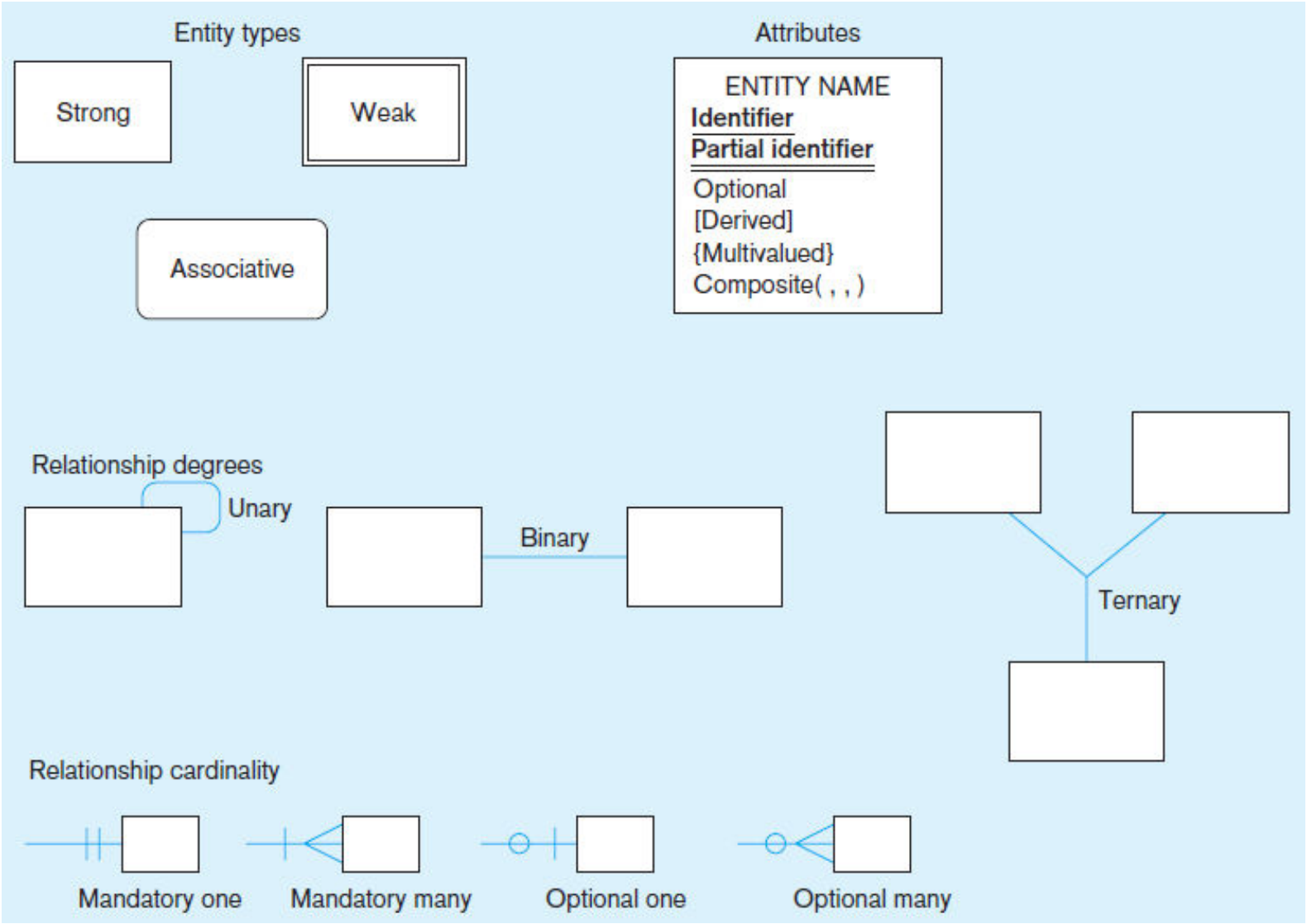
<entity> <minimum cardinality> <relationship> <maximum cardinality> <entity>

For example, rule 5 is:

<CUSTOMER> <may> <Submit> <any number> <ORDER>

This grammar gives you a standard way to put each relationship into a natural English business rule statement.

Different Notations that are Commonly Used In E-R Drawing Tools



Modeling the Rules of the Organization

- Business rules and policies govern creating, updating, and removing data in an information processing and storage system; thus they must be described along with the data to which they are related.

- For example, the policy “every student in the university must have a faculty adviser” forces data (in a database) about each student to be associated with data about some student adviser.

- Also, the statement “a student is any person who has applied for admission or taken a course or training program from any credit or noncredit unit of the university” not only defines the concept of “student” but also states a policy of the university (e.g., implicitly, alumni are students, and a high school student who attended a college fair but has not applied is not a student, assuming the college fair is not a noncredit training program).

Data Analyst

- Business rules and policies are not universal; different universities may have different policies for student advising and may include different types of people as students.
- Also, the rules and policies of an organization may change (usually slowly) over time; a university may decide that a student does not have to be assigned a faculty adviser until the student chooses a major.

- Your job as a database analyst is to
  - Identify and understand those rules that govern data
  - Represent those rules so that they can be unambiguously understood by information systems developers and users
  - Implement those rules in database technology

- Data models cannot represent all business rules (and do not need to, because not all business rules govern data); data models along with associated documentation and other types of information system models (e.g., models that document the processing of data) represent all business rules that must be enforced through information systems.

Overview of Business Rules

A **business rule** is "a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business . . . rules prevent, cause, or suggest things to happen" (GUIDE Business Rules Project, 1997). For example, the following two statements are common expressions of business rules that affect data processing and storage:

- "A student may register for a section of a course only if he or she has successfully completed the prerequisites for that course."
- "A preferred customer qualifies for a 10 percent discount, unless he has an overdue account balance."

Most organizations (and their employees) today are guided by thousands of combinations of such rules. In the aggregate, these rules influence behavior and determine how the organization responds to its environment (Gottesdiener, 1997; von Halle, 1997). Capturing and documenting business rules is an important, complex task. Thoroughly capturing and structuring business rules, then enforcing them through database technologies, helps to ensure that information systems work right and that users of the information understand what they enter and see.

TABLE 2-1 Characteristics of a Good Business Rule

Characteristic	Explanation
Declarative	A business rule is a statement of policy, not how policy is enforced or conducted; the rule does not describe a process or implementation, but rather describes what a process validates.
Precise	With the related organization, the rule must have only one interpretation among all interested people, and its meaning must be clear.
Atomic	A business rule marks one statement, not several; no part of the rule can stand on its own as a rule (that is, the rule is indivisible, yet sufficient).
Consistent	A business rule must be internally consistent (that is, not contain conflicting statements) and must be consistent with (and not contradict) other rules.
Expressible	A business rule must be able to be stated in natural language, but it will be stated in a structured natural language so that there is no misinterpretation.
Distinct	Business rules are not redundant, but a business rule may refer to other rules (especially to definitions).
Business-oriented	A business rule is stated in terms businesspeople can understand, and because it is a statement of business policy, only businesspeople can modify or invalidate a rule; thus, a business rule is owned by the business.

