CHAPTER

3

The Enhanced E-R Model

Visit www.pearsonhighered.com/hoffer to view the accompanying video for this chapter.

Learning Objectives

After studying this chapter, you should be able to:

- Concisely define each of the following key terms: **enhanced entity-relationship (EER) model, subtype, supertype, attribute inheritance, generalization, specialization, completeness constraint, total specialization rule, partial specialization rule, disjointness constraint, disjoint rule, overlap rule, subtype discriminator, supertype/subtype hierarchy, entity cluster,** and **universal data model.**
- Recognize when to use supertype/subtype relationships in data modeling.
- Use both specialization and generalization as techniques for defining supertype/subtype relationships.
- Specify both completeness constraints and disjointness constraints in modeling supertype/subtype relationships.
- Develop a supertype/subtype hierarchy for a realistic business situation.
- Develop an entity cluster to simplify presentation of an E-R diagram.
- Explain the major features and data modeling structures of a universal (packaged) data model.
- Describe the special features of a data modeling project when using a packaged data model.

INTRODUCTION

The basic E-R model described in Chapter 2 was first introduced during the mid-1970s. It has been suitable for modeling most common business problems and has enjoyed widespread use. However, the business environment has changed dramatically since that time. Business relationships are more complex, and as a result, business data are much more complex as well. For example, organizations must be prepared to segment their markets and to customize their products, which places much greater demands on organizational databases.

To cope better with these changes, researchers and consultants have continued to enhance the E-R model so that it can more accurately represent the complex data encountered in today's business environment. The term **enhanced entity-relationship (EER) model** is used to identify the model that has resulted from extending the original E-R model with these new modeling constructs. These extensions make the EER model semantically similar to object-oriented data modeling, which we cover in Chapter 13.

Enhanced entity-relationship (EER) model
A model that has resulted from extending the original E-R model with new modeling constructs.

The most important modeling construct incorporated in the EER model is supertype/subtype relationships. This facility enables us to model a general entity type (called the *supertype*) and then subdivide it into several specialized entity types (called *subtypes*). Thus, for example, the entity type CAR can be modeled as a supertype, with subtypes SEDAN, SPORTS CAR, COUPE, and so on. Each subtype inherits attributes from its supertype and in addition may have special attributes and be involved in relationships of its own. Adding new notation for modeling supertype/subtype relationships has greatly improved the flexibility of the basic E-R model.

E-R, and especially EER, diagrams can become large and complex, requiring multiple pages (or very small font) for display. Some commercial databases include hundreds of entities. Many users and managers specifying requirements for or using a database do not need to see all the entities, relationships, and attributes to understand the part of the database with which they are most interested. Entity clustering is a way to turn a part of an entity-relationship data model into a more macro-level view of the same data. Entity clustering is a hierarchical decomposition technique (a nesting process of breaking a system into further and further subparts), which can make E-R diagrams easier to read and databases easier to design. By grouping entities and relationships, you can lay out an E-R diagram in such a way that you to give attention to the details of the model that matter most in a given data modeling task.

As introduced in Chapter 2, universal and industry-specific generalizable data models, which extensively utilized EER capabilities, have become very important for contemporary data modelers. These packaged data models and data model patterns have made data modelers more efficient and produce data models of higher quality. The EER features of supertypes/subtypes are essential to create generalizable data models; additional generalizing constructs, such as typing entities and relationships, are also employed. It has become very important for data modelers to know how to customize a data model pattern or data model for a major software package (e.g., enterprise resource planning or customer relationship management), just as it has become commonplace for information system builders to customize off-the-shelf software packages and software components.

REPRESENTING SUPERTYPES AND SUBTYPES

Recall from Chapter 2 that an entity type is a collection of entities that share common properties or characteristics. Although the entity instances that compose an entity type are similar, we do not expect them to have exactly the same attributes. For example, recall required and optional attributes from Chapter 2. One of the major challenges in data modeling is to recognize and clearly represent entities that are almost the same; that is, entity types that share common properties but also have one or more distinct properties that are of interest to the organization.

Subtype
A subgrouping of the entities in an entity type that is meaningful to the organization and that shares common attributes or relationships distinct from other subgroupings.

Supertype
A generic entity type that has a relationship with one or more subtypes.

For this reason, the E-R model has been extended to include supertype/subtype relationships. A **subtype** is a subgrouping of the entities in an entity type that is meaningful to the organization. For example, STUDENT is an entity type in a university. Two subtypes of STUDENT are GRADUATE STUDENT and UNDERGRADUATE STUDENT. In this example, we refer to STUDENT as the supertype. A **supertype** is a generic entity type that has a relationship with one or more subtypes.

In the E-R diagramming we have done so far, supertypes and subtypes have been hidden. For example, consider again Figure 2-22, which is the E-R diagram (in Microsoft Visio) for Pine Valley Furniture Company. Notice that it is possible for a customer to not do business in any territory (i.e., no associated instances of the DOES BUSINESS IN associative entity). Why is this? One possible reason is that there are two types of customers—national account customers and regular customers—and only regular customers are assigned to a sales territory. Thus, in Figure 2-22, the reason for the optional cardinality next to the DOES BUSINESS IN associative entity coming from CUSTOMER is obscured. Explicitly drawing a customer entity supertype and several entity subtypes will help us to make the E-R diagram more meaningful. Later in this chapter, we show a revised E-R diagram for Pine Valley Furniture, which demonstrates several EER notations to make vague aspects of Figure 2-22 more explicit.

Basic Concepts and Notation

The notation that is used for supertype/subtype relationships in this text is shown in Figure 3-1a. The supertype is connected with a line to a circle, which in turn is connected with a line to each subtype that has been defined. The U-shaped symbol on each line connecting a subtype to the circle emphasizes that the subtype is a subset of the supertype. It also indicates the direction of the subtype/supertype relationship. (This U is optional because the meaning and direction of the supertype/subtype relationship is usually obvious; in most examples, we will not include this symbol). Figure 3-1b shows the type of EER notation used by Microsoft Visio (which is very similar to that used in this text), and Figure 3-1c shows the type of EER notation used by some CASE tools (e.g., Oracle Designer); the notation in Figure 3-1c is also the form often used for universal and industry-specific data models. These different formats have identical basic features, and you should easily become comfortable using any of these forms. We primarily use the text notation for examples in this chapter because advanced EER features are more standard with this format.

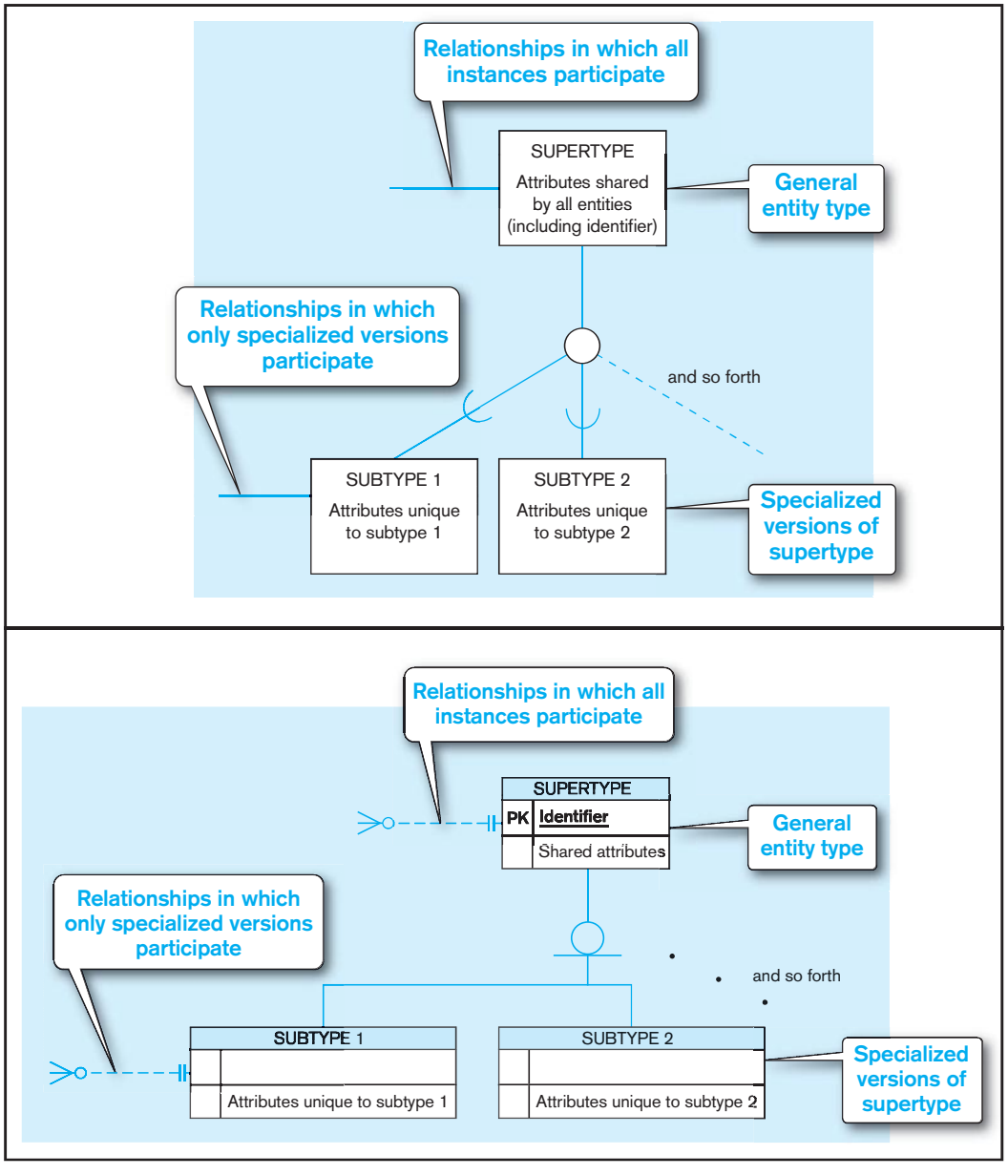


FIGURE 3-1 Basic notation for supertype/subtype relationships

(a) EER notation

(b) Microsoft Visio notation

(continued)

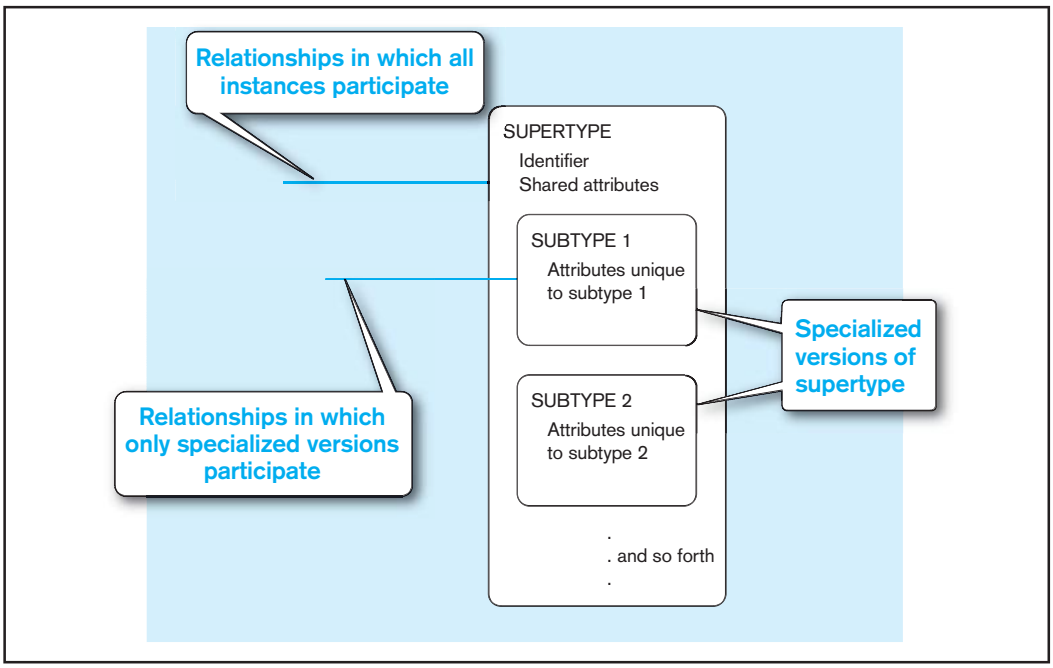


FIGURE 3-1 (*continued*)
(c) Subtypes inside supertypes notation

Attributes that are shared by all entities (including the identifier) are associated with the supertype. Attributes that are unique to a particular subtype are associated with that subtype. The same is true for relationships. Other components will be added to this notation to provide additional meaning in supertype/subtype relationships as we proceed through the remainder of this chapter.

AN EXAMPLE OF A SUPERTYPE/SUBTYPE RELATIONSHIP Let us illustrate supertype/subtype relationships with a simple yet common example. Suppose that an organization has three basic types of employees: hourly employees, salaried employees, and contract consultants. The following are some of the important attributes for each of these types of employees:

- ***Hourly employees*** Employee Number, Employee Name, Address, Date Hired, Hourly Rate
- ***Salaried employees*** Employee Number, Employee Name, Address, Date Hired, Annual Salary, Stock Option
- ***Contract consultants*** Employee Number, Employee Name, Address, Date Hired, Contract Number, Billing Rate

Notice that all of the employee types have several attributes in common: Employee Number, Employee Name, Address, and Date Hired. In addition, each type has one or more attributes distinct from the attributes of other types (e.g., Hourly Rate is unique to hourly employees). If you were developing a conceptual data model in this situation, you might consider three choices:

1. Define a single entity type called EMPLOYEE. Although conceptually simple, this approach has the disadvantage that EMPLOYEE would have to contain all of the attributes for the three types of employees. For an instance of an hourly employee (for example), attributes such as Annual Salary and Contract Number would not apply (optional attributes) and would be null or not used. When taken to a development environment, programs that use this entity type would necessarily need to be quite complex to deal with the many variations.
2. Define a separate entity type for each of the three entities. This approach would fail to exploit the common properties of employees, and users would have to be careful to select the correct entity type when using the system.
3. Define a supertype called EMPLOYEE with subtypes HOURLY EMPLOYEE, SALARIED EMPLOYEE, and CONSULTANT. This approach exploits the common properties of all employees, yet recognizes the distinct properties of each type.

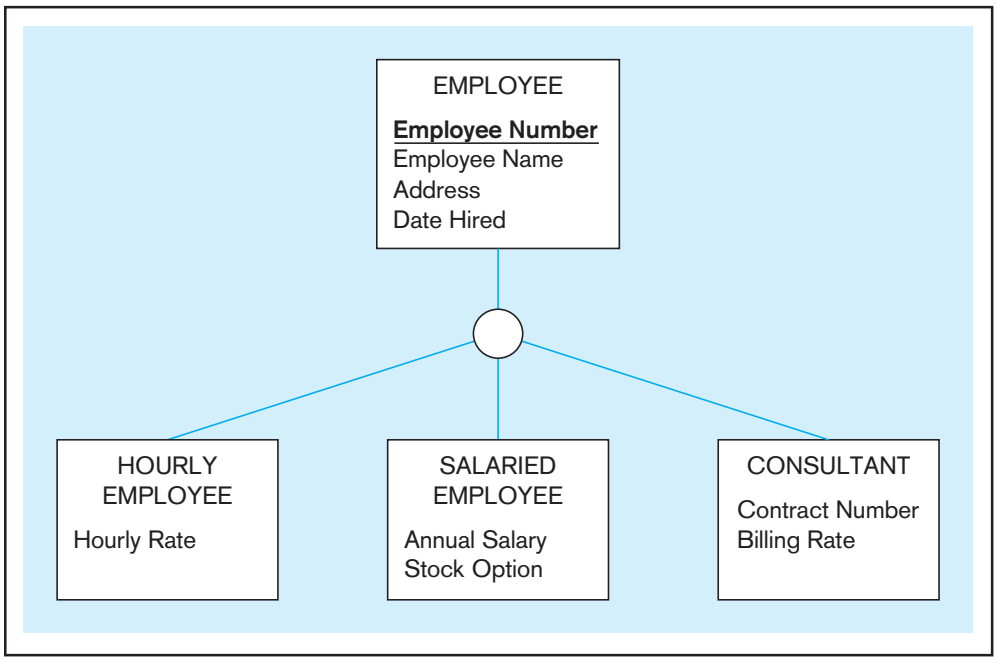


FIGURE 3-2 Employee supertype with three subtypes

Figure 3-2 shows a representation of the EMPLOYEE supertype with its three subtypes, using enhanced E-R notation. Attributes shared by all employees are associated with the EMPLOYEE entity type. Attributes that are peculiar to each subtype are included with that subtype only.

ATTRIBUTE INHERITANCE A subtype is an entity type in its own right. An entity instance of a subtype represents the same entity instance of the supertype. For example, if "Therese Jones" is an occurrence of the CONSULTANT subtype, then this same person is necessarily an occurrence of the EMPLOYEE supertype. As a consequence, an entity in a subtype must possess not only values for its own attributes, but also values for its attributes as a member of the supertype, including the identifier.

Attribute inheritance is the property by which subtype entities inherit values of all attributes and instance of all relationships of the supertype. This important property makes it unnecessary to include supertype attributes or relationships redundantly with the subtypes (remember, when it comes to data modeling, redundancy = bad, simplicity = good). For example, Employee Name is an attribute of EMPLOYEE (Figure 3-2) but not of the subtypes of EMPLOYEE. Thus, the fact that the employee's name is "Therese Jones" is inherited from the EMPLOYEE supertype. However, the Billing Rate for this same employee is an attribute of the subtype CONSULTANT.

Attribute inheritance
A property by which subtype entities inherit values of all attributes and instances of all relationships of their supertype.

We have established that a member of a subtype must be a member of the supertype. Is the converse also true—that is, is a member of the supertype also a member of one (or more) of the subtypes? This may or may not be true, depending on the business situation. (Sure, "it depends" is the classic academic answer, but it's true in this case.) We discuss the various possibilities later in this chapter.

WHEN TO USE SUPERTYPE/SUBTYPE RELATIONSHIPS So, how do you know when to use a supertype/subtype relationship? You should consider using subtypes when either (or both) of the following conditions are present:

1. There are attributes that apply to some (but not all) instances of an entity type. For example, see the EMPLOYEE entity type in Figure 3-2.
2. The instances of a subtype participate in a relationship unique to that subtype.

Figure 3-3 is an example of the use of subtype relationships that illustrates both of these situations. The hospital entity type PATIENT has two subtypes: OUTPATIENT and RESIDENT PATIENT. (The identifier is Patient ID.) All patients have an Admit

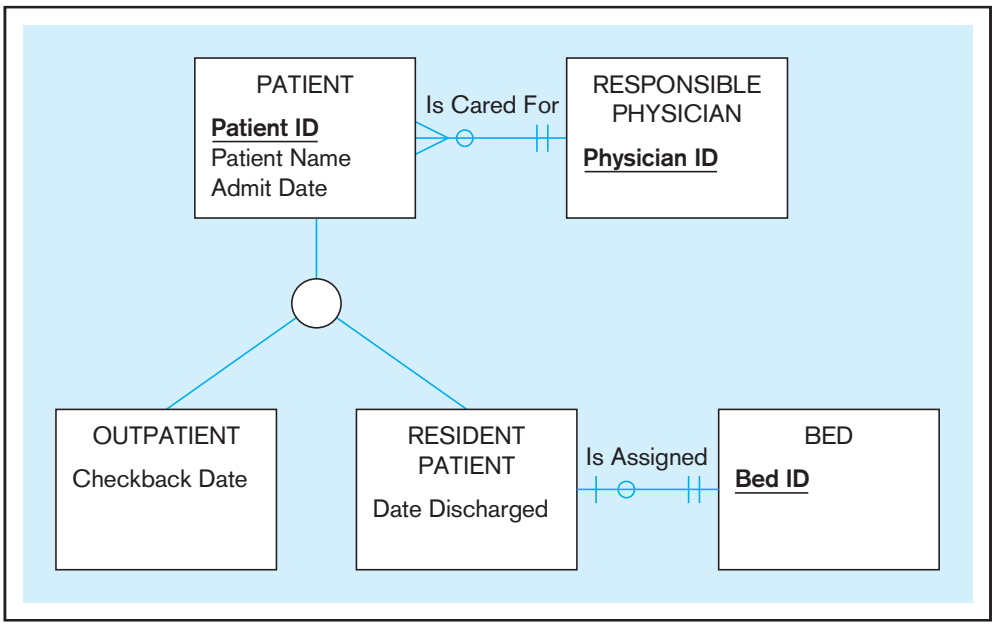


FIGURE 3-3 Supertype/subtype relationships in a hospital

Date attribute, as well as a Patient Name. Also, every patient is cared for by a RESPONSIBLE PHYSICIAN who develops a treatment plan for the patient.

Each subtype has an attribute that is unique to that subtype. Outpatients have Checkback Date, whereas resident patients have Date Discharged. Also, resident patients have a unique relationship that assigns each patient to a bed. (Notice that this is a mandatory relationship; it would be optional if it were attached to PATIENT.) Each bed may or may not be assigned to a patient.

Earlier we discussed the property of attribute inheritance. Thus, each outpatient and each resident patient inherits the attributes of the parent supertype PATIENT: Patient ID, Patient Name, and Admit Date. Figure 3-3 also illustrates the principle of relationship inheritance. OUTPATIENT and RESIDENT PATIENT are also instances of PATIENT; therefore, each Is Cared For by a RESPONSIBLE PHYSICIAN.

Representing Specialization and Generalization

We have described and illustrated the basic principles of supertype/subtype relationships, including the characteristics of "good" subtypes. But in developing real-world data models, how can you recognize opportunities to exploit these relationships? There are two processes—generalization and specialization—that serve as mental models in developing supertype/subtype relationships.

GENERALIZATION A unique aspect of human intelligence is the ability and propensity to classify objects and experiences and to generalize their properties. In data modeling, **generalization** is the process of defining a more general entity type from a set of more specialized entity types. Thus generalization is a bottom-up process.

Generalization
The process of defining a more general entity type from a set of more specialized entity types.

An example of generalization is shown in Figure 3-4. In Figure 3-4a, three entity types have been defined: CAR, TRUCK, and MOTORCYCLE. At this stage, the data modeler intends to represent these separately on an E-R diagram. However, on closer examination, we see that the three entity types have a number of attributes in common: Vehicle ID (identifier), Vehicle Name (with components Make and Model), Price, and Engine Displacement. This fact (reinforced by the presence of a common identifier) suggests that each of the three entity types is really a version of a more general entity type.

This more general entity type (named VEHICLE) together with the resulting supertype/subtype relationships is shown in Figure 3-4b. The entity CAR has the specific attribute No Of Passengers, whereas TRUCK has two specific attributes: Capacity and Cab Type. Thus, generalization has allowed us to group entity types along with

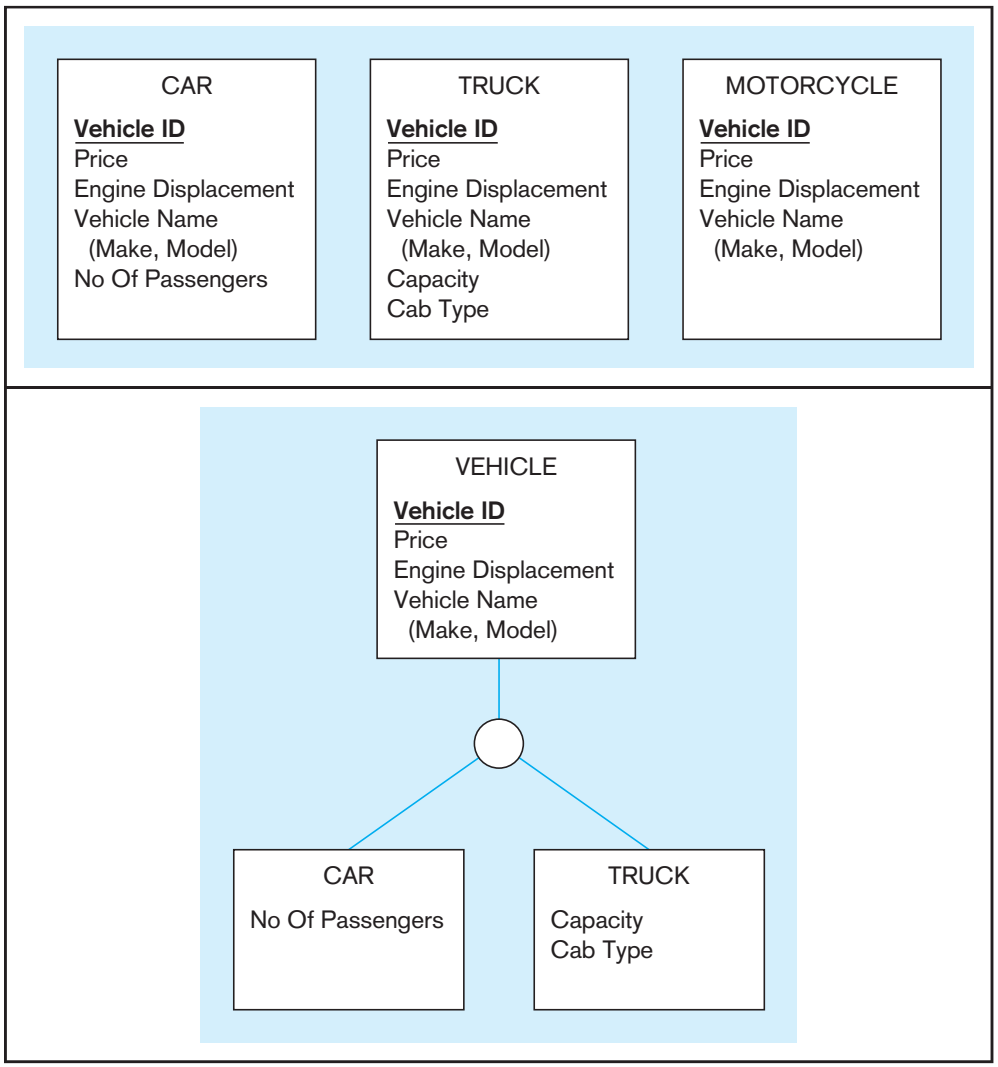


FIGURE 3-4 **Example of generalization**
(a) Three entity types: CAR, TRUCK, and MOTORCYCLE

(b) Generalization to VEHICLE supertype

their common attributes and at the same time preserve specific attributes that are peculiar to each subtype.

Notice that the entity type MOTORCYCLE is not included in the relationship. Is this simply an omission? No. Instead, it is deliberately not included because it does not satisfy the conditions for a subtype discussed earlier. Comparing Figure 3-4 parts a and b, you will notice that the only attributes of MOTORCYCLE are those that are common to all vehicles; there are no attributes specific to motorcycles. Furthermore, MOTORCYCLE does not have a relationship to another entity type. Thus there is no need to create a MOTORCYCLE subtype.

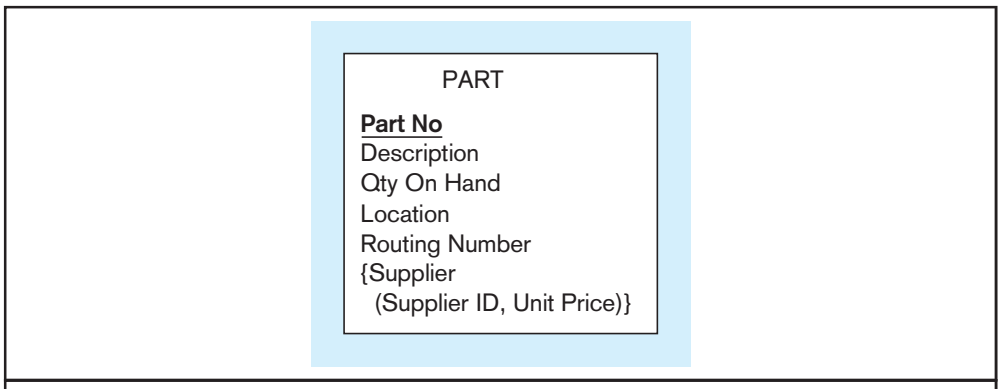
The fact that there is no MOTORCYCLE subtype suggests that it must be possible to have an instance of VEHICLE that is not a member of any of its subtypes. We discuss this type of constraint in the section on specifying constraints.

SPECIALIZATION As we have seen, generalization is a bottom-up process. **Specialization** is a top-down process, the direct reverse of generalization. Suppose that we have defined an entity type with its attributes. Specialization is the process of defining one or more subtypes of the supertype and forming supertype/subtype relationships. Each subtype is formed based on some distinguishing characteristic, such as attributes or relationships specific to the subtype.

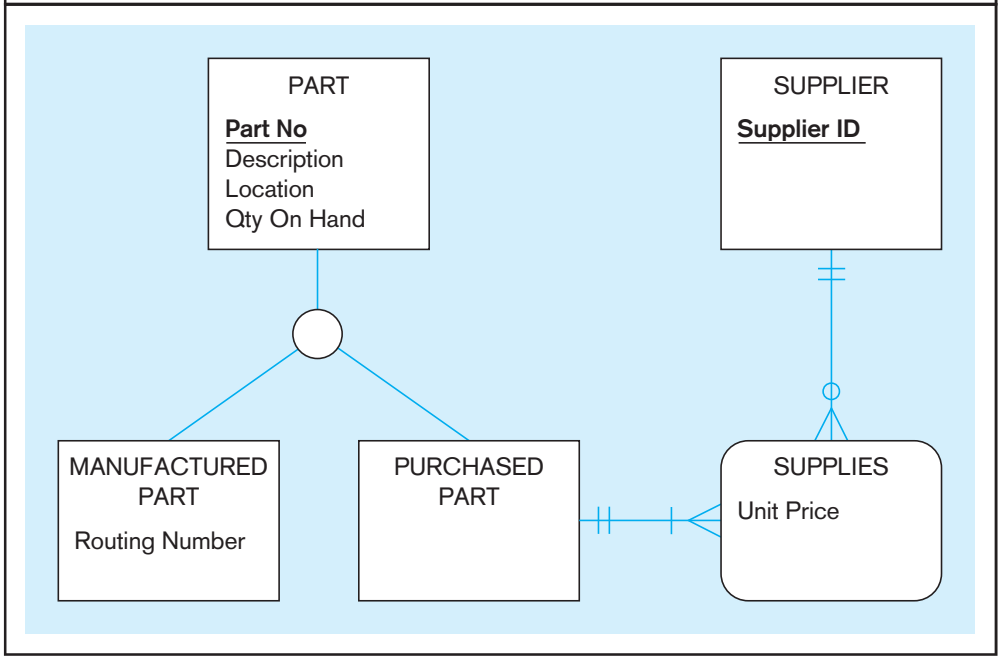
Specialization
The process of defining one or more subtypes of the supertype and forming supertype/subtype relationships.

An example of specialization is shown in Figure 3-5. Figure 3-5a shows an entity type named PART, together with several of its attributes. The identifier is Part No, and other attributes are Description, Unit Price, Location, Qty On Hand, Routing Number,

FIGURE 3-5 Example of specialization
(a) Entity type PART



(b) Specialization to MANUFACTURED PART and PURCHASED PART



and Supplier. (The last attribute is multivalued and composite because there may be more than one supplier with an associated unit price for a part.)

In discussions with users, we discover that there are two possible sources for parts: Some are manufactured internally, whereas others are purchased from outside suppliers. Further, we discover that some parts are obtained from both sources. In this case, the choice depends on factors such as manufacturing capacity, unit price of the parts, and so on.

Some of the attributes in Figure 3-5a apply to all parts, regardless of source. However, others depend on the source. Thus Routing Number applies only to manufactured parts, whereas Supplier ID and Unit Price apply only to purchased parts. These factors suggest that PART should be specialized by defining the subtypes MANUFACTURED PART and PURCHASED PART (Figure 3-5b).

In Figure 3-5b, Routing Number is associated with MANUFACTURED PART. The data modeler initially planned to associate Supplier ID and Unit Price with PURCHASED PART. However, in further discussions with users, the data modeler suggested instead that they create a SUPPLIER entity type and an associative entity linking PURCHASED PART with SUPPLIER. This associative entity (named SUPPLIES in Figure 3-5b) allows users to more easily associate purchased parts with their suppliers. Notice that the attribute Unit Price is now associated with the associative entity, so that the unit price for a part may vary from one supplier to another. In this example, specialization has permitted a preferred representation of the problem domain.

COMBINING SPECIALIZATION AND GENERALIZATION Specialization and generalization are both valuable techniques for developing supertype/subtype relationships. The technique you use at a particular time depends on several factors, such as the nature

of the problem domain, previous modeling efforts, and personal preference. You should be prepared to use both approaches and to alternate back and forth as dictated by the preceding factors.

SPECIFYING CONSTRAINTS IN SUPERTYPE/SUBTYPE RELATIONSHIPS

So far we have discussed the basic concepts of supertype/subtype relationships and introduced some basic notation to represent these concepts. We have also described the processes of generalization and specialization, which help a data modeler recognize opportunities for exploiting these relationships. In this section, we introduce additional notation to represent constraints on supertype/subtype relationships. These constraints allow us to capture some of the important business rules that apply to these relationships. The two most important types of constraints that are described in this section are completeness and disjointness constraints (Elmasri and Navathe, 1994).

Specifying Completeness Constraints

A **completeness constraint** addresses the question of whether an instance of a supertype must also be a member of at least one subtype. The completeness constraint has two possible rules: total specialization and partial specialization. The **total specialization rule** specifies that each entity instance of the supertype must be a member of some subtype in the relationship. The **partial specialization rule** specifies that an entity instance of the supertype is allowed not to belong to any subtype. We illustrate each of these rules with earlier examples from this chapter (see Figure 3-6).

Completeness constraint
A type of constraint that addresses whether an instance of a supertype must also be a member of at least one subtype.

Total specialization rule
A rule that specifies that each entity instance of a supertype must be a member of some subtype in the relationship.

Partial specialization rule
A rule that specifies that an entity instance of a supertype is allowed not to belong to any subtype.

TOTAL SPECIALIZATION RULE Figure 3-6a repeats the example of PATIENT (Figure 3-3) and introduces the notation for total specialization. In this example, the business rule is the following: A patient must be either an outpatient or a resident patient. (There are no other types of patient in this hospital.) Total specialization is indicated by the *double* line extending from the PATIENT entity type to the circle. (In the Microsoft Visio notation, total specialization is called "Category is complete" and is shown also by a *double* line under the category circle between the supertype and associated subtypes.)

In this example, every time a new instance of PATIENT is inserted into the supertype, a corresponding instance is inserted into either OUTPATIENT or RESIDENT PATIENT. If the instance is inserted into RESIDENT PATIENT, an instance of the relationship Is Assigned is created to assign the patient to a hospital bed.

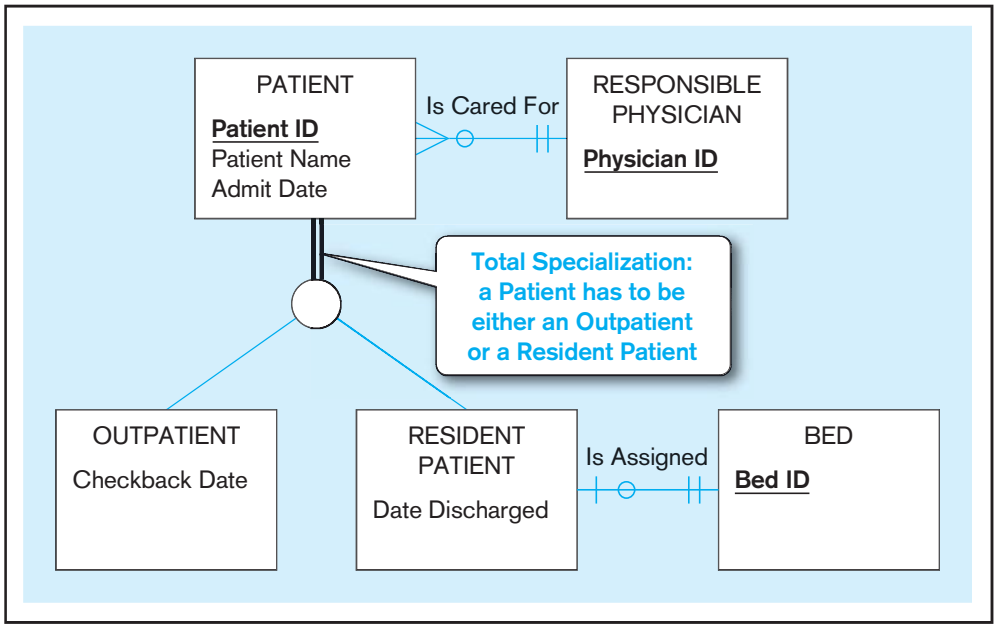
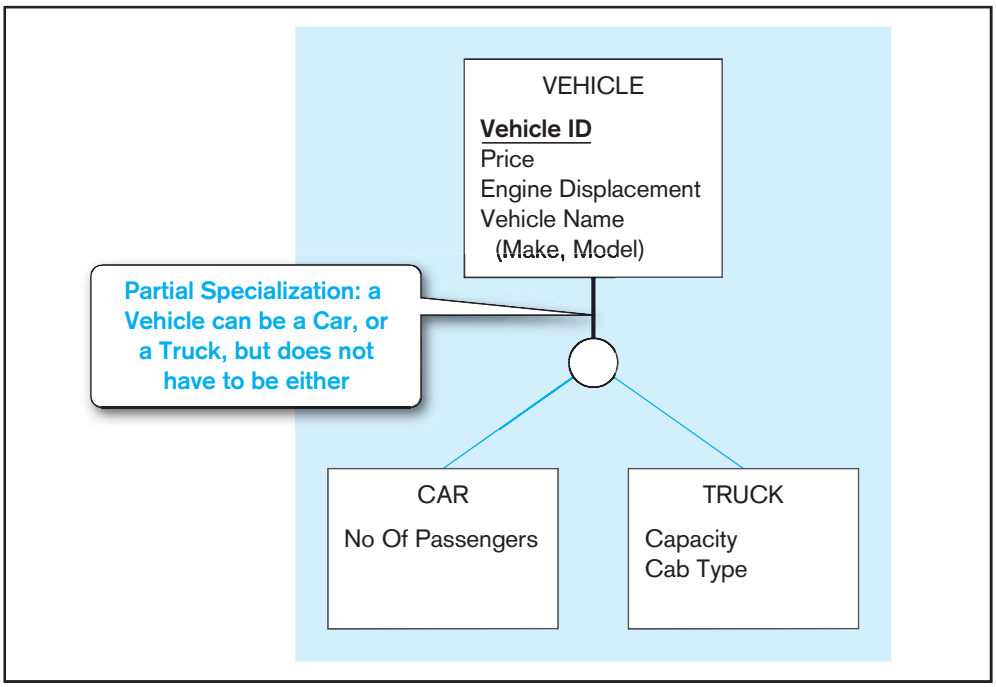


FIGURE 3-6 Examples of completeness constraints (a) Total specialization rule

(continued)

FIGURE 3-6 (*continued*)
(b) Partial specialization rule



PARTIAL SPECIALIZATION RULE Figure 3-6b repeats the example of VEHICLE and its subtypes CAR and TRUCK from Figure 3-4. Recall that in this example, motorcycle is a type of vehicle, but it is not represented as a subtype in the data model. Thus, if a vehicle is a car, it must appear as an instance of CAR, and if it is a truck, it must appear as an instance of TRUCK. However, if the vehicle is a motorcycle, it cannot appear as an instance of any subtype. This is an example of partial specialization, and it is specified by the single line from the VEHICLE supertype to the circle.