Source: Based on Gottesdiener (1999) and Plotkin (1999).

DATA NAMES We will provide specific guidelines for naming entities, relationships, and attributes as we develop the entity-relationship data model, but there are some general guidelines about naming any data object. Data names should (Salin, 1990; ISO/IEC, 2005)

- *Relate to business, not technical (hardware or software), characteristics;* so, Customer is a good name, but File10, Bit7, and Payroll Report Sort Key are not good names.
- *Be meaningful,* almost to the point of being self-documenting (i.e., the definition will refine and explain the name without having to state the essence of the object's meaning); you should avoid using generic words such as *has*, *is*, *person*, or *it*.
- *Be unique* from the name used for every other distinct data object; words should be included in a data name if they distinguish the data object from other similar data objects (e.g., Home Address versus Campus Address).
- *Be readable,* so that the name is structured as the concept would most naturally be said (e.g., Grade Point Average is a good name, whereas Average Grade Relative To A, although possibly accurate, is an awkward name).
- *Be composed of words taken from an approved list;* each organization often chooses a vocabulary from which significant words in data names must be chosen (e.g., maximum is preferred, never upper limit, ceiling, or highest); alternative, or alias names, also can be used as can approved abbreviations (e.g., CUST for CUSTOMER), and you may be encouraged to use the abbreviations so that data names are short enough to meet maximum length limits of database technology.
- *Be repeatable,* meaning that different people or the same person at different times should develop exactly or almost the same name; this often means that there is a standard hierarchy or pattern for names (e.g., the birth date of a student would be Student Birth Date and the birth date of an employee would be Employee Birth Date).
- *Follow a standard syntax,* meaning that the parts of the name should follow a standard arrangement adopted by the organization.

Modeling Entities and Attributes

- The basic constructs of the E-R model are entities, relationships, and attributes.
- As shown in the Basic ER Notations, the model allows numerous variations for each of these constructs.
- The richness of the E-R model allows designers to model real-world situations accurately and expressively, which helps account for the popularity of the model.

MODELING ENTITIES AND ATTRIBUTES

The basic constructs of the E-R model are entities, relationships, and attributes. As shown in Figure 2-2, the model allows numerous variations for each of these constructs. The richness of the E-R model allows designers to model real-world situations accurately and expressively, which helps account for the popularity of the model.

Entities

An **entity** is a person, a place, an object, an event, or a concept in the user environment about which the organization wishes to maintain data. Thus, an entity has a noun name. Some examples of each of these *kinds* of entities follow:

Person:	EMPLOYEE, STUDENT, PATIENT
Place:	STORE, WAREHOUSE, STATE
Object:	MACHINE, BUILDING, AUTOMOBILE
Event:	SALE, REGISTRATION, RENEWAL
Concept:	ACCOUNT, COURSE, WORK CENTER

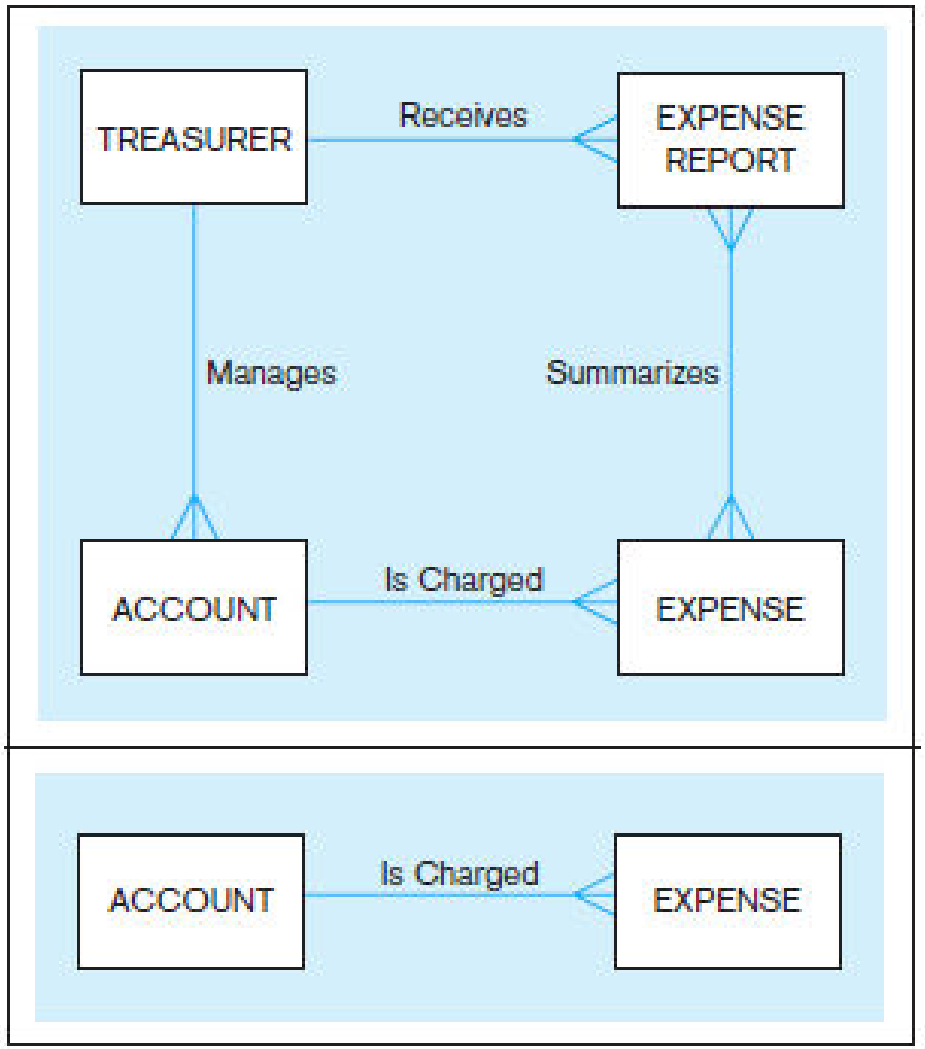
ENTITY TYPE VERSUS ENTITY INSTANCE There is an important distinction between entity types and entity instances. An **entity type** is a collection of entities that share common properties or characteristics. Each entity type in an E-R model is given a name. Because the name represents a collection (or set) of items, it is always singular. We use capital letters for names of entity type(s). In an E-R diagram, the entity name is placed inside the box representing the entity type (see Figure 2-1).

An **entity instance** is a single occurrence of an entity type. Figure 2-3 illustrates the distinction between an entity type and two of its instances. An entity type is described just once (using metadata) in a database, whereas many instances of that entity type may be represented by data stored in the database. For example, there is one EMPLOYEE entity type in most organizations, but there may be hundreds (or even thousands) of instances of this entity type stored in the database. We often use the single term *entity* rather than *entity instance* when the meaning is clear from the context of our discussion.