Specifying Disjointness Constraints

Disjointness constraint
A constraint that addresses whether an instance of a supertype may simultaneously be a member of two (or more) subtypes.

A **disjointness constraint** addresses whether an instance of a supertype may simultaneously be a member of two (or more) subtypes. The disjointness constraint has two possible rules: the disjoint rule and the overlap rule. The 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. The overlap rule specifies that an entity instance can simultaneously be a member of two (or more) subtypes. An example of each of these rules is shown in Figure 3-7.

Disjoint rule
A rule that specifies that an instance of a supertype may not simultaneously be a member of two (or more) subtypes.

DISJOINT RULE Figure 3-7a shows the PATIENT example from Figure 3-6a. The business rule in this case is the following: *At any given time*, a patient must be either an outpatient or a resident patient, but cannot be both. This is the **disjoint rule**, as specified by the letter *d* in the circle joining the supertype and its subtypes. Note in this figure, the subclass of a PATIENT may change over time, but at a given time, a PATIENT is of only one type. (The Microsoft Visio notation does not have a way to designate disjointness or overlap; however, you can place a *d* or an *o* inside the category circle using the Text tool.)

OVERLAP RULE Figure 3-7b shows the entity type PART with its two subtypes, MANUFACTURED PART and PURCHASED PART (from Figure 3-5b). Recall from our discussion of this example that some parts are both manufactured and purchased. Some clarification of this statement is required. In this example, an instance of PART is a particular part number (i.e., a *type of part*), not an individual part (indicated by the identifier, which is Part No). For example, consider part number 4000. At a given time, the quantity on hand for this part might be 250, of which 100 are manufactured and the remaining 150 are purchased parts. In this case, it is not important to keep track of individual

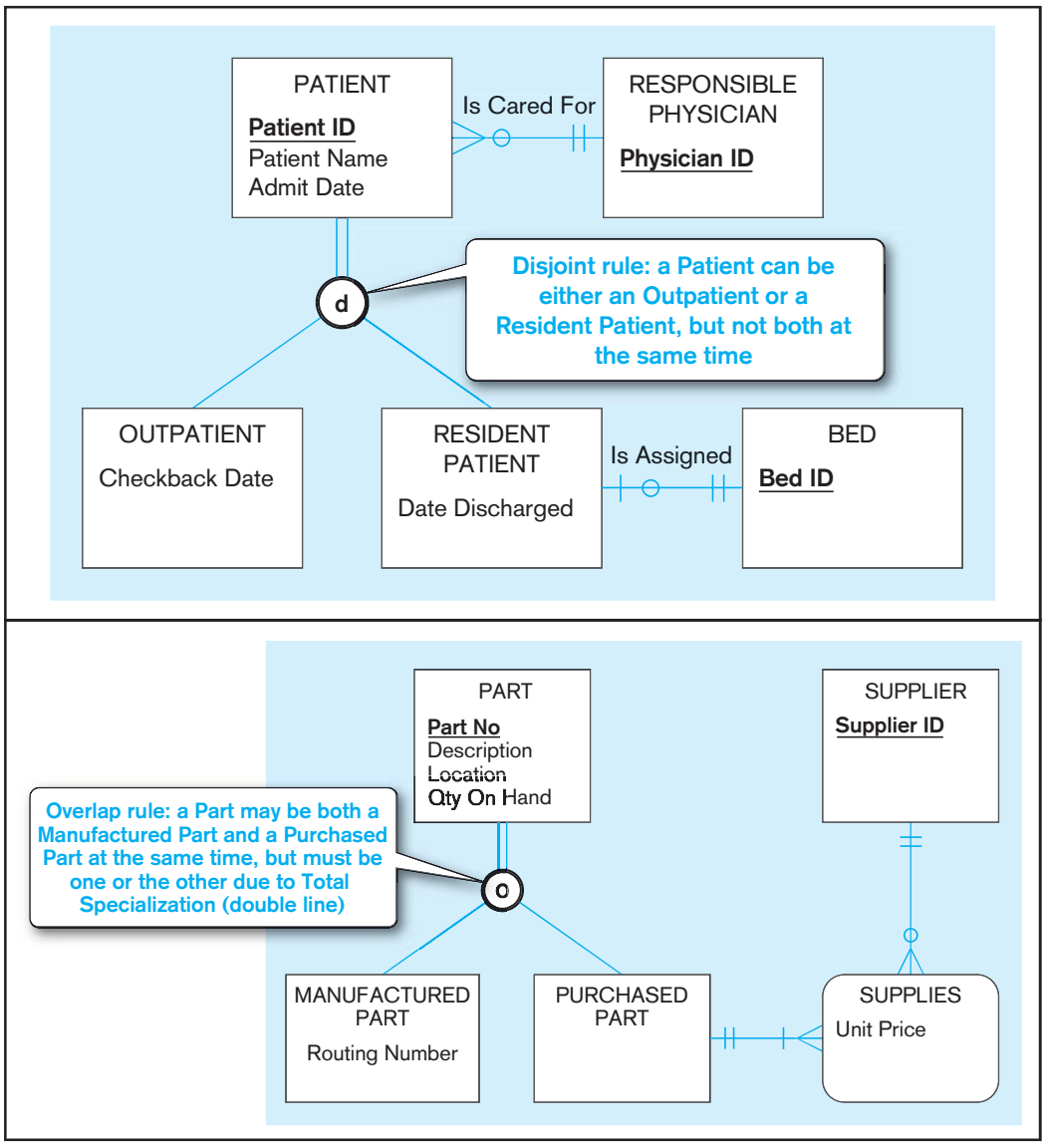


FIGURE 3-7 **Examples of disjointness constraints**
(a) Disjoint rule

(b) Overlap rule

parts. When tracking individual parts is important, each part is assigned a serial number identifier, and the quantity on hand is one or zero, depending on whether that individual part exists or not.

The **overlap rule** is specified by placing the letter *o* in the circle, as shown in Figure 3-7b. Notice in this figure that the total specialization rule is also specified, as indicated by the double line. Thus, any part must be either a purchased part or a manufactured part, or it may simultaneously be both of these.

Overlap rule
A rule that specifies that an instance of a supertype may simultaneously be a member of two (or more) subtypes.

Defining Subtype Discriminators

Given a supertype/subtype relationship, consider the problem of inserting a new instance of a supertype. Into which of the subtypes (if any) should this instance be inserted? We have already discussed the various possible rules that apply to this situation. We need a simple mechanism to implement these rules, if one is available. Often this can be accomplished by using a subtype discriminator. A **subtype discriminator** is an attribute of a supertype whose values determine the target subtype or subtypes.

Subtype discriminator
An attribute of a supertype whose values determine the target subtype or subtypes.

DISJOINT SUBTYPES An example of the use of a subtype discriminator is shown in Figure 3-8. This example is for the EMPLOYEE supertype and its subtypes, introduced in Figure 3-2. Notice that the following constraints have been added to this

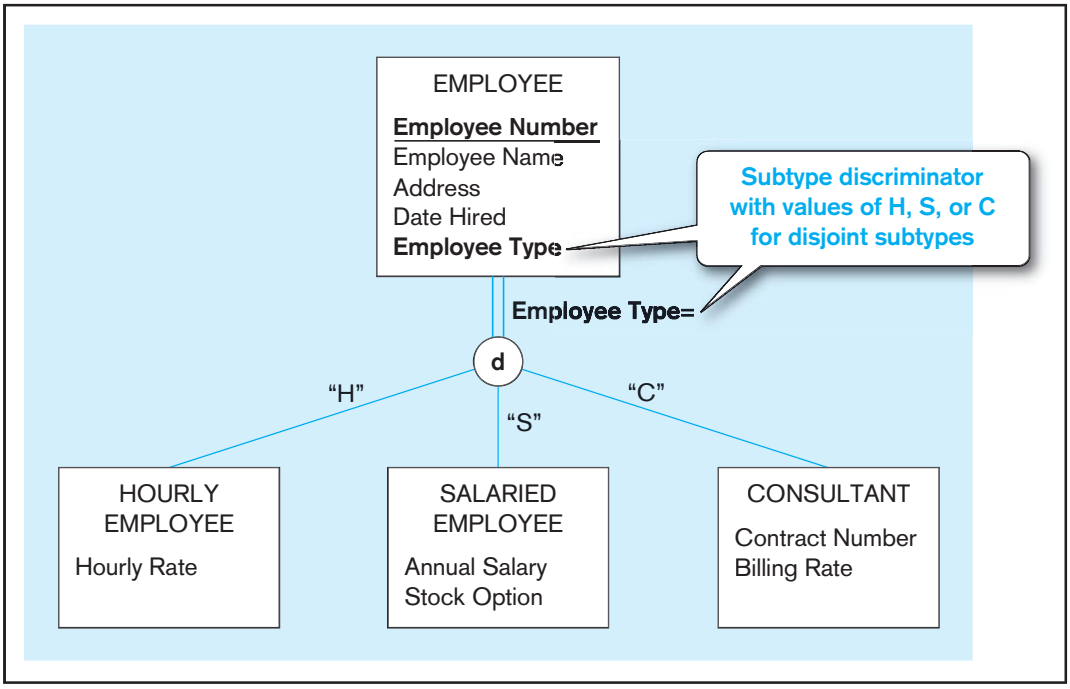


FIGURE 3-8 **Introducing a subtype discriminator (disjoint rule)**

figure: total specialization and disjoint subtypes. Thus, each employee must be either hourly, salaried, or a consultant.

A new attribute (Employee Type) has been added to the supertype, to serve as a subtype discriminator. When a new employee is added to the supertype, this attribute is coded with one of three values, as follows: "H" (for Hourly), "S" (for Salaried), or "C" (for Consultant). Depending on this code, the instance is then assigned to the appropriate subtype. (An attribute of the supertype may be selected in the Microsoft Visio notation as a discriminator, which is shown similarly next to the category symbol.)

The notation we use to specify the subtype discriminator is also shown in Figure 3-8. The expression Employee Type= (which is the left side of a condition statement) is placed next to the line leading from the supertype to the circle. The value of the attribute that selects the appropriate subtype (in this example, either "H," "S," or "C") is placed adjacent to the line leading to that subtype. Thus, for example, the condition Employee Type="S" causes an entity instance to be inserted into the SALARIED EMPLOYEE subtype.

OVERLAPPING SUBTYPES When subtypes overlap, a slightly modified approach must be applied for the subtype discriminator. The reason is that a given instance of the supertype may require that we create an instance in more than one subtype.

An example of this situation is shown in Figure 2-9 for PART and its overlapping subtypes. A new attribute named Part Type has been added to PART. Part Type is a composite attribute with components Manufactured? and Purchased? Each of these attributes is a Boolean variable (i.e., it takes on only the values yes, "Y," and no, "N"). When a new instance is added to PART, these components are coded as follows:

Type of Part	Manufactured?	Purchased?
Manufactured only	"Y"	"N"
Purchased only	"N"	"Y"
Purchased and manufactured	"Y"	"Y"

The method for specifying the subtype discriminator for this example is shown in Figure 3-9. Notice that this approach can be used for any number of overlapping subtypes.

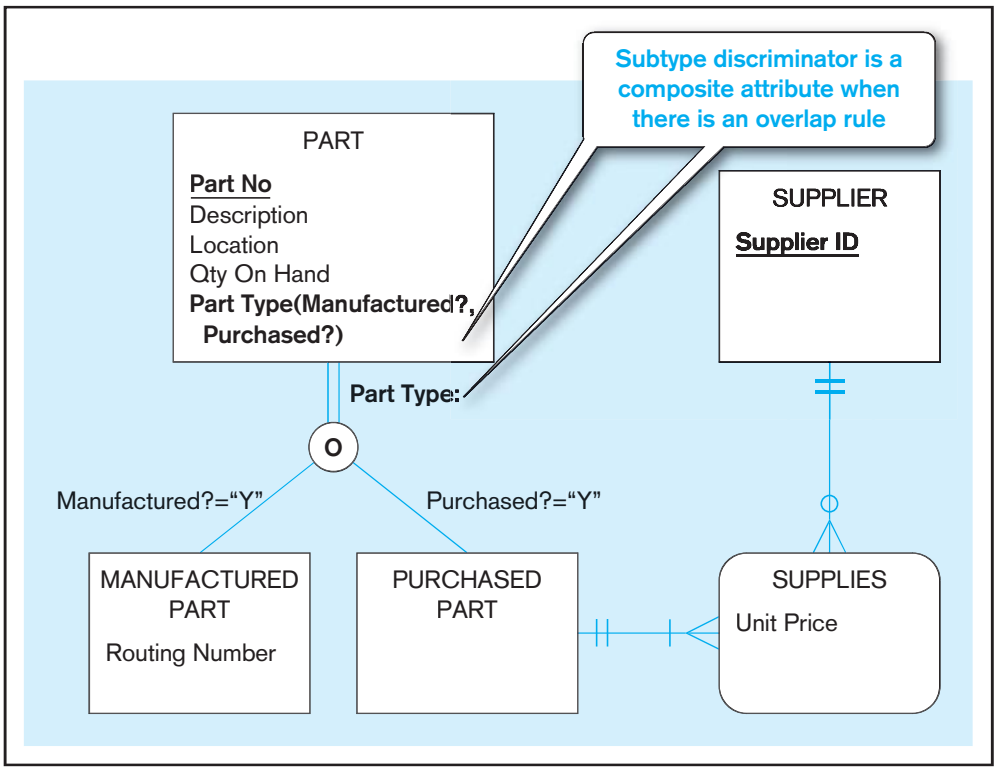


FIGURE 3-9 Subtype discriminator (overlap rule)

Defining Supertype/Subtype Hierarchies

We have considered a number of examples of supertype/subtype relationships in this chapter. It is possible for any of the subtypes in these examples to have other subtypes defined on it (in which case, the subtype becomes a supertype for the newly defined subtypes). A **supertype/subtype hierarchy** is a hierarchical arrangement of supertypes and subtypes, where each subtype has only one supertype (Elmasri and Navathe, 1994).

Supertype/subtype hierarchy A hierarchical arrangement of supertypes and subtypes in which each subtype has only one supertype.

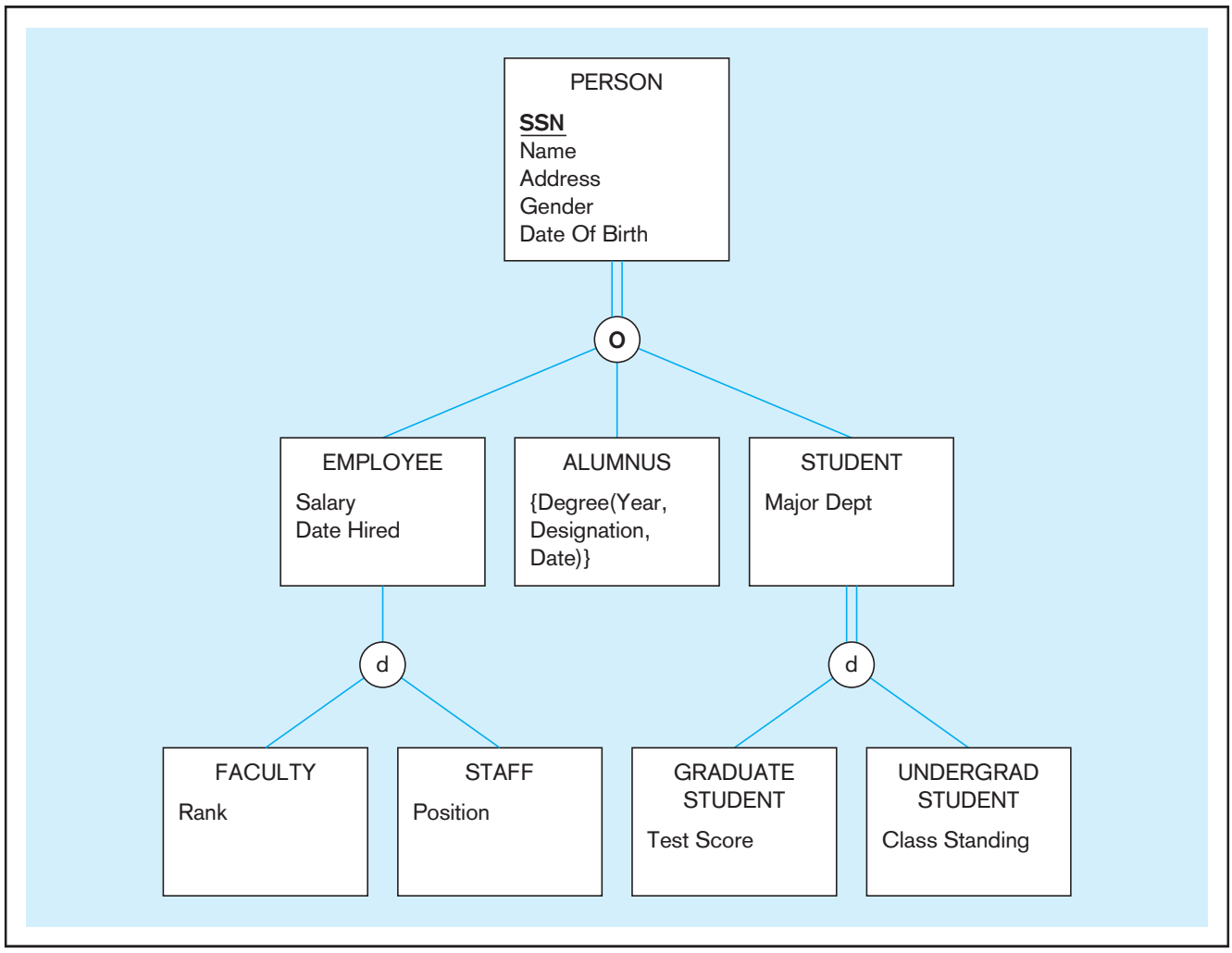
We present an example of a supertype/subtype hierarchy in this section in Figure 3-10. (For simplicity, we do not show subtype discriminators in this and most subsequent examples. See Problem and Exercise 2 and 3.) This example includes most of the concepts and notation we have used in this chapter to this point. It also presents a methodology (based on specialization) that you can use in many data modeling situations.

AN EXAMPLE OF A SUPERTYPE/SUBTYPE HIERARCHY Suppose that you are asked to model the human resources in a university. Using specialization (a top-down approach), you might proceed as follows. Starting at the top of a hierarchy, model the most general entity type first. In this case, the most general entity type is PERSON. List and associate all attributes of PERSON. The attributes shown in Figure 3-10 are SSN (identifier), Name, Address, Gender, and Date Of Birth. The entity type at the top of a hierarchy is sometimes called the *root*.

Next, define all major subtypes of the root. In this example, there are three subtypes of PERSON: EMPLOYEE (persons who work for the university), STUDENT (persons who attend classes), and ALUMNUS (persons who have graduated). Assuming that there are no other types of persons of interest to the university, the total specialization rule applies, as shown in the figure. A person might belong to more than one subtype (e.g., ALUMNUS and EMPLOYEE), so the overlap rule is used. Note that overlap allows for any overlap. (A PERSON may be simultaneously in any pair or in all three subtypes.) If certain combinations are not allowed, a more refined supertype/subtype hierarchy would have to be developed to eliminate the prohibited combinations.

Attributes that apply specifically to each of these subtypes are shown in the figure. Thus, each instance of EMPLOYEE has a value for Date Hired and Salary. Major Dept

FIGURE 3-10 **Example of supertype/subtype hierarchy**



is an attribute of STUDENT, and Degree (with components Year, Designation, and Date) is a multivalued, composite attribute of ALUMNUS.

The next step is to evaluate whether any of the subtypes already defined qualify for further specialization. In this example, EMPLOYEE is partitioned into two subtypes: FACULTY and STAFF. FACULTY has the specific attribute Rank, whereas STAFF has the specific attribute Position. Notice that in this example the subtype EMPLOYEE becomes a supertype to FACULTY and STAFF. Because there may be types of employees other than faculty and staff (such as student assistants), the partial specialization rule is indicated. However, an employee cannot be both faculty and staff at the same time. Therefore, the disjoint rule is indicated in the circle.

Two subtypes are also defined for STUDENT: GRADUATE STUDENT and UNDERGRAD STUDENT. UNDERGRAD STUDENT has the attribute Class Standing, whereas GRADUATE STUDENT has the attribute Test Score. Notice that total specialization and the disjoint rule are specified; you should be able to state the business rules for these constraints.

SUMMARY OF SUPERTYPE/SUBTYPE HIERARCHIES We note two features concerning the attributes contained in the hierarchy shown in Figure 3-10:

1. Attributes are assigned at the highest logical level that is possible in the hierarchy. For example, because SSN (i.e., Social Security Number) applies to all persons, it is assigned to the root. In contrast, Date Hired applies only to employees, so it is assigned to EMPLOYEE. This approach ensures that attributes can be shared by as many subtypes as possible.

2. Subtypes that are lower in the hierarchy inherit attributes not only from their immediate supertype, but from all supertypes higher in the hierarchy, up to the root. Thus, for example, an instance of faculty has values for all of the following attributes: SSN, Name, Address, Gender, and Date Of Birth (from PERSON); Date Hired and Salary (from EMPLOYEE); and Rank (from FACULTY).

In the student case at the end of this chapter, we ask you to develop an enhanced E-R diagram for Mountain View Community Hospital using the same procedure we outlined in this section.

EER MODELING EXAMPLE: PINE VALLEY FURNITURE COMPANY

In Chapter 2, we presented a sample E-R diagram for Pine Valley Furniture. (This diagram, developed using Microsoft Visio, is repeated in Figure 3-11.) After studying this diagram, you might use some questions to help you clarify the meaning of entities and relationships. Three such areas of questions are (see annotations in Figure 3-11 that indicate the source of each question):

1. Why do some customers not do business in one or more sales territories?
2. Why do some employees not supervise other employees, and why are they not all supervised by another employee? And, why do some employees not work in a work center?
3. Why do some vendors not supply raw materials to Pine Valley Furniture?

You may have other questions, but we will concentrate on these three to illustrate how supertype/subtype relationships can be used to convey a more specific (semantically rich) data model.

After some investigation into these three questions, we discover the following business rules that apply to how Pine Valley Furniture does business:

1. There are two types of customers: regular and national account. Only regular customers do business in sales territories. A sales territory exists only if it has at least one regular customer associated with it. A national account customer is associated with an account manager. It is possible for a customer to be both a regular and a national account customer.
2. Two special types of employees exist: management and union. Only union employees work in work centers, and a management employee supervises union employees. There are other kinds of employees besides management and union. A union employee may be promoted into management, at which time that employee stops being a union employee.
3. Pine Valley Furniture keeps track of many different vendors, not all of which have ever supplied raw materials to the company. A vendor is associated with a contract number once that vendor becomes an official supplier of raw materials.

These business rules have been used to modify the E-R diagram in Figure 3-11 into the EER diagram in Figure 3-12. (We have left most attributes off this diagram except for those that are essential to see the changes that have occurred.) Rule 1 means that there is a total, overlapping specialization of CUSTOMER into REGULAR CUSTOMER and NATIONAL ACCOUNT CUSTOMER. A composite attribute of CUSTOMER, Customer Type (with components National and Regular), is used to designate whether a customer instance is a regular customer, a national account, or both. Because only regular customers do business in sales territories, only regular customers are involved in the Does Business In relationship (associative entity).

Rule 2 means that there is a partial, disjoint specialization of EMPLOYEE into MANAGEMENT EMPLOYEE and UNION EMPLOYEE. An attribute of EMPLOYEE, Employee Type, discriminates between the two special types of employees. Specialization is partial because there are other kinds of employees besides these two types. Only union employees are involved in the Works In relationship, but all union employees work in some work center, so the minimum cardinality of next to Works In

FIGURE 3-11 E-R diagram for Pine Valley Furniture Company

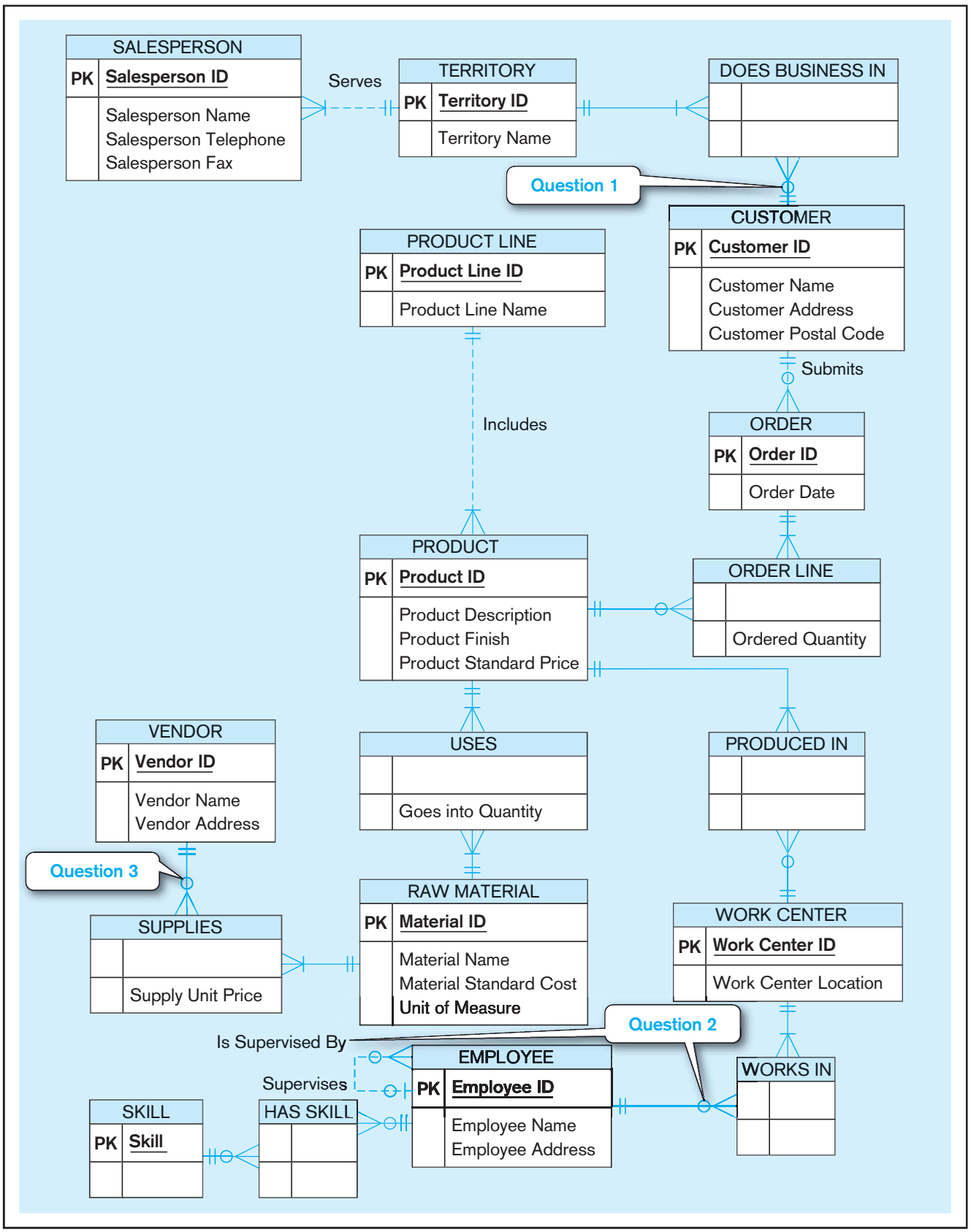
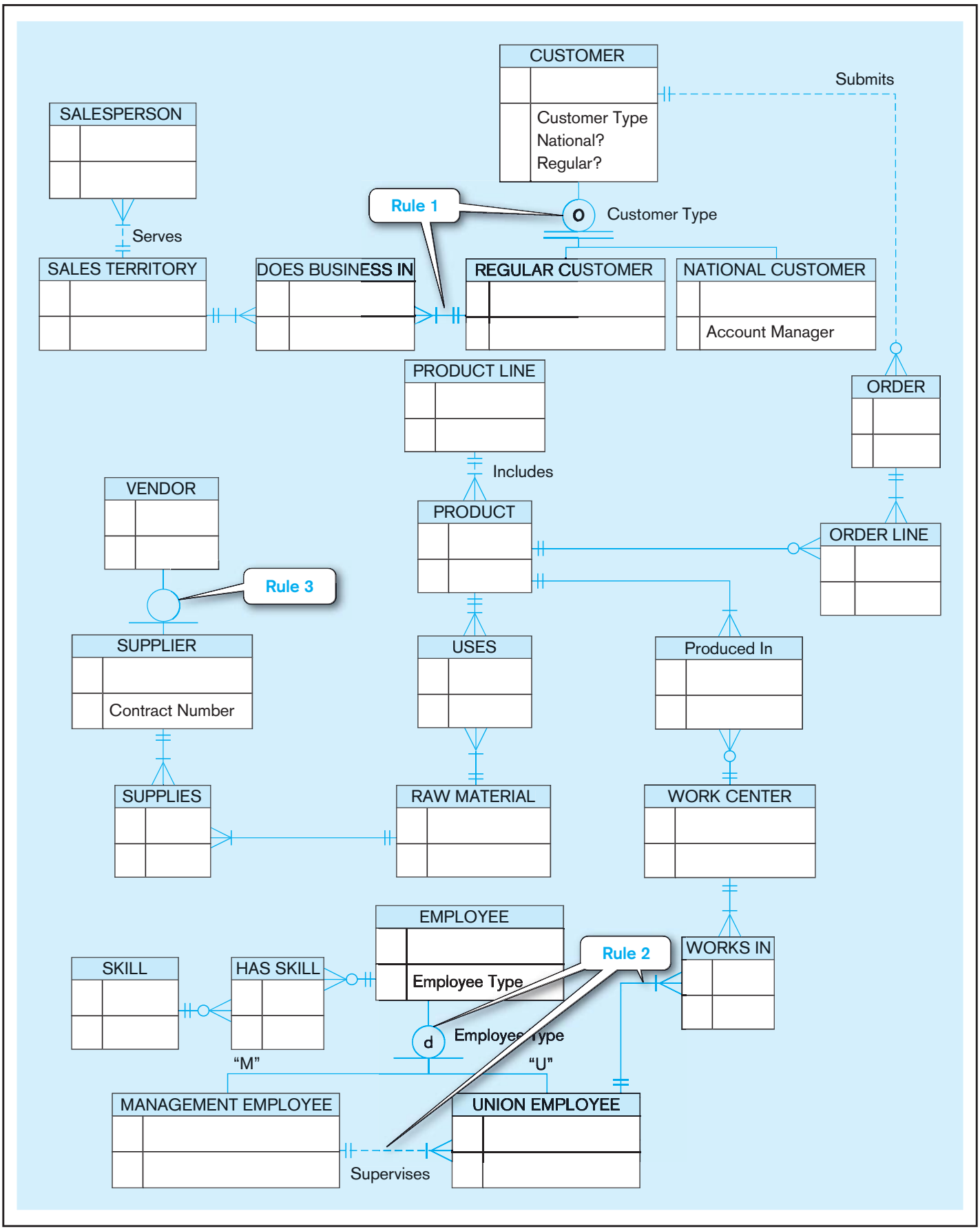


FIGURE 3-12 EER diagram for Pine Valley Furniture Company using Microsoft Visio



from UNION EMPLOYEE is now mandatory. Because an employee cannot be both management and union at the same time (although they can change status over time), the specialization is disjoint.

Rule 3 means that there is a partial specialization of VENDOR into SUPPLIER because only some vendors become suppliers. A supplier, not a vendor, has a contract number. Because there is only one subtype of VENDOR, there is no reason to specify a disjoint or overlap rule. Because all suppliers supply some raw material, the minimum cardinality next to RAW MATERIAL in the Supplies relationship (associative entity in Visio) now is one.

This example shows how an E-R diagram can be transformed into an EER diagram once generalization/specialization of entities is understood. Not only are supertype and subtype entities now in the data model, but additional attributes, including discriminating attributes, also are added, minimum cardinalities change (from optional to mandatory), and relationships move from the supertype to a subtype.

This is a good time to emphasize a point made earlier about data modeling. A data model is a conceptual picture of the data required by an organization. A data model does not map one-for-one to elements of an implemented database. For example, a database designer may choose to put all customer instances into one database table, not separate ones for each type of customer. Such details are not important now. The purpose now is to explain all the rules that govern data, not how data will be stored and accessed to achieve efficient, required information processing. We will address technology and efficiency issues in subsequent chapters when we cover database design and implementation.

Although the EER diagram in Figure 3-12 clarifies some questions and makes the data model in Figure 3-11 more explicit, it still can be difficult for some people to comprehend. Some people will not be interested in all types of data, and some may not need to see all the details in the EER diagram to understand what the database will cover. The next section addresses how we can simplify a complete and explicit data model for presentation to specific user groups and management.

ENTITY CLUSTERING

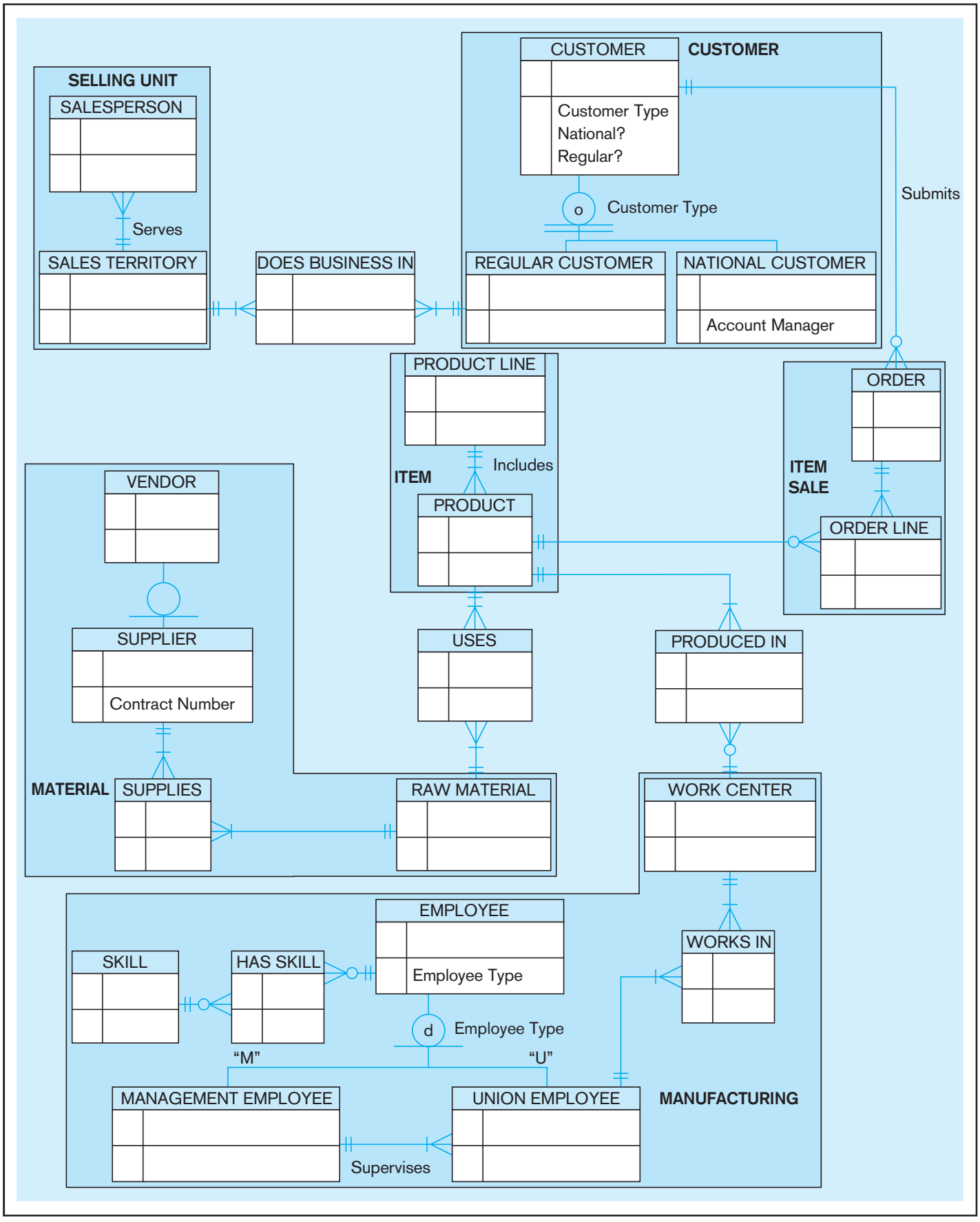
Some enterprise-wide information systems have more than 1,000 entity types and relationships. How do we present such an unwieldy picture of organizational data to developers and users? With a *really big* piece of paper? On the wrap-around walls of a large conference room? (Don't laugh about that one; we've seen it done!) Well, the answer is, we don't have to. In fact, there would be very few people who need to see the whole ERD in detail. If you are familiar with the principles of systems analysis and design (see, e.g., Hoffer et al., 2010), you know about the concept of functional decomposition. Briefly, functional decomposition is an iterative approach to breaking a system down into related components so that each component can be redesigned by itself without destroying the connections with other components. Functional decomposition is powerful because it makes redesign easier and allows people to focus attention on the part of the system in which they are interested. In data modeling, a similar approach is to create multiple, linked E-R diagrams, each showing the details of different (possibly overlapping) segments or subsets of the data model (e.g., different segments that apply to different departments, information system applications, business processes, or corporate divisions).