Entity type: EMPLOYEE

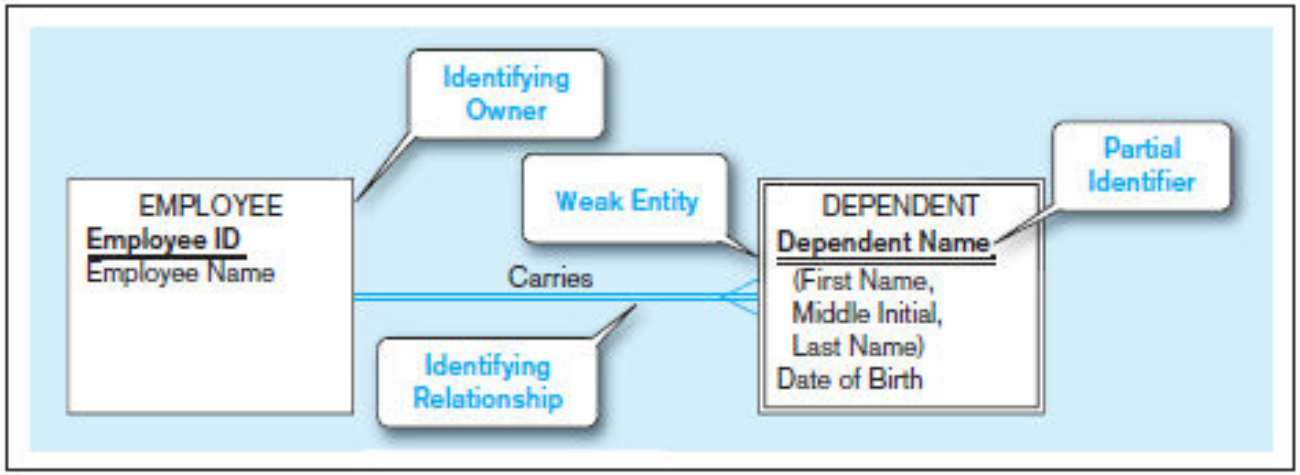
Attributes	**Attribute Data Type**	**Example Instance**	**Example Instance**
Employee Number	CHAR (10)	642-17-8360	534-10-1971
Name	CHAR (25)	Michelle Brady	David Johnson
Address	CHAR (30)	100 Pacific Avenue	450 Redwood Drive
City	CHAR (20)	San Francisco	Redwood City
State	CHAR (2)	CA	CA
Zip Code	CHAR (9)	98173	97142
Date Hired	DATE	03-21-1992	08-16-1994
Birth Date	DATE	06-19-1968	09-04-1975

STRONG VERSUS WEAK ENTITY TYPES Most of the basic entity types to identify in an organization are classified as strong entity types. A **strong entity type** is one that exists independently of other entity types. (Some data modeling software, in fact, use the term *independent entity*.) Examples include STUDENT, EMPLOYEE, AUTOMOBILE, and COURSE.



NAMING AND DEFINING ENTITY TYPES In addition to the general guidelines for naming and defining data objects, there are a few special guidelines for *naming* entity types, which follow:

- An entity type name is a *singular noun* (such as CUSTOMER, STUDENT, or AUTOMOBILE); an entity is a person, a place, an object, an event, or a concept, and the name is for the entity type, which represents a set of entity instances (i.e., STUDENT represents students Hank Finley, Jean Krebs, and so forth). It is common to also specify the plural form (possibly in a CASE tool repository accompanying the E-R diagram), because sometimes the E-R diagram is read best by using plurals. For example, in Figure 2-1, we would say that a SUPPLIER may supply ITEMs. Because plurals are not always formed by adding an *s* to the singular noun, it is best to document the exact plural form.
- An entity type name should be *specific to the organization*. Thus, one organization may use the entity type name CUSTOMER and another organization may use the entity type name CLIENT (this is one task, for example, done to customize a purchased data model). The name should be descriptive for everyone in the organization and distinct from all other entity type names within that organization. For example, a PURCHASE ORDER for orders placed with suppliers is distinct from CUSTOMER ORDER for orders placed with us by our customers. Both of these entity types cannot be named ORDER.



Attributes

Each entity type has a set of attributes associated with it. An **attribute** is a property or characteristic of an entity type that is of interest to the organization. (Later we will see that some types of relationships may also have attributes.) Thus, an attribute has a noun name. Following are some typical entity types and their associated attributes:

STUDENT	Student ID, Student Name, Home Address, Phone Number, Major
AUTOMOBILE	Vehicle ID, Color, Weight, Horsepower
EMPLOYEE	Employee ID, Employee Name, Payroll Address, Skill

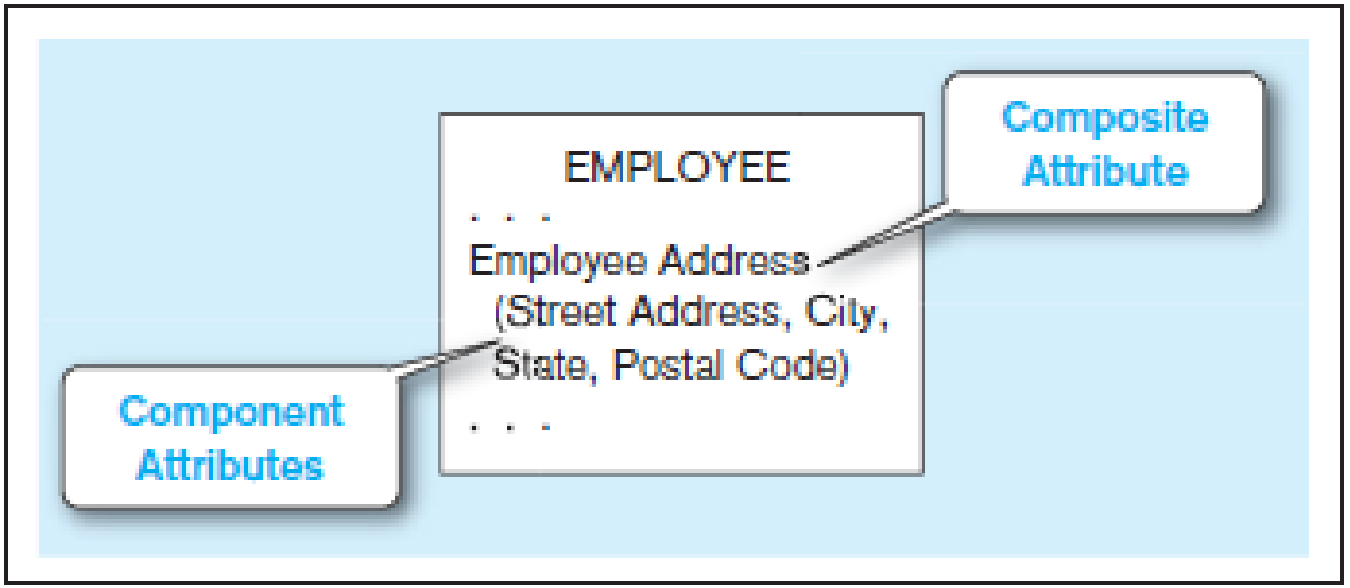
In naming attributes, we use an initial capital letter followed by lowercase letters. If an attribute name consists of more than one words, we use a space between the words and we start each word with a capital letter; for example Employee Name or Student Home Address. In E-R diagrams, we represent an attribute by placing its name in the entity it describes. Attributes may also be associated with relationships, as described later. Note that an attribute is associated with exactly one entity or relationship.