Entity cluster
A set of one or more entity types and associated relationships grouped into a single abstract entity type.

Entity clustering (Teorey, 1999) is a useful way to present a data model for a large and complex organization. An **entity cluster** is a set of one or more entity types and associated relationships grouped into a single abstract entity type. Because an entity cluster behaves like an entity type, entity clusters and entity types can be further grouped to form a higher-level entity cluster. Entity clustering is a hierarchical decomposition of a macro-level view of the data model into finer and finer views, eventually resulting in the full, detailed data model.

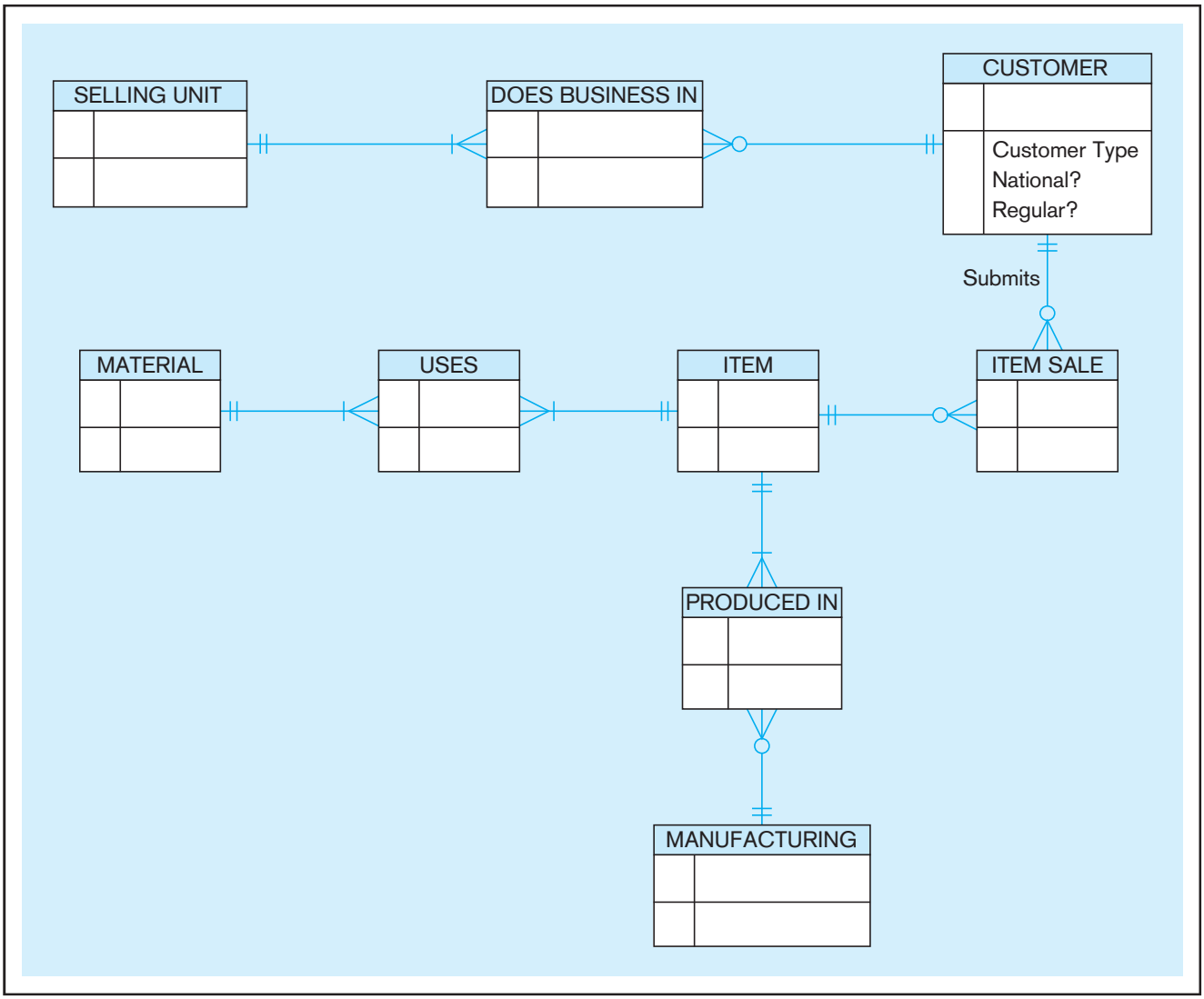
Figure 3-13 illustrates one possible result of entity clustering for the Pine Valley Furniture Company data model of Figure 3-12. Figure 3-13a shows the complete data

FIGURE 3-13 Entity clustering for Pine Valley Furniture Company
(a) Possible entity clusters (using Microsoft Visio)



(continued)

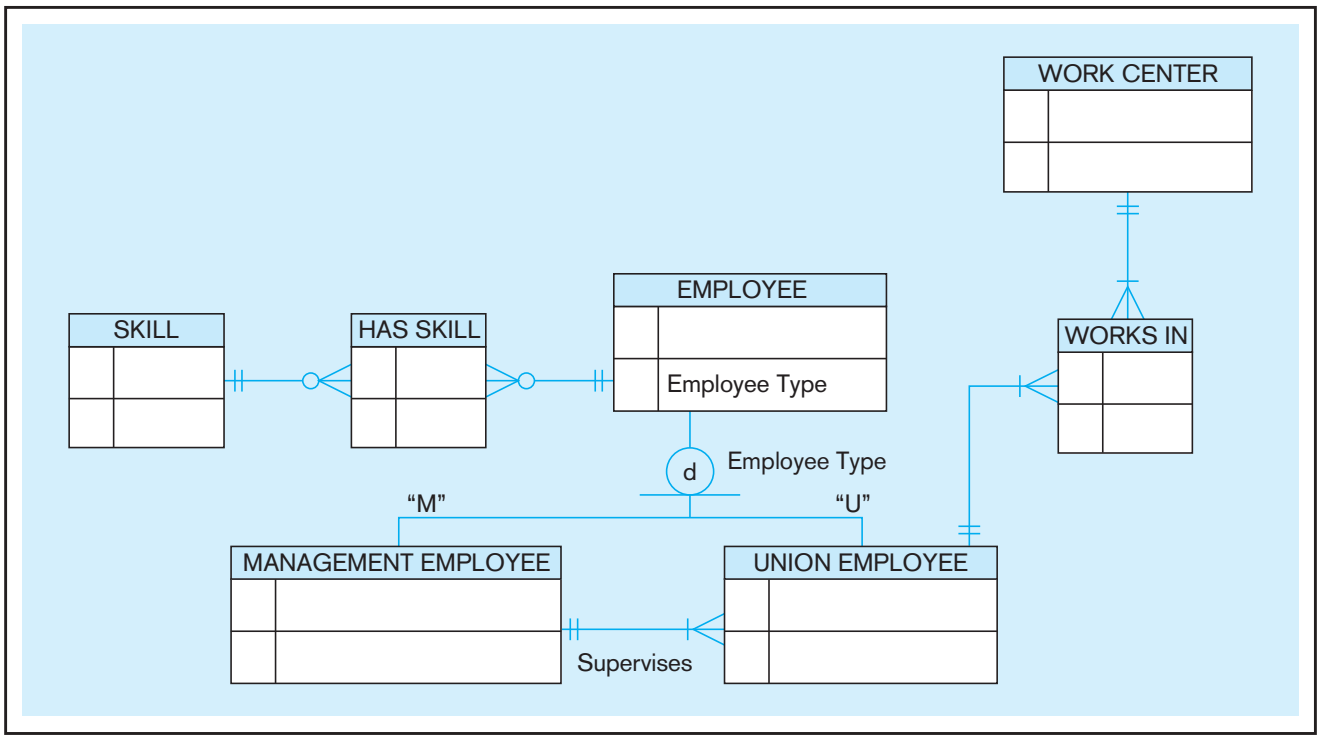
FIGURE 3-13 (*continued*)
(b) EER diagram for entity clusters (using Microsoft Visio)



model with shaded areas around possible entity clusters; Figure 3-13b shows the final result of transforming the detailed EER diagram into an EER diagram of only entity clusters and relationships. (An EER diagram may include both entity clusters and entity types, but this diagram includes only entity clusters.) In this figure, the entity cluster

- SELLING UNIT represents the SALESPERSON and SALES TERRITORY entity types and the Serves relationship
- CUSTOMER represents the CUSTOMER entity supertype, its subtypes, and the relationship between supertype and subtypes
- ITEM SALE represents the ORDER entity type and ORDER LINE associative entity as well as the relationship between them
- ITEM represents the PRODUCT LINE and PRODUCT entity types and the Includes relationship
- MANUFACTURING represents the WORK CENTER and EMPLOYEE supertype entity and its subtypes as well as the Works In associative entity and Supervises relationships and the relationship between the supertype and its subtypes. (Figure 3-14 shows an explosion of the MANUFACTURING entity cluster into its components.)
- MATERIAL represents the RAW MATERIAL and VENDOR entity types, the SUPPLIER subtype, the Supplies associative entity, and the supertype/subtype relationship between VENDOR and SUPPLIER.

FIGURE 3-14 MANUFACTURING entity cluster



The E-R diagrams in Figures 3-13 and 3-14 can be used to explain details to people most concerned with assembly processes and the information needed to support this part of the business. For example, an inventory control manager can see in Figure 3-13b that the data about manufacturing can be related to item data (the Produced In relationship). Furthermore, Figure 3-14 shows what detail is kept about the production process involving work centers and employees. This person probably does not need to see the details about, for example, the selling structure, which is embedded in the SELLING UNIT entity cluster.

Entity clusters in Figure 3-13 were formed (1) by abstracting a supertype and its subtype (see the CUSTOMER entity cluster) and (2) by combining directly related entity types and their relationships (see the SELLING UNIT, ITEM, MATERIAL, and MANUFACTURING entity clusters). An entity cluster can also be formed by combining a strong entity and its associated weak entity types (not illustrated here). Because entity clustering is hierarchical, if it were desirable, we could draw another EER diagram in which we combine the SELLING UNIT and CUSTOMER entity clusters with the DOES BUSINESS IN associative entity one entity cluster, because these are directly related entity clusters.

An entity cluster should focus on an area of interest to some community of users, developers, or managers. Which entity types and relationships are grouped to form an entity cluster depends on your purpose. For example, the ORDER entity type could be grouped in with the CUSTOMER entity cluster and the ORDER LINE entity type could be grouped in with the ITEM entity cluster in the example of entity clustering for the Pine Valley Furniture data model. This regrouping would eliminate the ITEM SALE cluster, which might not be of interest to any group of people. Also, you can do several different entity clusterings of the full data model, each with a different focus.

PACKAGED DATA MODELS

According to Len Silverston (1998), "The age of the data modeler as artisan is passing. Organizations can no longer afford the labor or time required for handcrafting data models from scratch. In response to these constraints, the age of the data modeler as engineer is dawning." As one executive explained to us, "the acquisition of a [packaged data model] was one of the key strategic things his organization did to gain

quick results and long-term success" for the business. Packaged data models are a game-changer for data modeling.

As introduced in Chapter 2, an increasingly popular approach of beginning a data modeling project is by acquiring a packaged or predefined data model, either a so-called universal model or an industry-specific model (some providers call these logical data models [LDMs], but these are really EER diagrams as explained in this chapter; the data model may also be part of a purchased software package, such as an enterprise resource planning or customer relationship management system). These packaged data models are not fixed, rather, the data modeler customizes the predefined model to fit the business rules of his or her organization based on a best-practices data model for their industry (e.g., transportation or communications) or chosen functional area (e.g., finance or manufacturing). The key assumption of this data modeling approach is that underlying structures or patterns of enterprises in the same industry or functional area are similar. Packaged data models are available from various consultants and database technology vendors. Although packaged data models are not inexpensive, many believe the total cost is lower and the quality of data modeling is better by using such resources. Some generic data models can be found in publications (e.g., see articles and books by Hay and by Silverston listed at the end of this chapter).

Universal data model
A generic or template data model that can be reused as a starting point for a data modeling project.

A **universal data model** is a generic or template data model that can be reused as a starting point for a data modeling project. Some people call these data model patterns, similar to the notion of patterns of reusable code for programming. A universal data model is not the "right" data model, but it is a successful starting point for developing an excellent data model for an organization.

Why has this approach of beginning from a universal data model for conducting a data modeling project become so popular? The following are some of the most compelling reasons professional data modelers are adopting this approach (we have developed this reasoning from Hoberman, 2006, and from an in-depth study we have conducted at the leading online retailer Overstock.com, which has adopted several packaged data models from Teradata Corporation):

- Data models can be developed using proven components evolved from cumulative experiences (as stated by the data administrator in the company we studied, "why reinvent when you can adapt?"). These data models are kept up-to-date by the provider as new kinds of data are recognized in an industry (e.g., RFID).
- Projects take less time and cost because the essential components and structures are already defined and only need to be quickly customized to the particular situation. The company we studied stated that the purchased data model was about 80 percent right before customization and that the cost of the package was about equal to the cost of one database modeler for one year.
- Data models are less likely to miss important components or make modeling errors due to not recognizing common possibilities. For example, the company we studied reported that their packaged data models helped them to avoid the temptation of simply mirroring existing databases, with all the historical "warts" of poor naming conventions, data structures customized for some historical purpose, and the inertia of succumbing to the pressure to simply duplicate the inadequate past practices. As another example, one vendor of packaged data models, Teradata, claims that one of its data models was scoped using more than 1,000 business questions and key performance indicators.
- Because of a holistic, enterprise view and development from best practices of data modeling experts found in a universal data model, the resulting data model for a particular enterprise tends to be easier to evolve as additional data requirements are identified for the given situation. A purchased model results in reduced rework in the future because the package gets it correct right out of the box and anticipates the future needs.
- The generic model provides a starting point for asking requirements questions so that most likely all areas of the model are addressed during requirements

determination. In fact, the company we studied said that their staff was "intrigued by all the possibilities" to meet even unspoken requirements from the capabilities of the prepackaged data models.
- Data models of an existing database are easier to read by data modelers and other data management professionals the first time because they are based on common components seen in similar situations.
- Extensive use of supertype/subtype hierarchies and other structures in universal data models promotes reusing data and taking a holistic, rather than narrow, view of data in an organization.
- Extensive use of many-to-many relationships and associative entities even where a data modeler might place a one-to-many relationship gives the data model greater flexibility to fit any situation, and naturally handles time stamping and retention of important history of relationships, which can be important to comply with regulations and financial record-keeping rules.
- Adaptation of a data model from your DBMS vendor usually means that your data model will easily work with other applications from this same vendor or their software partners.
- If multiple companies in the same industry use the same universal data model as the basis for their organizational databases, it may be easier to share data for interorganizational systems (e.g., reservation systems between rental car and airline firms).

A Revised Data Modeling Process with Packaged Data Models

Data modeling from a packaged data model requires no less skill than data modeling from scratch. Packaged data models are not going to put data modelers out of work (or keep you from getting that job as an entry-level data analyst you want now that you've started studying database management!). In fact, working with a package requires advanced skills, like those you are learning in this chapter and Chapter 2. As we will see, the packaged data models are rather complex because they are thorough and developed to cover all possible circumstances. A data modeler has to be very knowledgeable of the organization as well as the package to customize the package to fit the specific rules of that organization.

What do you get when you purchase a data model? What you are buying is metadata. You receive, usually on a CD, a fully populated description of the data model, usually specified in a structured data modeling tool, such as ERwin from Computer Associates or Oracle Designer from Oracle Corporation. The supplier of the data model has drawn the EER diagram, named and defined all the elements of the data model, and given all the attributes characteristics of data type (character, numeric, image), length, format, and so forth. You can print the data model and various reports about its contents to support the customization process. Once you customize the model, you can then use the data modeling tool to automatically generate the SQL commands to define the database to a variety of database management systems.