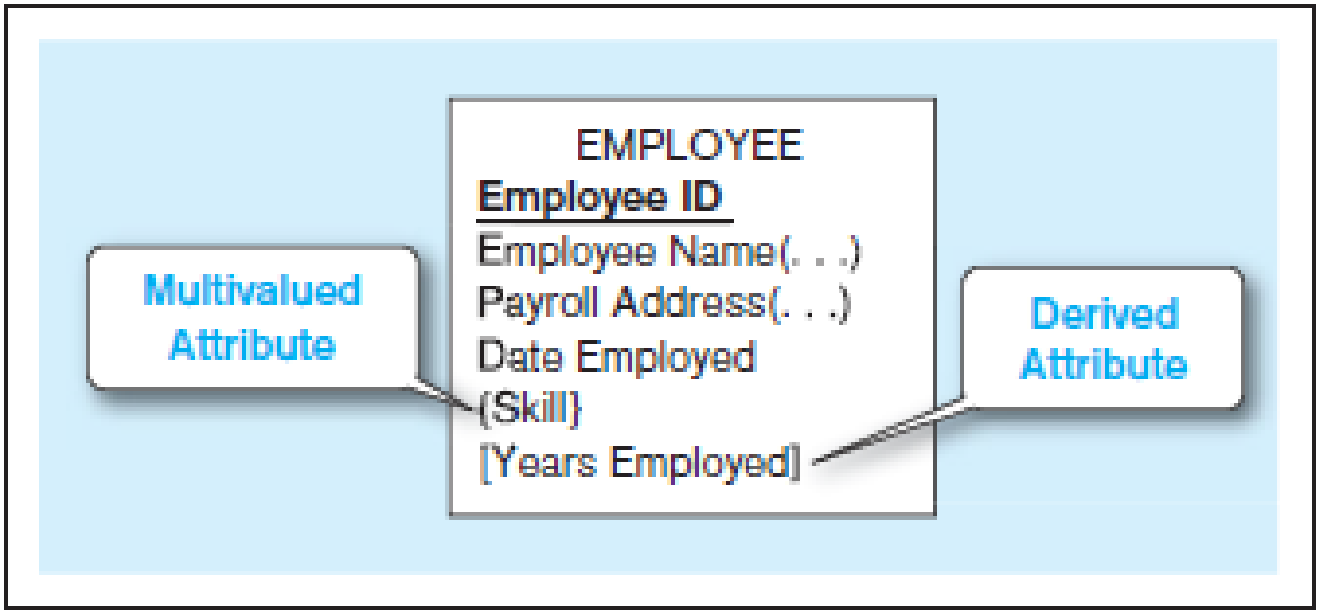
REQUIRED VERSUS OPTIONAL ATTRIBUTES Each entity (or instance of an entity type) potentially has a value associated with each of the attributes of that entity type. An attribute that must be present for each entity instance is called a **required attribute**, whereas an attribute that may not have a value is called an **optional attribute**. For example, Figure 2-6 shows two STUDENT entities (instances) with their respective

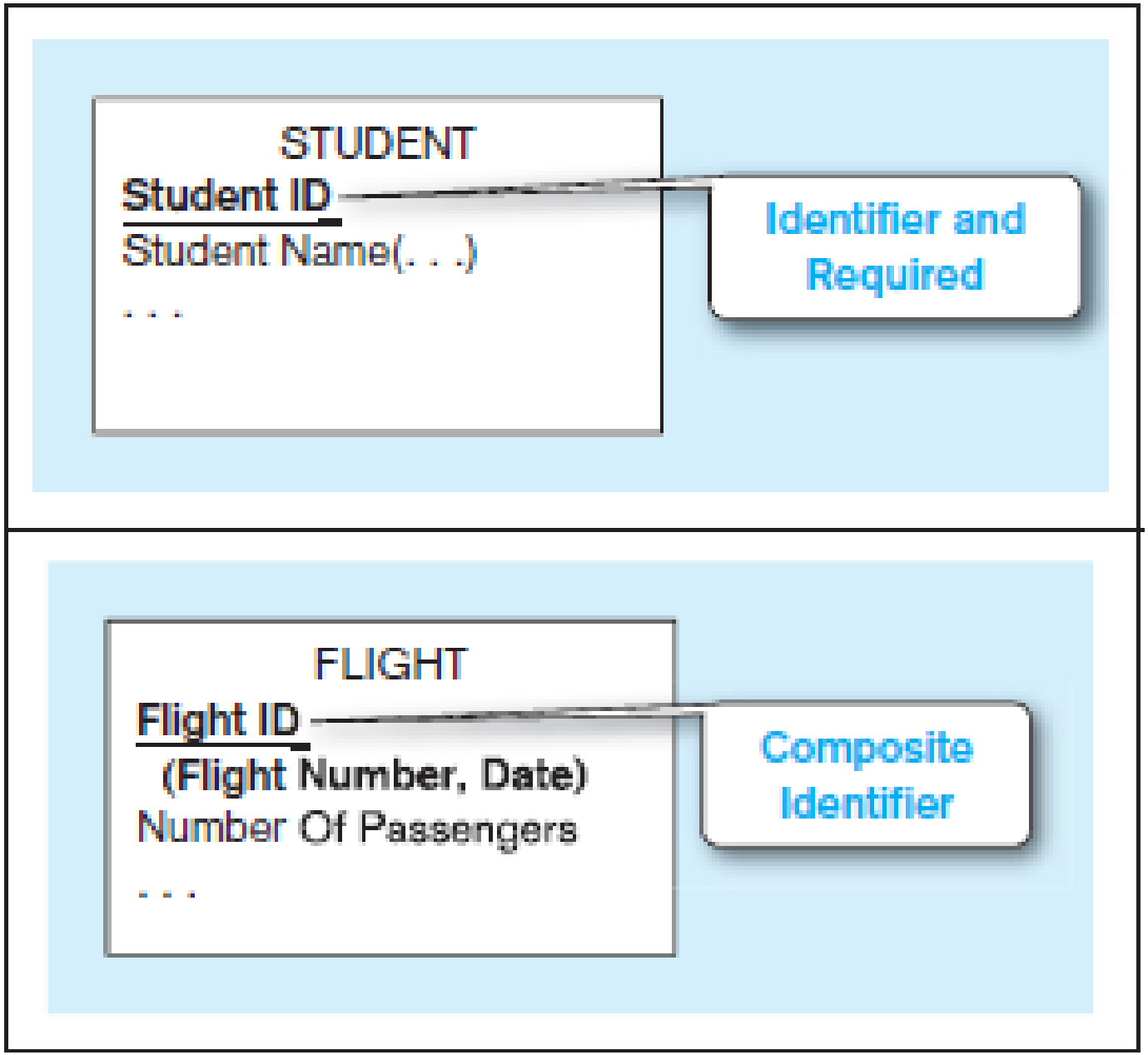
Entity type: STUDENT

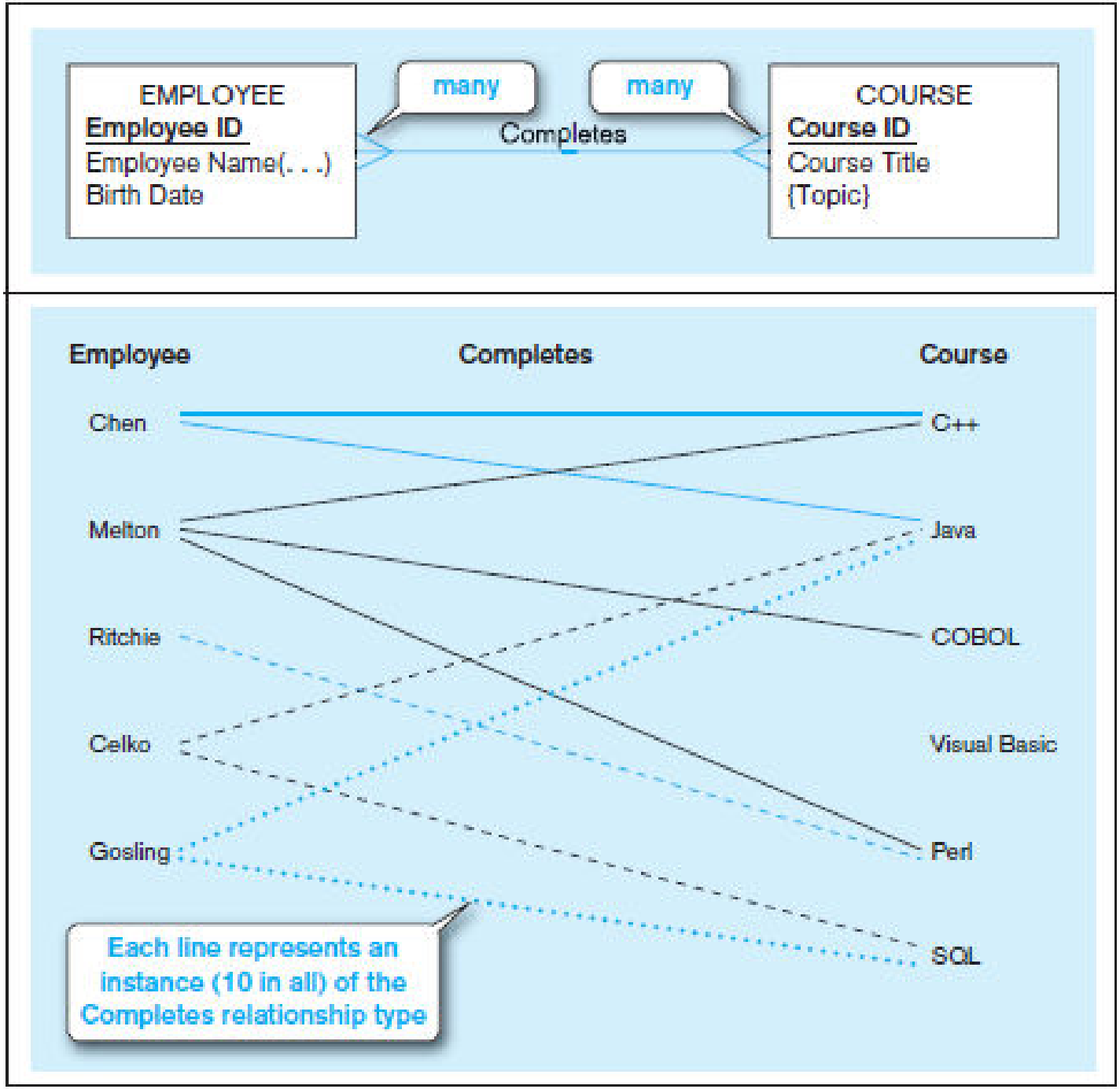
Attributes	Attribute Data Type	Required or Optional	Example Instance	Example Instance
Student ID	CHAR (10)	Required	876-24-8217	822-24-4456
Student Name	CHAR (40)	Required	Michael Grant	Melissa Kraft
Home Address	CHAR (30)	Required	314 Baker St.	1422 Heft Ave
Home City	CHAR (20)	Required	Centerville	Miami
Home State	CHAR (2)	Required	OH	FL
Home Zip Code	CHAR (9)	Required	45459	33321
Major	CHAR (3)	Optional	MIS	

SINGLE-VALUED VERSUS MULTIVALUED ATTRIBUTES Figure 2-6 shows two entity instances with their respective attribute values. For each entity instance, each of the attributes in the figure has one value. It frequently happens that there is an attribute that may have more than one value for a given instance. For example, the EMPLOYEE entity type in Figure 2-8 has an attribute named Skill, whose values record the skill (or skills) for that employee. Of course, some employees may have more than one skill,

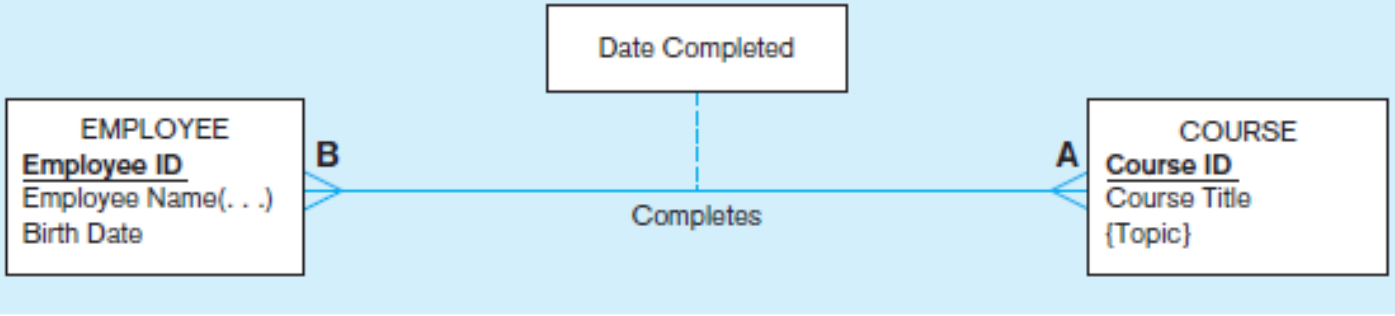



EMPLOYEE
Employee ID
Employee Name(. . .)
Payroll Address(. . .)
Date Employed
{Skill}
[Years Employed]
Multivalued Attribute
Derived Attribute


STUDENT
Student ID
Student Name(. . .)
. . .
Identifier and Required
FLIGHT
Flight ID
(Flight Number, Date)
Number Of Passengers
. . .
Composite Identifier