How is the data modeling process different when starting with a purchased solution? The following are the key differences (our understanding of these differences are enhanced by the interviews we conducted at Overstock.com):

- Because a purchased data model is extensive, you begin by identifying the parts of the data model that apply to your data modeling situation. Concentrate on these parts first and in the most detail. Start, as with most data modeling activities, first with entities, then attributes, and finally relationships. Consider how your organization will operate in the future, not just today.
- You then rename the identified data elements to terms local to the organization rather than the generic names used in the package.
- In many cases, the packaged data model will be used in new information systems that replace existing databases as well as to extend into new areas. So the next step is to map the data to be used from the package to data in current databases. One way this mapping will be used is to design migration plans to convert existing

databases to the new structures. The following are some key points about this mapping process:

- There will be data elements from the package that are not in current systems, and there will be some data elements in current databases not in the package. Thus, some elements won't map between the new and former environments. This is to be expected because the package anticipates information needs you have not, yet, satisfied by your current databases and because you do some special things in your organization that you want to retain but that are not standard practices. However, be sure that each non-mapped data element is really unique and needed. For example, it is possible that a data element in a current database may actually be derived from other more atomic data in the purchased data model. Also, you need to decide if data elements unique to the purchased data model are needed now or can be added on when you are ready to take advantage of these capabilities in the future.
- In general, the business rules embedded in the purchased data model cover all possible circumstances (e.g., the maximum number of customers associated with a customer order). The purchased data model allows for great flexibility, but a general-purpose business rule may be too weak for your situation (e.g., you are sure you will never allow more than one customer per customer order). As you will see in the next section, the flexibility and generalizability of a purchased data model results in complex relationships and many entity types. Although the purchased model alerts you to what is possible, you need to decide if you really need this flexibility and if the complexity is worthwhile.
- Because you are starting with a prototypical data model, it is possible to engage users and managers to be supported by the new database early and often in the data modeling project. Interviews, JAD sessions, and other requirements gathering activities are based on concrete ERDs rather than wish lists. The purchased data model essentially suggests specific questions to be asked or issues to discuss (e.g., "Would we ever have a customer order with more than one customer associated with it?" or "Might an employee also be a customer?"). The purchased model in a sense provides a visual checklist of items to discuss (e.g., Do we need these data? Is this business rule right for us?); further, it is comprehensive, so it is less likely that an important requirement will be missed.
- Because the purchased data model is comprehensive, there is no way you will be able to build and populate the full database or even customize the whole data model in one project. However, you don't want to miss the opportunity to visualize future requirements shown in the full data model. Thus, you will get to a point where you have to make a decision on what will be built first and possible future phases to build out as much of the purchased data model as will make sense. One approach to explaining the build-out schedule is to use entity clustering to show segments of the full data model that will be built in different phases. Future mini-projects will address detailed customization for new business needs and other segments of the data model not developed in the initial project.

You will learn in subsequent chapters of this book important database modeling and design concepts and skills that are important in any database development effort, including those based on purchased data models. There are, however, some important things to note about projects involving purchased data models. Some of these involve using existing databases to guide how to customize a purchased data model, including the following:

- Over time the same attribute may have been used for different purposes—what people call overloaded columns in current systems. This means that the data values in existing databases may not have uniform meaning for the migration to the new database. Oftentimes these multiple uses are not documented and are not known until the migration begins. Some data may no longer be needed (maybe used for a special business project), or there may be hidden requirements that were not formally incorporated into the database design. More on how to deal with this in a moment.

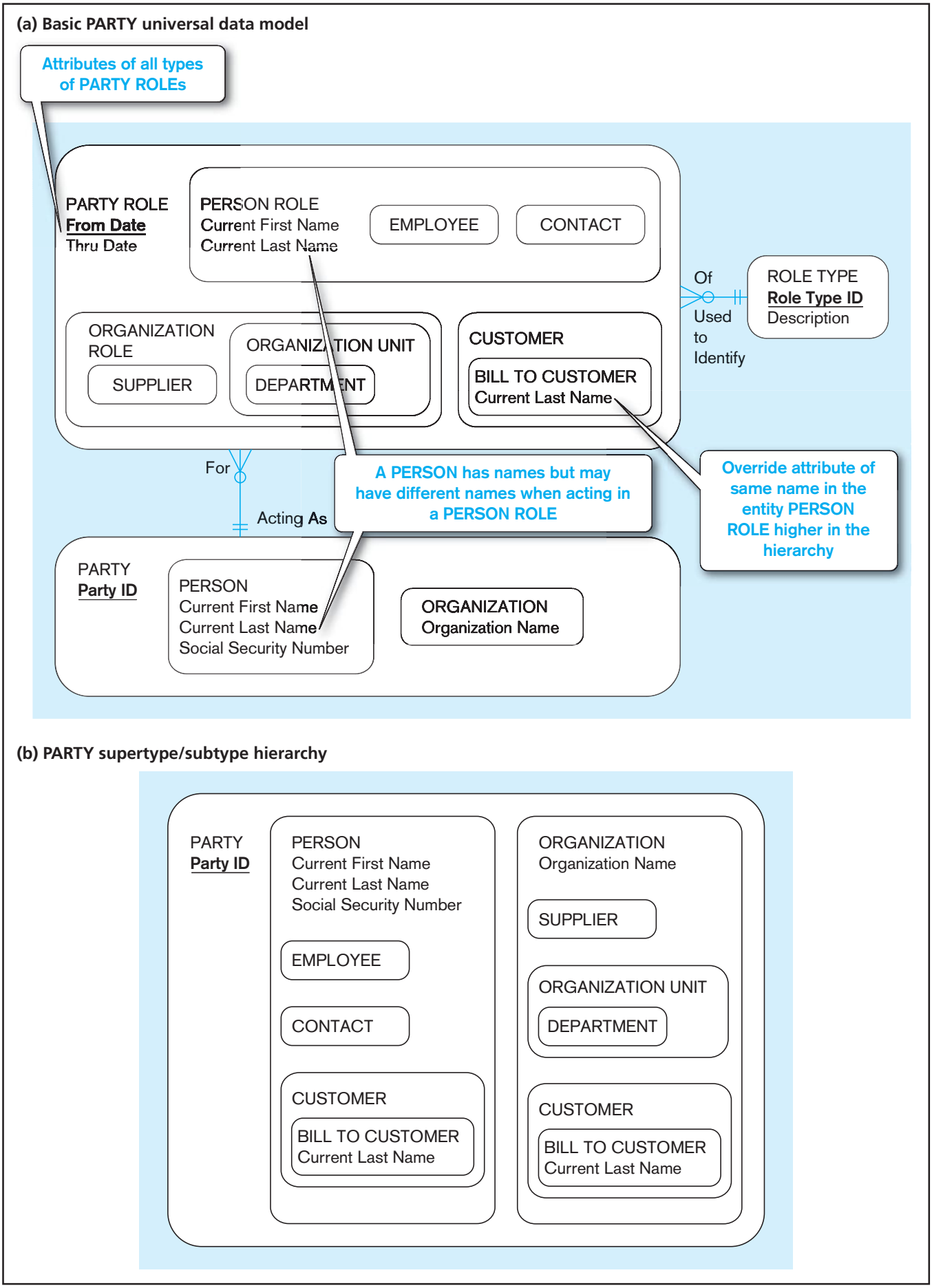
- Similarly, some attributes may be empty (i.e., have no values), at least for some periods of time. For example, some employee home addresses could be missing, or product engineering attributes for a given product line might be absent for products developed a few years ago. This could have occurred because of application software errors, human data entry mistakes, or other reasons. As we have studied, missing data may suggest optional data, and the need for entity subtypes. So missing data need to be studied to understand why the data are sparse.
- A good method for understanding hidden meaning and identifying inconsistencies in existing data models, and hence data and business rules that need to be included in the customized purchased data model, is data profiling. Profiling is a way to statistically analyze data to uncover hidden patterns and flaws. Profiling can find outliers, identify shifts in data distribution over time, and identify other phenomenon. Each perturbation of the distribution of data may tell a story, such as showing when major application system changes occurred, or when business rules changed. Often these patterns suggest poorly designed databases (e.g., data for separate entities were combined to improve processing speed for a special set of queries but the better structure was never restored). Data profiling can also be used to assess how accurate current data are and anticipate the clean-up effort that will be needed to populate the purchased data model with high-quality data.
- Arguably the most important challenge of customizing a purchased data model is determining the business rules that will be established through the data model. A purchased data model will anticipate the vast majority of the needed rules, but each rule must be verified for your organization. Fortunately, you don't have to guess which ones to address; each is laid out by the entities, attributes, and relationships with their metadata (names, definitions, data types, formats, lengths, etc.) in the purchased model. It simply takes time to go through each of these data elements with the right subject matter experts to make sure you have the relationship cardinalities and all other aspects of the data model right.

Packaged Data Model Examples

What, then, do packaged or universal data models look like? Central to the universal data model approach are supertype/subtype hierarchies. For example, a core structure of any universal data model is the entity type PARTY, which generalizes persons or organizations as actors for the enterprise, and an associated entity type PARTY ROLE, which generalizes various roles parties can play at different times. A PARTY ROLE instance is a situation in which a PARTY acts in a particular ROLE TYPE. These notions of PARTY, PARTY ROLE, and ROLE TYPE supertypes and their relationship are shown in Figure 3-15a. We use the supertype/subtype notation from Figure 3-1c because this is the notation most frequently used in publically available universal data models. (Most packaged data models are proprietary intellectual property of the vendor, and, hence, we cannot show them in this textbook.) This is a very generic data model (albeit simple to begin our discussion). This type of structure allows a specific party to serve in different roles during different time periods. It allows attribute values of a party to be "overridden" (if necessary in the organization) by values pertinent to the role being played during the given time period (e.g., although a PERSON of the PARTY supertype has a Current Last Name as of now, when in the party role of BILL TO CUSTOMER a different Current Last Name could apply during the particular time period [From Date to Thru Date] of that role). Note that even for this simple situation, the data model is trying to capture the most general circumstances. For example, an instance of the EMPLOYEE subtype of the PERSON ROLE subtype of PARTY ROLE would be associated with an instance of the ROLE TYPE that describes the employee-person role-party role. Thus, one description of a role type explains all the instances of the associated party roles of that role type.

An interesting aspect of Figure 3-15a is that PARTY ROLE actually simplifies what could be a more extensive set of subtypes of PARTY. Figure 3-15b shows one PARTY supertype with many subtypes covering many party roles. With partial specialization and overlap of subtypes, this alternative would appear to accomplish the same data

FIGURE 3-15 PARTY, PARTY ROLE, and ROLE TYPE in a universal data model



modeling semantics as Figure 3-15a. However, Figure 3-15a recognizes the important distinction between enterprise actors (PARTYs) and the roles each plays from time to time (PARTY ROLEs). Thus, the PARTY ROLE concept actually adds to the generalization of the data model and the universal applicability of the predefined data model.

The next basic construct of most universal data models is the representation of relationships between parties in the roles they play. Figure 3-16 shows this next extension of the basic universal data model. PARTY RELATIONSHIP is an associative entity, which hence allows any number of parties to be related as they play particular roles. Each instance of a relationship between PARTYs in PARTY ROLEs would be a separate instance of a PARTY RELATIONSHIP subtype. For example, consider the employment of a person by some organization unit during some time span, which over time is a many-to-many association. In this case, the EMPLOYMENT subtype of PARTY RELATIONSHIP would (for a given time period) likely link one PERSON playing the role of an EMPLOYEE subtype of PARTY ROLE with one ORGANIZATION ROLE playing some pertinent party role, such as ORGANIZATION UNIT. (That is, a person is

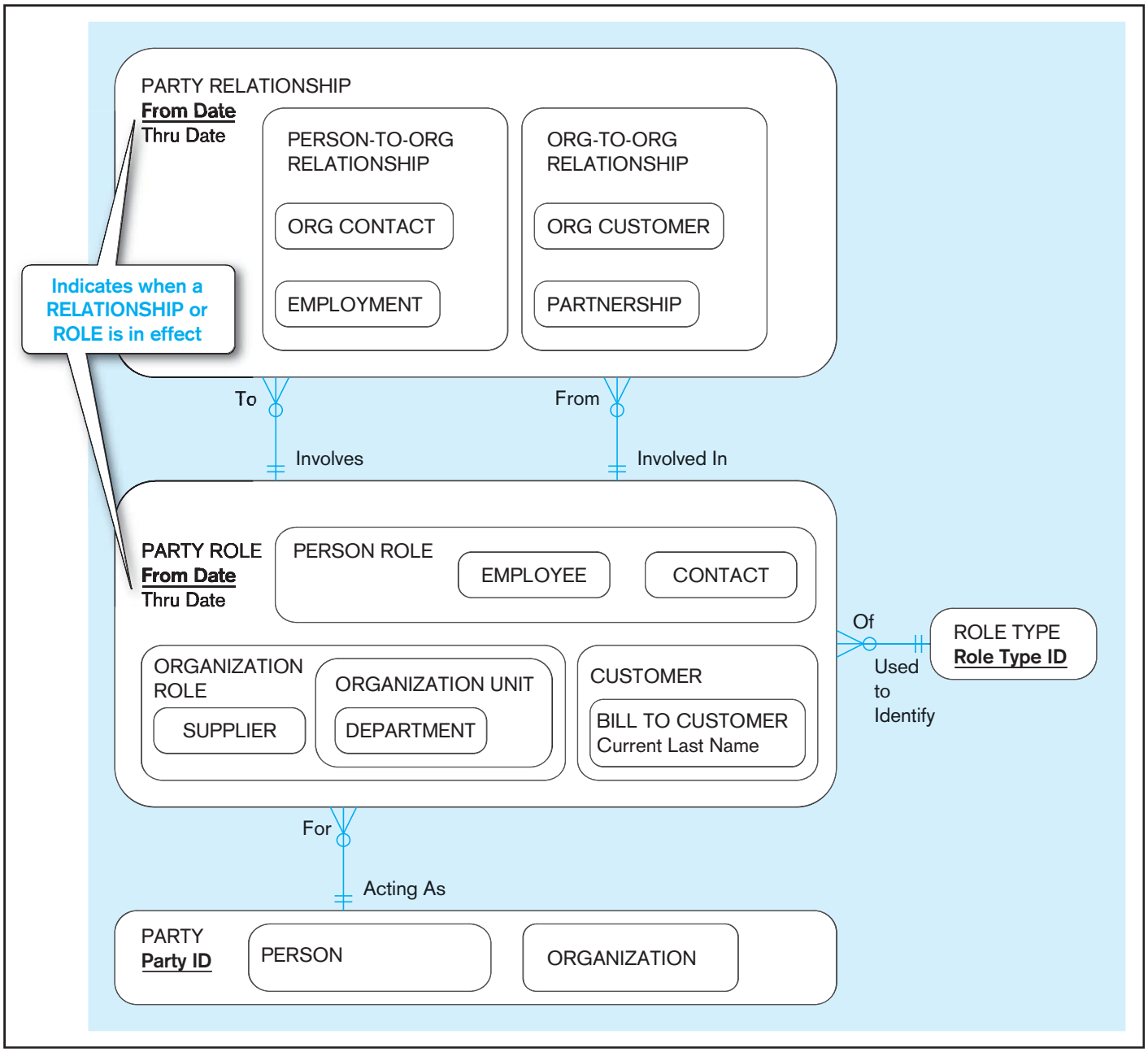


FIGURE 3-16 **Extension of a universal data model to include PARTY RELATIONSHIPs**

employed in an organization unit during a period of From Date to Thru Date in PARTY RELATIONSHIP.)

PARTY RELATIONSHIP is represented very generally, so it really is an associative entity for a unary relationship among PARTY ROLE instances. This makes for a very general, flexible pattern of relationships. What might be obscured, however, are which subtypes might be involved in a particular PARTY RELATIONSHIP, and stricter relationships that are not many-to-many, probably because we don't need to keep track of the relationship over time. For example, because the Involves and Involved In relationships link the PARTY ROLE and PARTY RELATIONSHIP supertypes, this does not restrict EMPLOYMENT to an EMPLOYEE with an ORGANIZATION UNIT. Also, if the enterprise needs to track only current employment associations, the data model in Figure 3-16 will not enforce that a PERSON PARTY in an EMPLOYEE PARTY ROLE can be associated with only one ORGANIZATION UNIT at a time. We will see in the next section how we can include additional business rule notation on an EER diagram to make this specific. Alternatively, we could draw specific relationships from just the EMPLOYEE PARTY ROLE and the ORGANIZATION UNIT PARTY ROLE to the EMPLOYMENT PARTY RELATIONSHIP to represent this particular one-to-many association. As you can imagine, to handle very many special cases like this would create a diagram with a large number of relationships between PARTY ROLE and PARTY RELATIONSHIP, and, hence, a very busy diagram. Thus, more restrictive cardinality rules (at least most of them) would likely be implemented outside the data model (e.g., in database stored procedures or application programs) when using a packaged data model.

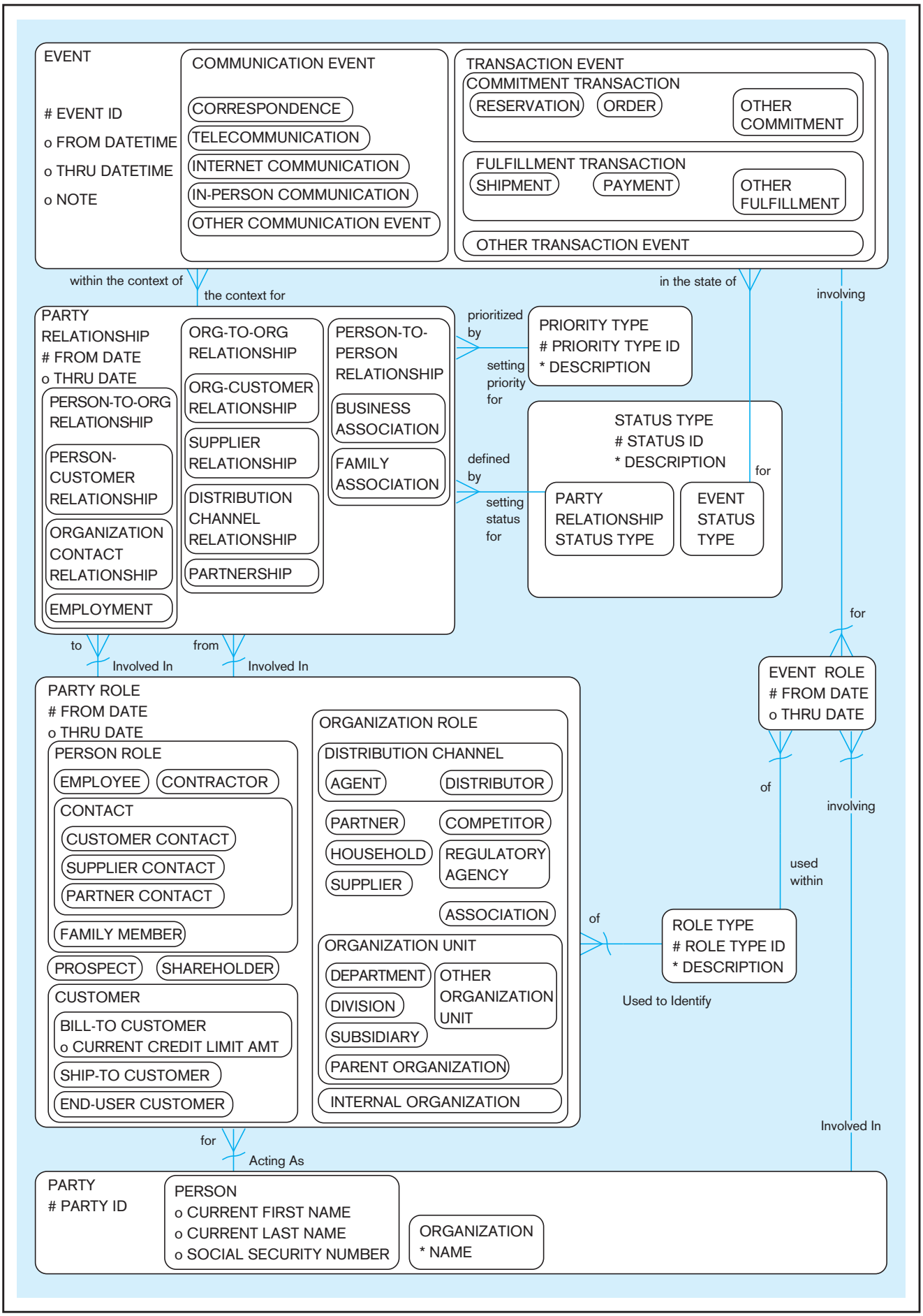
We could continue introducing various common, reusable building blocks of universal data models. However, Silverston in a two-volume set (2001a, 2001b) and Hay (1996) provide extensive coverage. To bring our discussion of packaged, universal data models to a conclusion, we show in Figure 3-17, a universal data model for a relationship development organization. In this figure, we use the original notation of Silverston (see several references at the end of the chapter), which is pertinent to Oracle data modeling tools. Now that you have studied EER concepts and notations and have been introduced to universal data models, you can understand more about the power of this data model.