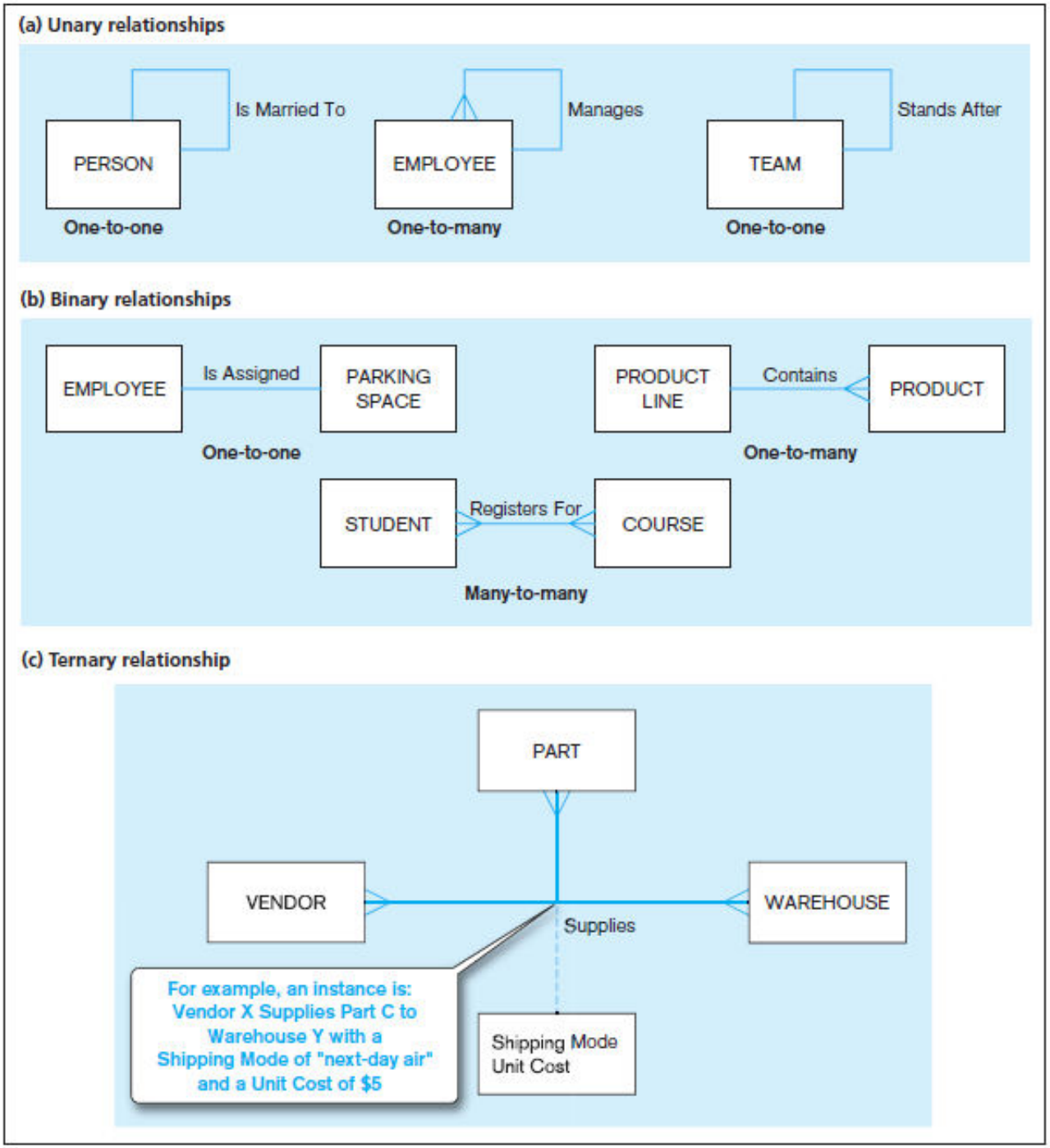

EMPLOYEE
Employee ID
Employee Name(. . .)
Birth Date
many
many
Completes
COURSE
Course ID
Course Title
{Topic}
Employee
Completes
Course
Chen
Melton
Ritchie
Celko
Gosling
C++
Java
COBOL
Visual Basic
Perl
SQL
Each line represents an instance (10 in all) of the Completes relationship type

(a) Attribute on a relationship



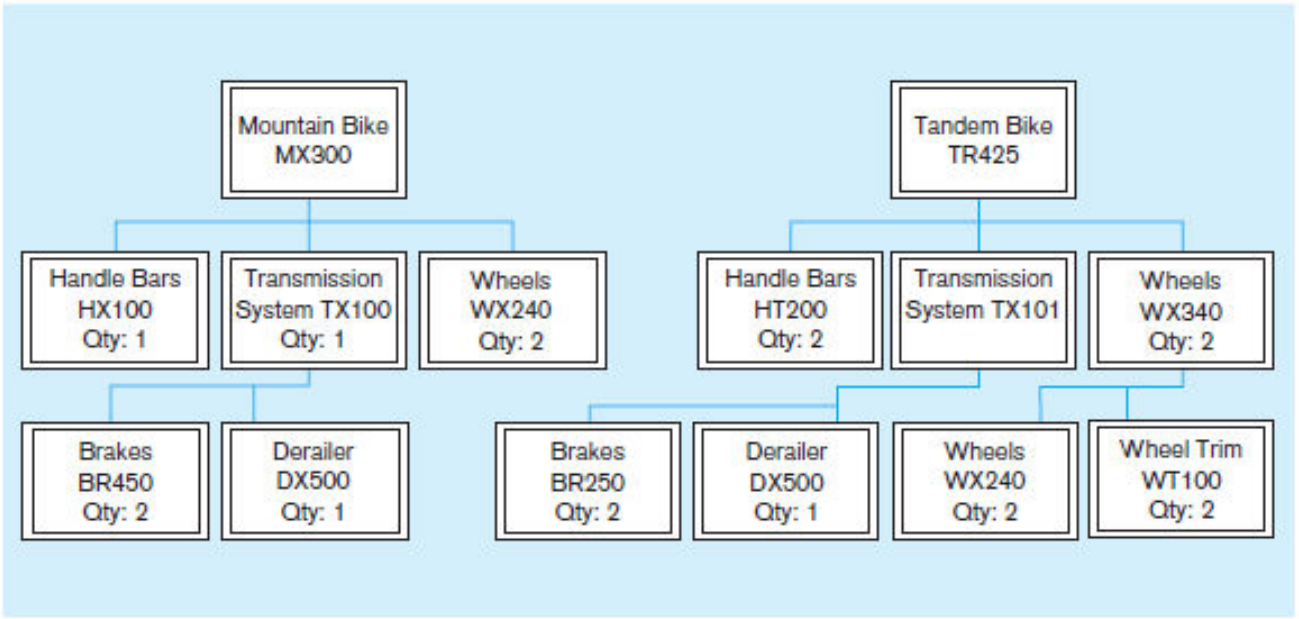
(b) An associative entity (CERTIFICATE)

(c) An associative entity using Microsoft VISIO


(a) Unary relationships
PERSON
Is Married To
One-to-one
EMPLOYEE
Manages
One-to-many
TEAM
Stands After
One-to-one
(b) Binary relationships
EMPLOYEE
Is Assigned
PARKING SPACE
One-to-one
PRODUCT LINE
Contains
PRODUCT
One-to-many
STUDENT
Registers For
COURSE
Many-to-many
(c) Ternary relationship
PART
VENDOR
WAREHOUSE
Supplies
Shipping Mode
Unit Cost
For example, an instance is:
Vendor X Supplies Part C to
Warehouse Y with a
Shipping Mode of "next-day air"
and a Unit Cost of $5

(a) Many-to-many relationship

(b) Two ITEM bill-of-materials structure instances



(c) Associative entity

FIGURE 2-14 **Ternary relationship as an associative entity**

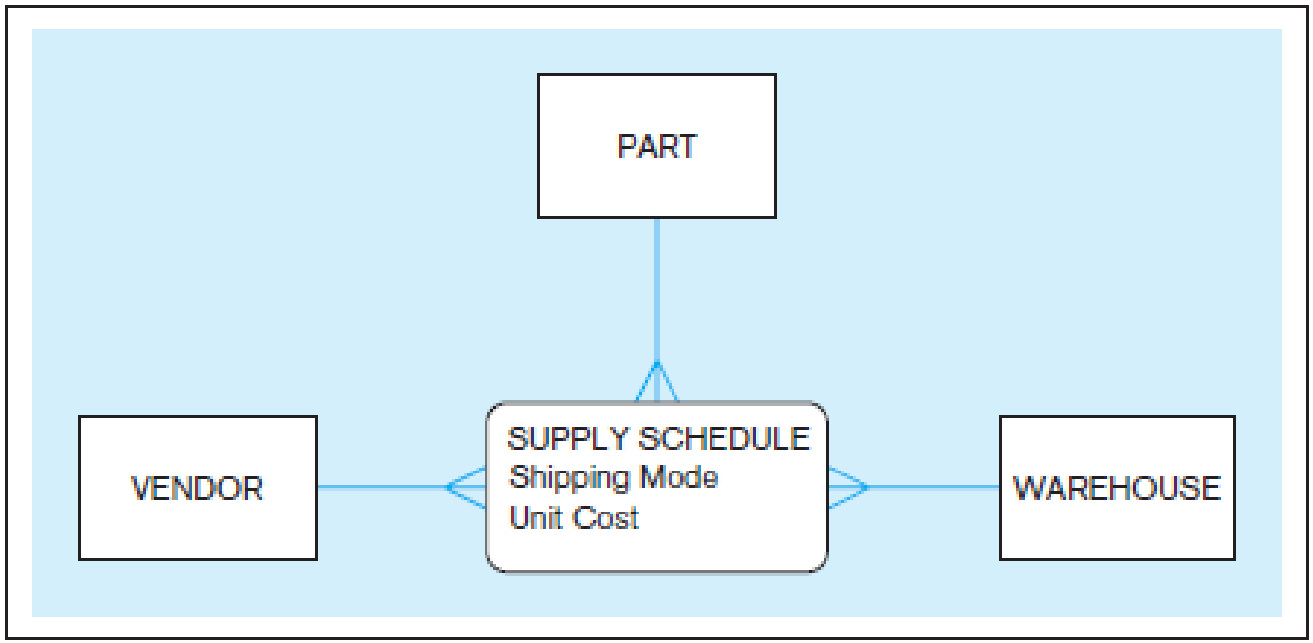
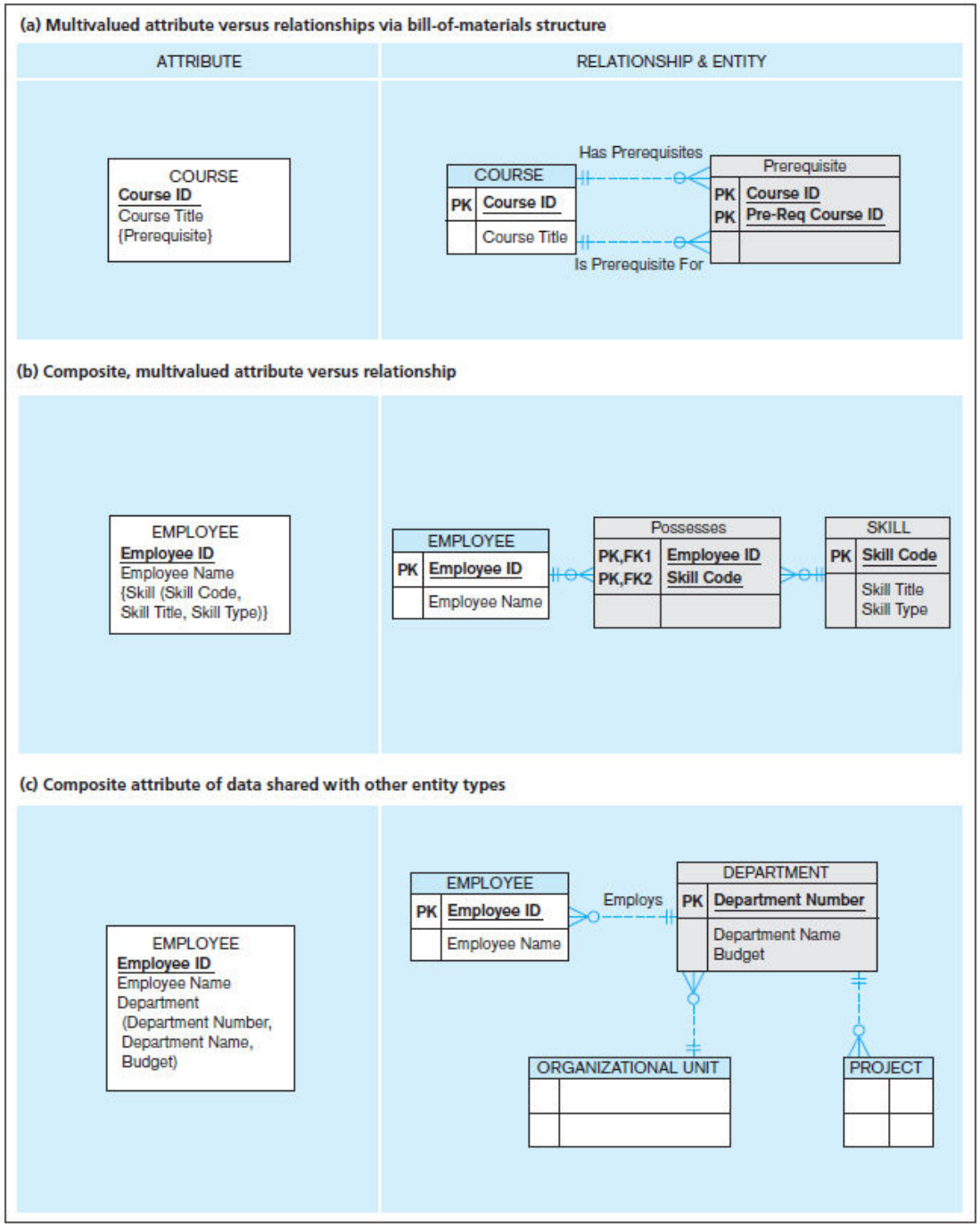