To help you better understand the EER diagram in Figure 3-17, consider the definitions of the highest-level entity type in each supertype/subtype hierarchy:

PARTY	Persons and organizations independent of the roles they play
PARTY ROLE	Information about a party for an associated role, thus allowing a party to act in multiple roles
PARTY RELATIONSHIP	Information about two parties (those in the "to" and "from" roles) within the context of a relationship
EVENT	Activities that may occur within the context of relationships (e.g., a CORRESPONDENCE can occur within the context of a PERSON-CUSTOMER relationship in which the "to" party is a CUSTOMER role for an ORGANIZATION and the from party is an EMPLOYEE role for a PERSON)
PRIORITY TYPE	Information about a priority that may set the priority for a given PARTY RELATIONSHIP
STATUS TYPE	Information about the status (e.g., active, inactive, pending) of events or party relationships
EVENT ROLE	Information about all of the PARTYs involved in an EVENT
ROLE TYPE	Information about the various PARTY ROLEs and EVENT ROLEs

In Figure 3-17, supertype/subtype hierarchies are used extensively. For example, in the PARTY ROLE entity type, the hierarchy is as many as four levels deep (e.g., PARTY ROLE to PERSON ROLE to CONTACT to CUSTOMER CONTACT). Attributes can be located with any entity type in the hierarchy [e.g., PARTY has the identifier PARTY ID (# means identifier), PERSON has three optional attributes (*o* means optional), and ORGANIZATION has a required attribute (* means required)].

FIGURE 3-17 **A universal data model for relationship development**



Relationships can be between entity types anywhere in the hierarchy. For example, any EVENT is "in the state of" an EVENT STATUS TYPE, a subtype, whereas any EVENT is "within the context of" a PARTY RELATIONSHIP, a supertype.

As stated previously, packaged data models are not meant to be exactly right straight out of the box for a given organization; they are meant to be customized. To be the most generalized, such models have certain properties before they are customized for a given situation:

1. Relationships are connected to the highest-level entity type in a hierarchy that makes sense. Relationships can be renamed, eliminated, added, and moved as needed for the organization.
2. Strong entities almost always have *M:N* relationships between them (e.g., EVENT and PARTY), so at least one, and sometimes many, associative entities are used. Consequently, all relationships are 1:*M*, and there is an entity type in which to store intersection data. Intersection data are often dates, showing over what span of time the relationship was valid. Thus, the packaged data model is designed to allow tracking of relationships over time. (Recall that this is a common issue that was discussed with Figure 2-20.) 1:*M* relationships are optional, at least on the many side (e.g., the dotted line next to EVENT for the "involving" relationship signifies that an EVENT may involve an EVENT ROLE, as is done with Oracle Designer).
3. Although not clear on this diagram, all supertype/subtype relationships follow the total specialization and overlap rules, which makes the diagram as thorough and flexible as possible.
4. Most entities on the many side of a relationship are weak, thus inheriting the identifier of the entity on the one side (e.g., the ~ on the "acting as" relationship from PARTY to PARTY ROLE signifies that PARTY ROLE implicitly includes PARTY ID).

Summary

This chapter has described how the basic E-R model has been extended to include supertype/subtype relationships. A *supertype* is a generic entity type that has a relationship with one or more subtypes. A *subtype* is a grouping of the entities in an entity type that is meaningful to the organization. For example, the entity type PERSON is frequently modeled as a supertype. Subtypes of PERSON may include EMPLOYEE, VOLUNTEER, and CLIENT. Subtypes inherit the attributes and relationships associated with their supertype.

Supertype/subtype relationships should normally be considered in data modeling with either (or both) of the following conditions present: First, there are attributes that apply to some (but not all) of the instances of an entity type. Second, the instances of a subtype participate in a relationship unique to that subtype.

The techniques of generalization and specialization are important guides in developing supertype/subtype relationships. Generalization is the bottom-up process of defining a generalized entity type from a set of more specialized entity types. Specialization is the top-down process of defining one or more subtypes of a supertype that has already been defined.

The EER notation allows us to capture the important business rules that apply to supertype/subtype relationships. The completeness constraint allows us to specify whether an instance of a supertype must also be a member of at least one subtype. There are two cases: With total specialization, an instance of the supertype must be a member of at least one subtype. With partial specialization, an instance of a supertype may or may not be a member of any subtype. The disjointness constraint allows us to specify whether an instance of a supertype may simultaneously be a member of two or more subtypes. Again, there are two cases. With the disjoint rule, an instance can be a member of only one subtype at a given time. With the overlap rule, an entity instance can simultaneously be a member of two (or more) subtypes.

A subtype discriminator is an attribute of a supertype whose values determine to which subtype (or subtypes) a supertype instance belongs. A supertype/subtype hierarchy is a hierarchical arrangement of supertypes and subtypes, where each subtype has only one supertype.

There are extensions to the E-R notation other than supertype/subtype relationships. One of the other useful extensions is aggregation, which represents how some entities are part of other entities (e.g., a PC is composed of a disk drive, RAM, a motherboard, etc.). Due to space limitations we have not discussed these extensions here. Most of these extensions, like aggregation, are also a part of object-oriented data modeling, which is explained in Chapters 13 and 14.

E-R diagrams can become large and complex, including hundreds of entities. Many users and managers do not need to see all the entities, relationships, and attributes to understand the part of the database with

which they are most interested. Entity clustering is a way to turn a part of an entity-relationship data model into a more macro-level view of the same data. An entity cluster is a set of one or more entity types and associated relationships grouped into a single abstract entity type. Several entity clusters and associated relationships can be further grouped into even a higher entity cluster, so entity clustering is a hierarchical decomposition technique. By grouping entities and relationships, you can lay out an E-R diagram to allow you to give attention to the details of the model that matter most in a given data modeling task.

Packaged data models, so called universal and industry-specific data models, extensively utilize EER features. These generalizable data models often use multiple level supertype/subtype hierarchies and associative entities. Subjects and the roles subjects play are separated, creating many entity types; this complexity can be simplified when customized for a given organization, and entity clusters can be used to present simpler views of the data model to different audiences.

The use of packaged data models can save considerable time and cost in data modeling. The skills required for a data modeling project with a packaged data model are quite advanced and are built on the data modeling principles covered in this text. You have to consider not only current needs but also future requirements to customize the general-purpose package. Data elements must be renamed to local terms, and current data need to be mapped to the target database design. This mapping can be challenging due to various forms of mismatches between the data in current databases with those found in the best-practices purchased database model. Fortunately, having actual data models "right out of the box" helps to structure the customization process for completeness and ease of communication with subject matter experts. Overloaded columns, poor metadata, and abuses of the structure of current databases can make the customization and migration processes challenging. Data profiling can be used to understand the current data and uncover hidden meanings and business rules in data for your organization.

Chapter Review

Key Terms

Attribute inheritance *117*
Completeness constraint *121*
Disjoint rule *122*
Disjointness constraint *122*
Enhanced entity-relationship (EER) model *113*
Entity cluster *130*
Generalization *118*
Overlap rule *123*
Partial specialization rule *121*
Specialization *119*
Subtype *114*
Subtype discriminator *123*
Supertype *114*
Supertype/subtype hierarchy *125*
Total specialization rule *121*
Universal data model *134*

Review Questions

1. Define each of the following terms:
   a. supertype
   b. subtype
   c. specialization
   d. entity cluster
   e. completeness constraint
   f. enhanced entity-relationship (EER) model
   g. subtype discriminator
   h. total specialization rule
   i. generalization
   j. disjoint rule
   k. overlap rule
   l. partial specialization rule
   m. universal data model

2. Match the following terms and definitions:

____ supertype	a. subset of supertype
____ entity cluster	b. entity belongs to two subtypes
____ subtype	c. subtype gets supertype attributes
____ specialization	d. generalized entity type
____ subtype discriminator	e. creating subtypes for an entity type
____ attribute inheritance	f. a group of associated entity types and relationships
____ overlap rule	g. locates target subtype for an entity

3. Contrast the following terms:
   a. supertype; subtype
   b. generalization; specialization
   c. disjoint rule; overlap rule
   d. total specialization rule; partial specialization rule
   e. PARTY; PARTY ROLE
   f. entity; entity cluster

4. State two conditions that indicate when a database designer should consider using supertype/subtype relationships.

5. State the reason for entity clustering.

6. Give an example (other than those discussed in the chapter) of a supertype/subtype relationship.

7. What is attribute inheritance? Why is it important?

8. Give an example of each of the following:
   a. a supertype/subtype relationship where the disjoint rule applies
   b. a supertype/subtype relationship where the overlap rule applies

9. What types of business rules are normally captured in an EER diagram?

10. What is the purpose of a subtype discriminator?

11. When would a packaged data model be useful?

12. In what ways is starting a data modeling project with a packaged data model different from starting a data modeling project with a clean sheet of paper?

13. How can data profiling be used during a data modeling project, especially one using a packaged data model?
14. Does a data modeling project using a packaged data model require less or greater skill than a project not using a packaged data model? Why or why not?
15. What do you purchase when you acquire a packaged data model?
16. When might a supertype/subtype hierarchy be useful?
17. When is a member of a supertype always a member of at least one subtype?

Problems and Exercises

1. Examine the hierarchy for the university EER diagram (Figure 3-10). As a student, you are an instance of one of the subtypes: either UNDERGRAD STUDENT or GRADUATE STUDENT. List the names of all the attributes that apply to you. For each attribute, record the data value that applies to you.
2. Add a subtype discriminator for each of the supertypes shown in Figure 3-10. Show the discriminator values that assign instances to each subtype. Use the following subtype discriminator names and values:
   a. PERSON: Person Type (Employee? Alumnus? Student?)
   b. EMPLOYEE: Employee Type (Faculty, Staff)
   c. STUDENT: Student Type (Grad, Undergrad)
3. For simplicity, subtype discriminators were left off many figures in this chapter. Add subtype discriminator notation in each figure listed below. If necessary, create a new attribute for the discriminator.
   a. Figure 3-2
   b. Figure 3-3
   c. Figure 3-4b
   d. Figure 3-7a
   e. Figure 3-7b
4. Refer to the employee EER diagram in Figure 3-2. Make any assumptions that you believe are necessary. Develop a sample definition for each entity type, attribute, and relationship in the diagram.
5. Refer to the EER diagram for patients in Figure 3-3. Make any assumptions you believe are necessary. Develop sample definitions for each entity type, attribute, and relationship in the diagram.
6. Figure 3-13 shows the development of entity clusters for the Pine Valley Furniture E-R diagram. In Figure 3-13b, explain the following:
   a. Why is the minimum cardinality next to the DOES BUSINESS IN associative entity coming from CUSTOMER zero?
   b. What would be the attributes of ITEM (refer to Figure 2-22)?
   c. What would be the attributes of MATERIAL (refer to Figure 2-22)?
7. A rental car agency classifies the vehicles it rents into four categories: compact, midsize, full-size, and sport utility. The agency wants to record the following data for all vehicles: Vehicle ID, Make, Model, Year, and Color. There are no unique attributes for any of the four classes of vehicle. The entity type vehicle has a relationship (named Rents) with a customer entity type. None of the four vehicle classes has a unique relationship with an entity type. Would you consider creating a supertype/subtype relationship for this problem? Why or why not?
8. At a weekend retreat, the entity type PERSON has three subtypes: CAMPER, BIKER, and RUNNER. Draw a separate EER diagram segment for each of the following situations:
   a. At a given time, a person must be exactly one of these subtypes.
   b. A person may or may not be one of these subtypes. However, a person who is one of these subtypes cannot at the same time be one of the other subtypes.
   c. A person may or may not be one of these subtypes. On the other hand, a person may be any two (or even three) of these subtypes at the same time.
   d. At a given time, a person must be at least one of these subtypes.
9. A bank has three types of accounts: checking, savings, and loan. Following are the attributes for each type of account:
   CHECKING: Acct No, Date Opened, Balance, Service Charge
   SAVINGS: Acct No, Date Opened, Balance, Interest Rate
   LOAN: Acct No, Date Opened, Balance, Interest Rate, Payment
   Assume that each bank account must be a member of exactly one of these subtypes. Using generalization, develop an EER model segment to represent this situation using the traditional EER notation, the Visio notation, or the subtypes inside supertypes notation, as specified by your instructor. Remember to include a subtype discriminator.
10. Refer to your answer to Problem and Exercise 20 in Chapter 2. Develop entity clusters for this E-R diagram and redraw the diagram using the entity clusters. Explain why you chose the entity clusters you used.
11. Refer to your answer to Problem and Exercise 23 in Chapter 2. Develop entity clusters for this E-R diagram and redraw the diagram using the entity clusters. Explain why you chose the entity clusters you used.
12. Draw an EER diagram for the following problem using this textbook's EER notation, the Visio notation, or the subtypes inside supertypes notation, as specified by your instructor.

    A nonprofit organization depends on a number of different types of persons for its successful operation. The organization is interested in the following attributes for all of these persons: SSN, Name, Address, City/State/Zip, and Telephone. Three types of persons are of greatest interest: employees, volunteers, and donors. Employees have only a Date Hired attribute, and volunteers have only a Skill attribute. Donors have only a relationship (named Donates) with an Item entity type. A donor must have donated one or more items, and an item may have no donors, or one or more donors.

    There are persons other than employees, volunteers, and donors who are of interest to the organization, so that a person need not belong to any of these three groups. On the other hand, at a given time a person may belong to two or more of these groups (e.g., employee and donor).

13. Add a subtype discriminator (named Person Type) to the diagram you created in Problem and Exercise 12.

14. Develop an EER model for the following situation, using the traditional EER notation, the Visio notation, or the subtypes inside supertypes notation, as specified by your instructor:

A person may be employed by one or more organizations, and each organization may be the employer of one or more persons. An organization can be an internal organizational unit or an external organization. For persons and organizations, we want to know their ID, name, address, and phone number. For persons, we want to know their birth date, and for organizations, we want to know their budget number. For each employment, we want to know the employment date, termination date, and bonus.

Employment of a person by an organization may result in the person holding many positions over time. For each position, we want to know its title, and each time someone holds that position, we need to know the start date and termination date and salary. An organization is responsible for each position. It is possible for a person to be employed by one organization and hold a position for which another organization is responsible.

15. Draw an EER diagram for the following description of a law firm:

Each case handled by the firm has a unique case number; a date opened, date closed, and judgment description are also kept on each case. A case is brought by one or more plaintiffs, and the same plaintiff may be involved in many cases. A plaintiff has a requested judgment characteristic. A case is against one or more defendants and the same defendant may be involved in many cases. A plaintiff or defendant may be a person or an organization. Over time, the same person or organization may be a defendant or a plaintiff in cases. In either situation, such legal entities are identified by an entity number, and other attributes are name and net worth.

16. Develop an EER model for the following situation using the traditional EER notation, the Visio notation, or the subtypes inside supertypes notation, as specified by your instructor:

An international school of technology has hired you to create a database management system to assist in scheduling classes. After several interviews with the president, you have come up with the following list of entities, attributes, and initial business rules:

- Room is identified by Building ID and Room No and also has a Capacity. A room can be either a lab or a classroom. If it is a classroom, it has an additional attribute called Board Type.
- Media is identified by MType ID and has attributes of Media Type and Type Description. Note: Here we are tracking type of media (such as a VCR, projector, etc.), not the individual piece of equipment. Tracking of equipment is outside of the scope of this project.
- Computer is identified by CType ID and has attributes Computer Type, Type Description, Disk Capacity, and Processor Speed. Please note: As with Media Type, we are tracking only the type of computer, not an individual computer. You can think of this as a class of computers (e.g., PIII 900MHZ).
- Instructor has identifier Emp ID and has attributes Name, Rank, and Office Phone.
- Timeslot has identifier TSIS and has attributes Day Of Week, Start Time, and End Time.
- Course has identifier Course ID and has attributes Course Description and Credits. Courses can have one, none, or many prerequisites. Courses also have one or more sections.
- Section has identifier Section ID and attribute Enrollment Limit.

After some further discussions, you have come up with some additional business rules to help you create the initial design:

- An instructor teaches one, none, or many sections of a course in a given semester.
- An instructor specifies preferred time slots.
- Scheduling data are kept for each semester, uniquely identified by semester and year.
- A room can be scheduled for one section or no section during one time slot in a given semester of a given year. However, one room can participate in many schedules, one schedule, or no schedules; one time slot can participate in many schedules, one schedule, or no schedules; one section can participate in many schedules, one schedule, or no schedules. Hint: Can you associate this to anything that you have seen before?
- A room can have one type of media, several types of media, or no media.
- Instructors are trained to use one, none, or many types of media.
- A lab has one or more computer types. However, a classroom does not have any computers.
- A room cannot be both a classroom and a lab. There also are no other room types to be incorporated into the system.

17. Develop an EER model for the following situation using the traditional EER notation, the Visio notation, or the subtypes inside supertypes notation, as specified by your instructor:

Wally Los Gatos and his partner Henry Chordate have formed a new limited partnership, Fin and Finicky Security Consultants. Fin and Finicky consults with corporations to determine their security needs. You have been hired by Wally and Henry to design a database management system to help them manage their business.

Due to a recent increase in business, Fin and Finicky has decided to automate their client tracking system. You and your team have done a preliminary analysis and come up with the following set of entities, attributes, and business rules:

Consultant

There are two types of consultants: business consultants and technical consultants. Business consultants are contacted by a business in order to first determine security needs and provide an estimate for the actual services to be performed. Technical consultants perform services according to the specifications developed by the business consultants.