FIGURE 2-15 **Using relationships and entities to link related attributes**



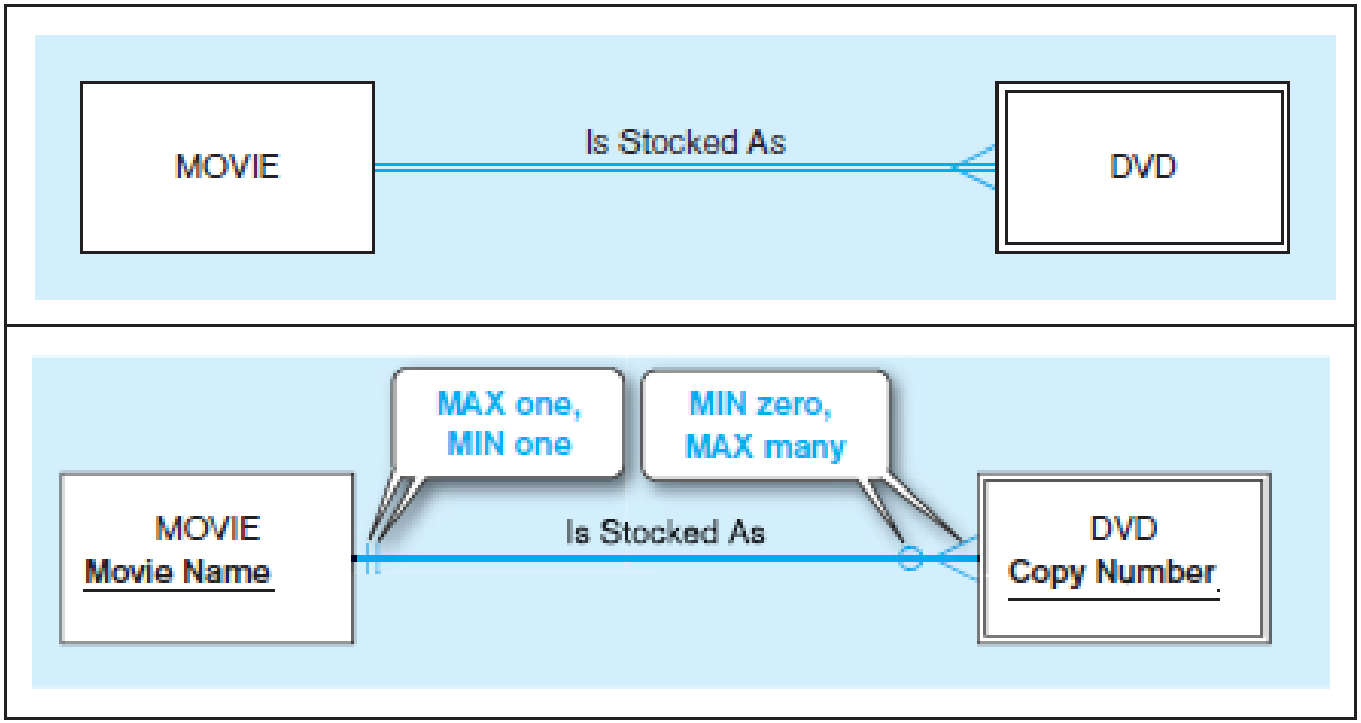
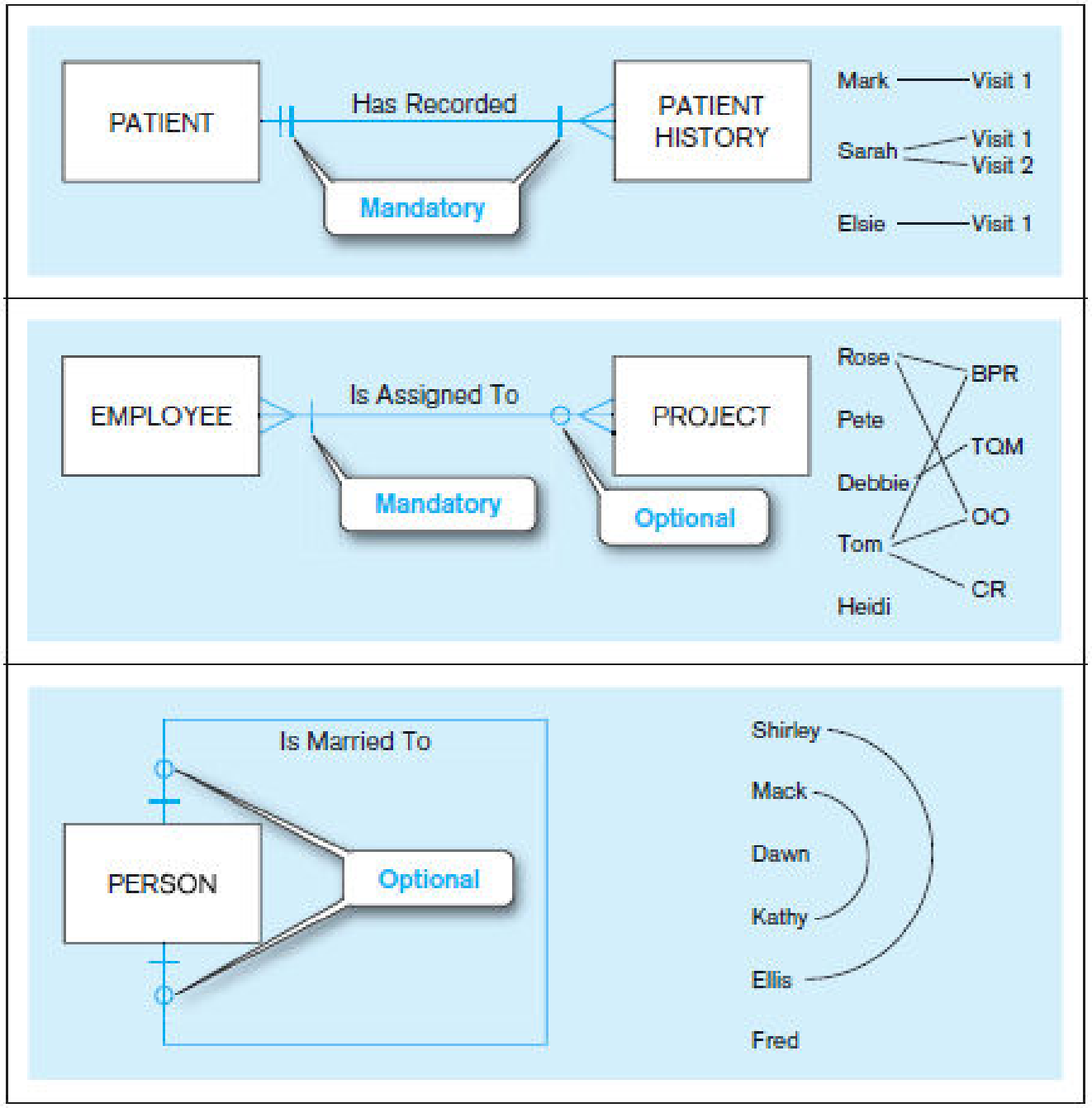


FIGURE 2-16 Introducing cardinality constraints
(a) Basic relationship

(b) Relationship with cardinality constraints

FIGURE 2-17 Examples of cardinality constraints
(a) Mandatory cardinalities



(b) One optional, one mandatory cardinality

(c) Optional cardinalities

FIGURE 2-18 **Cardinality constraints in a ternary relationship**