Attributes of business consultant are the following: Employee ID (identifier), Name, Address (which is composed of Street, City, State, and Zip Code), Telephone, Date Of Birth, Age, Business Experience (which is composed of Number of Years, Type of Business [or businesses], and Degrees Received).

Attributes of technical consultant are the following: Employee ID (identifier), Name, Address (which is composed of Street, City, State, and Zip Code), Telephone, Date Of Birth, Age, Technical Skills, and Degrees Received.

Customer

Customers are businesses that have asked for consulting services. Attributes of customer are Customer ID (identifier), Company Name, Address (which is composed of Street, City, State, and Zip Code), Contact Name, Contact Title, Contact Telephone, Business Type, and Number Of Employees.

Location

Customers can have multiple locations. Attributes of location are Customer ID (identifier), Location ID (which is unique only for each Customer ID), Address (which is composed of Street, City, State, and Zip Code), Telephone, and Building Size.

Service

A security service is performed for a customer at one or more locations. Before services are performed, an estimate is prepared. Attributes of service are Service ID (identifier), Description, Cost, Coverage, and Clearance Required.

Additional Business Rules

In addition to the entities outlined previously, the following information will need to be stored to tables and should be shown in the model. These may be entities, but they also reflect a relationship between more than one entity:

- Estimates, which have characteristics of Date, Amount, Business Consultant, Services, and Customer
- Services Performed, which have characteristics of Date, Amount, Technical Consultant, Services, and Customer

In order to construct the EER diagram, you may assume the following:

> A customer can have many consultants providing many services. You wish to track both actual services performed as well as services offered. Therefore, there should be two relationships between customer, service, and consultant, one to show services performed and one to show services offered as part of the estimate.

18. Based on the EER diagram constructed for Problem and Exercise 17, develop a sample definition for each entity type, attribute, and relationship in the diagram.

19. You are working for a large country club. This country club wants to keep a database on its members and their guests. For each member, the club keeps mail and telephone contact information, name, and membership number. When you join this club, you can join as a social member (which allows you two rounds of golf a year as well as privileges to the swimming pool and weight room), a tennis member (which allows you all the privileges of a social member as well as use of the tennis courts and four rounds of golf), or a golfing member (which allows you all the privileges of a tennis member and unlimited use of the golf course). This database needs to track how often a member (who has limited use of the golf course; all golfing members have unlimited use of the golf course) has used the golf course, and how many guests any and each member has brought to the club. All members have guest privileges. The club also wants to attract new members by mailing to all those who came to the club as guests and live in the state. The mailing includes information about their visits (i.e., date of visit and which member was their host for each visit). Once a person becomes a member of any type, information about them as guests is no longer important to retain. Develop an EER diagram for this situation.

20. Draw an EER diagram for the following situation:

> TomKat Entertainment is a chain of theaters owned by former husband and wife actors/entertainers who, for some reason, can't get a job performing anymore. The owners want a database to track what is playing or has played on each screen in each theater of their chain at different times of the day. A theater (identified by a Theater ID and described by a theater name and location) contains one or more screens for viewing various movies. Within each theater each screen is identified by its number and is described by the seating capacity for viewing the screen. Movies are scheduled for showing in time slots each day. Each screen can have different time slots on different days (i.e., not all screens in the same theater have movies starting at the same time, and even on different days the same movie may play at different times on the same screen). For each time slot, the owners also want to know the end time of the time slot (assume all slots end on the same day the slot begins), attendance during that time slot, and the price charged for attendance in that time slot. Each movie (which can be either a trailer, feature, or commercial) is identified by a Movie ID and further described by its title, duration, and type (i.e., trailer, feature, or commercial). In each time slot, one or more movies are shown. The owners want to also keep track of in what sequence the movies are shown (e.g., in a time slot there might be two trailers, followed by two commercials, followed by a feature film, and closed with another commercial).

21. Add the following to Figure 3-16: an EMPLOYMENT party relationship is further explained by the positions and assignments to positions during the time a person is employed. A position is defined by an organization unit, and a unit may define many positions over time. Over time, positions are assigned to various employment relationships (i.e., somebody employed by some organization unit is assigned a particular position). For example, a position of Business Analyst is defined by the Systems Development organization unit. Carl Gerber, while employed by the Data Warehousing organization unit, is assigned the position of Systems Analyst. In the spirit of universal data modeling, enhance Figure 3-16 for the most general case consistent with this description.

Field Exercises

1. Interview a friend or family member to elicit common examples of supertype/subtype relationships they may come into contact with at work. You will have to explain the meaning of this term to the person you are interviewing and provide a common example, such as PROPERTY: RESIDENTIAL, COMMERCIAL or BONDS: CORPORATE, MUNICIPAL. Use the information the person provides to construct an EER diagram segment and present it to the person. Revise, if necessary, until it seems appropriate to you and your friend or family member.
2. Visit two local small businesses, one in the service sector and one in manufacturing. Interview employees from these organizations to obtain examples of both supertype/subtype relationships and business rules (such as "A customer can return merchandise only with a valid sales slip"). In which of these environments is it easier to find examples of these constructs? Why?
3. Ask a database administrator or database or system analyst in a local company to show you an EER (or E-R) diagram for one of the organization's primary databases. Does this organization model have supertype/subtype relationships? If so, what notation is used, and does the CASE tool the company uses support these relationships? Also, what types of business rules are included during the EER modeling phase? How are business rules represented, and how and where are they stored?
4. Read the summary of business rules published by the GUIDE Business Rules Project (1997) and the article by Gottesdiener (1997). Search the Web for additional information on business rules. Then write a three-page executive summary of current directions in business rules and their potential impact on systems development and maintenance.
5. Research universal data models. Find articles on universal (or packaged, industry, or functional area) data models, or find information on some commercial offerings. Identify common features across these models as well as different ways to model the same concepts. Discuss what you think are the advantages and disadvantages of the different ways used to model the same concepts.

References

Elmasri, R., and S. B. Navathe. 1994. *Fundamentals of Database Systems*. Menlo Park, CA: Benjamin/Cummings.

Gottesdiener, E. 1997. "Business Rules Show Power, Promise." *Application Development Trends* 4,3 (March): 36–54.

GUIDE. 1997 (October). "GUIDE Business Rules Project." Final Report, revision 1.2.

Hay, D. C. 1996. *Data Model Patterns: Conventions of Thought*. New York: Dorset House Publishing.

Hoberman, S. 2006. "Industry Logical Data Models." *Teradata Magazine*. Available at **www.teradata.com**.

Hoffer, J. A., J. F. George, and J. S. Valacich. 2010. *Modern Systems Analysis and Design*, 6th ed. Upper Saddle River, NJ: Prentice Hall.

Silverston, L. 1998. "Is Your Organization Too Unique to Use Universal Data Models?" *DM Review* 8,8 (September), accessed at **www.information-management.com/issues/19980901/425-1.html**.

Silverston, L. 2001a. *The Data Model Resource Book, Volume 1*, Rev. ed. New York: Wiley.

Silverston, L. 2001b. *The Data Model Resource Book, Volume 2*, Rev. ed. New York: Wiley.

Silverston, L. 2002. "A Universal Data Model for Relationship Development." *DM Review* 12,3 (March): 44–47, 65.

Teorey, T. 1999. *Database Modeling & Design*. San Francisco: Morgan Kaufman Publishers.

Further Reading

Frye, C. 2002. "Business Rules Are Back." *Application Development Trends* 9, 7 (July): 29–35.

Moriarty, T. "Using Scenarios in Information Modeling: Bringing Business Rules to Life." *Database Programming & Design* 6, 8 (August): 65–67.

Ross, R. G. 1997. *The Business Rule Book*. Version 4. Boston: Business Rule Solutions, Inc.

Ross, R. G. 1998. *Business Rule Concepts: The New Mechanics of Business Information Systems*. Boston: Business Rule Solutions, Inc.

Ross, R. G. 2003. *Principles of the Business Rule Approach*. Boston: Addison-Wesley.

Schmidt, B. 1997. "A Taxonomy of Domains." *Database Programming & Design* 10, 9 (September): 95, 96, 98, 99.

Silverston, L. 2002. Silverston has a series of articles in *DM Review* that discuss universal data models in different settings. See in particular Vol. 12 issues 1 (January) on clickstream analysis, 5 (May) on health care, 7 (July) on financial services, and 12 (December) on manufacturing.

von Halle, B. 1996. "Object-Oriented Lessons." *Database Programming & Design* 9,1 (January): 13–16.

von Halle, B. 2001. von Halle has a series of articles in *DM Review* on building a business rules system. These articles are in Vol. 11, issues 1–5 (January–May).

von Halle, B., and R. Kaplan. 1997. "Is IT Falling Short?" *Database Programming & Design* 10, 6 (June): 15–17.

Web Resources

www.adtmag.com Web site of *Application Development Trends*, a leading publication on the practice of information systems development.

www.brsolutions.com Web site of Business Rule Solutions, the consulting company of Ronald Ross, a leader in the development of a business rule methodology. Or you can check out **www.BRCommunity.com**, which is a virtual community site for people interested in business rules (sponsored by Business Rule Solutions).

www.businessrulesgroup.org Web site of the Business Rules Group, formerly part of GUIDE International, which formulates and supports standards about business rules.

www.databaseanswers.org/data_models A fascinating site that shows over 100 sample E-R diagrams for a wide variety of applications and organizations. A variety of notations are used, so this a good site to also learn about variations in E-R diagramming.

www.intelligententerprise.com Web site of *Intelligent Enterprise*, a leading publication on database management and related areas. This magazine is the result of combining two previous publications, *Database Programming & Design* and *DBMS*.

www.kpiusa.com The homepage for Knowledge Partners International, founded by Barbara von Halle. This site has some interesting case studies and white papers about business rules.

http://researchlibrary.theserverside.net/detail/RES/1214505974_136.html Link to the white paper "Modeling Unstructured Data" by Steve Hoberman. Unstructured data (e.g., e-mails, images, sounds) is an emerging area for databases, and there are some special issues in data modeling for unstructured data.

www.tdan.com Web site of *The Data Administration Newsletter*, which regularly publishes new articles, special reports, and news on a variety of data modeling and administration topics.

www.teradatastudentnetwork.com Web site for Teradata Student Network, a free resource for a wide variety of information about database management and related topics. Go to this site and search on "entity relationship" to see many articles and assignments related to EER data modeling.

CASE

Mountain View Community Hospital

Case Description

After developing a preliminary E-R model and discussing it with the rest of your team, you realize that you need to delve deeper into the interview notes and documentation you obtained to add more detail to the model and possibly add entities and relationships you had overlooked. Several issues need to be addressed.

As a large service organization, Mountain View Community Hospital (MVCH) depends on four major groups of persons for its continued success: employees, physicians, patients, and volunteers. A small number of persons in the hospital community do not belong to any of these four groups. A particular person may belong to two (or more) of these groups at a given time. For example, a volunteer or employee may also be a patient at the hospital at some point in time.

The four groups of people listed previously share many common characteristics such as a unique identifier, Name, Address, City/State/Zip, Birth Date, Phone, and E-mail. Then there are characteristics that apply to only one of these groups. For example, a hire date (Date Hired) is recorded for employees only. Volunteer Services records skills and interests of their volunteers in order to place them appropriately. Physicians have a pager number (Pager#) and a DEA number (a physician needs a DEA registration number from the Drug Enforcement Administration to be able to prescribe controlled substances). For patients, the hospital records the date of first contact with the hospital (Contact Date). There are also characteristics that apply to some, but not all of the groups. For example, both physicians and nurses have a specialty (e.g., pediatrics, oncology, etc.).

In addition to the characteristics already mentioned, the hospital records a number of other characteristics about its patients: emergency contact information (last and first name, relationship to patient, address, and phone), insurance information (insurance company name, policy number, group number, and insurance phone number), information about the insurance subscriber in case the patient is not the insurance subscriber (last and first name, relationship to patient, address, and phone), and contact information for the patient's primary care physician or other physician who referred the patient to the hospital.

At MVCH, each patient has one (and only one) physician responsible for that patient. A given physician may not be responsible for a patient at a given time or may be responsible for one or more patients. The primary patient segments are resident patients and outpatients. Outpatients may come in for many reasons, including routine examinations at an outpatient care center (e.g., the MS Center), ambulatory/outpatient surgery, diagnostic services, or emergency room care. Each outpatient is scheduled for zero or more visits. A visit has several attributes: a unique identifier (Visit#), date, and time. Notice that an instance of visit cannot exist without an outpatient owner entity. Some patients that are seen as outpatients, for example, in the emergency room, are subsequently admitted to the hospital and become resident patients. Each resident patient has a Date Admitted attribute as well as a Discharge Date.

The volunteer application form in MVCH Figure 3-1 shows all the information that Volunteer Services under Mr. Davis requires from persons interested in volunteering. Volunteers work in many areas of the hospital based on their interests and skills. Volunteer Services keeps track of a person's time of service (begin and end date), work unit where a person works as a volunteer, and the volunteer's supervisor. Each volunteer is supervised by an employee or physician, but not all employees and physicians supervise volunteers. Volunteer Services also keep track of a volunteer's number of hours worked and recognizes outstanding volunteers at an annual awards ceremony.

Employees fall into three categories: nurses, technicians, and staff. Each nurse has a certificate/degree indicating his or her qualification as an RN or LPN. (LPNs work under the direction of RNs at MVCH.) Each nurse must also have a current Colorado nursing license and may hold certifications in special fields such as dialysis, pediatrics, anesthesia, critical care, pain management, and so on. Most nurses are assigned to one (and only one) care center at a time, although over time, they may be working in more than one care center. Some nurses are floaters who are not assigned to a specific care center but instead work wherever they are needed. As described earlier, one of the nurses assigned to a care center is appointed nurse-in-charge (Nurse In Charge). Only nurses with an RN certificate can be appointed nurse-in-charge.

Specific job-related competency skills are recorded for the hospital's technicians. A cardiovascular technician for example may be skilled in specific equipment, such as setting up and getting readings from a Holter monitor, a portable device that monitors a patient's EKG for a period of 24 to 48 hours during routine activities. Medical laboratory technicians need to be able to set up, operate, and control equipment, perform a variety of tests, analyze the test data, and summarize test results for physicians who use them to diagnose and treat patients. Emergency room technicians' skills include the ability to perform CPR, or set up an IV. Dialysis technicians, who may be skilled in different types of dialysis, (e.g., pediatric dialysis, outpatient dialysis) need a variety of skills related to setting up treatment, assessing the patient during dialysis, and assessing and troubleshooting equipment problems during dialysis. Each technician is assigned to a work unit in the hospital (a care center, the central medical laboratory, radiology, etc.).

Staff members have a job classification (Job Class), such as secretary, administrative assistant, admitting specialist, collection specialist, and so on. Like the technicians, each staff member is assigned to a work unit in the hospital (a care center, the central medical laboratory, radiology, etc.).

Work units such as a care center have a Name (identifier) and Location. The location denotes the facility (e.g., main building) and floor (e.g., 3 West, 2 South). A care center often has one or more beds (up to any number) assigned to it, but there are also care centers without assigned beds. The only attribute of bed is the identifier Bed ID, which consists of two components: Bed#

Mountain View Community Hospital
VOLUNTEER APPLICATION

Last Name _____ First Name _____ Date of Birth _____

Street Address _____ City _____ State ____ Zip _____

Home Phone: (___) _____ Work Phone: (___) _____ E-Mail _____

Have you been *convicted of a felony* within the past seven years?

○ NO ○ YES If YES, please explain _____

Emergency Contact Last Name _____ First Name _____

Relationship _____ Phone: (___) _____

References (Not Relatives)

Last Name _____	Last Name _____
First Name _____	First Name _____
Relationship _____ Phone (___) _____	Relationship _____ Phone (___) _____
Address: _____	Address _____
City _____ State _____ Zip _____	City _____ State _____ Zip _____

Current or Last Employment

Name of Employer _____

Employer Address _____

Position (Type of work) _____ Dates of Employment _____

Prior Volunteer Service

Have you volunteered at Mountain Valley Community Hospital before?

○ NO ○ YES If YES, please list _____

Do you have previous volunteer experience elsewhere?

○ NO ○ YES If YES, please list _____

Interests & Preferences

Why do you want to become a volunteer? _____

What are your hobbies, skills, other interests? _____

Which languages do you speak? _____

What do you envision yourself doing as a volunteer? _____

Please indicate days and times when you are available to volunteer.

	Mon	Tues	Wed	Thur	Fri	Sat	Sun
Morning							
Afternoon							
Evening							

Applicant's Signature _____ Date _____

MVCH FIGURE 3-1 **Volunteer application form**

and Room#. Each resident patient must be assigned to a bed. Because MVCH doesn't always fill all its beds, a bed may or may not have a resident patient assigned to it at a given time.

Case Questions

1. Is the ability to model supertype/subtype relationships important in a hospital environment such as MVCH? Why or why not?
2. Are there any weak entities, multivalued attributes, or multiple relationships in the description of the data requirements in this case segment? If so, what are they?
3. Can you think of any other business rules (other than the one explicitly described in the case) that are likely to be used in a hospital environment? Can these be represented on an EER diagram for MVCH?
4. Are there any universal data models that can be reused as a starting point for modeling MVCH's data requirements?

Would you recommend using such as model for the MVCH project? Why or why not?

Case Exercises

1. Draw an EER diagram to represent the requirements described in this case segment carefully following the notation from this chapter.
2. Suppose each care center had two nurses-in-charge, one for the day shift, and another one for the evening shift. How would that change the diagram you developed in Case Exercise 1?
3. Develop definitions for each of the following types of objects in your EER diagram from Case Exercise 1. Consult with some member of the hospital or health-care community (if one is available); do some research on the Internet, or otherwise make reasonable assumptions based on your own knowledge and experience.
   a. Entity types
   b. Attributes
   c. Relationships
4. Figure 3-17 shows the following entity types in a universal data model: PARTY, PARTY ROLE, PARTY RELATIONSHIP, EVENT, PRIORITY TYPE, STATUS TYPE, EVENT ROLE, and ROLE TYPE. How would these apply to the MVCH case? Give examples of each entity type based on the information provided in the case descriptions up to this point.
5. Derive and clearly state the business rules that are implicit in the volunteer application form shown in MVCH Figure 3-1.
6. Compare the EER diagram that you developed in this chapter with the E-R diagram you developed in Chapter 2. What are the differences between these two diagrams? Why are there differences?

Project Assignments

P1. Revise the list of business rules you developed in Chapter 2 in light of the information provided in this case segment and your insights from the Case Exercises.

P2. Following the notation from this chapter, merge your Chapter 2 E-R diagram with the EER diagram you developed for Case Exercises 1 and 2 to represent the data requirements for MVCH's new system.

P3. Document and explain the decisions you made during merging.